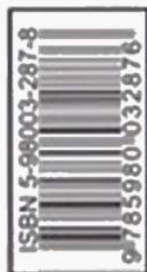


Библиотека Инженера

Николайчук О. И.

Современные средства автоматизации Практические решения



Командно-информационные сети

Схемотехника универсальных контроллеров

Проектирование систем малой автоматизации



Серия «Библиотека инженера»

О. И. Николайчук

**СОВРЕМЕННЫЕ
СРЕДСТВА
АВТОМАТИЗАЦИИ**

Практические решения

**Москва
СОЛОН-Пресс
2016**

УДК 621.396.218
ББК 32.884.1
Н28

О. И. Николайчук

Н28 Современные средства автоматизации / О. И. Николайчук — М.: СОЛОН-Пресс, 2016. 256 с. — (Серия «Библиотека инженера»)

ISBN 5-98003-287-8

Книга призвана ознакомить читателей с современными принципами и приемами разработки систем малой автоматизации — распределенных систем управления и контроля, реализованных на основе специализированных малых и средних локальных вычислительных сетей, ориентированных на работу в режиме реального времени. Эти системы широко применяются для автоматизации небольших технологических, измерительных, исследовательских и лабораторных установок, обычно территориально расположенных в пределах одного помещения, а в последнее время и для автоматизации домашнего хозяйства.

Предназначена для широкого круга читателей, интересующихся микроконтроллерной техникой, как профессионалов, так и любителей.

ISBN 5-98003-287-8

© Макет и обложка «СОЛОН-Пресс», 2016
© О. И. Николайчук, 2016

Введение

Одной из наиболее динамично развивающихся областей современной вычислительной техники является микроконтроллерная техника. Без микроконтроллеров сегодня немислим ни один бытовой прибор, не говоря уже о более сложных изделиях. Сегодняшний микроконтроллер — это достаточно мощный и быстродействующий «интеллектуальный» элемент, как правило, имеющий достаточно высокую производительность, значительные объемы встроенной памяти программ и оперативной памяти, достаточно мощное вычислительное ядро с системой команд, ориентированной на решение задач управления и контроля. Многие современные микроконтроллеры имеют развитую встроенную цифровую и аналоговую периферию: подсистемы прерываний, таймеры/счетчики, охранные таймеры, супервизоры питания, широкий набор быстродействующих усовершенствованных последовательных интерфейсов, таких как UART, SPI, SMBus (I2C), CAN, JTAG, аналого-цифровые и цифроаналоговые преобразователи, компараторы, мультиплексоры, масштабирующие усилители и другие узлы.

Современные микроконтроллеры и базирующиеся на них изделия являются основой так называемых «систем малой автоматизации», широко применяющихся для автоматизации небольших технологических, измерительных, исследовательских и лабораторных систем, а также для автоматизации сложного бытового оборудования и домашнего хозяйства. Используемые в системах малой автоматизации контроллеры должны быть достаточно многофункциональными и универсальными по структуре, а также обеспечивать работу в составе современных специализированных локальных сетей, имеющих упрощенный алгоритм работы и содержащих персональный компьютер в качестве главного управляющего элемента. Фактически современные системы малой автоматизации, построенные на базе специализированных локальных сетей, объединяющих несколько универсальных контроллеров, являются рассредоточенным интеллектуальным интерфейсом персонального компьютера, объединяющим в себе измерительные, контрольные и управляющие функции.

Схемотехника универсальных контроллеров, используемых в системах малой автоматизации, имеет ряд специфических особенностей и требует детального рассмотрения и определенной методики проектирования. Состав функциональных узлов таких контроллеров и принципы их построения определяются рядом требований, направленных на обеспечение высокой производительности, универсальности, гибкости, надежности, высокой ремонтпригодности, низкой себестоимости и т. п.

Специализированные локальные сети, используемые в системах малой автоматизации, также имеют ряд специфических особенностей, связанных с необходимостью обеспечения упрощенного алгоритма функционирования, высокой надежности и производительности, низкой стоимости, простоты установки, наладки и обслуживания.

В связи с этим настоящая книга посвящается детальному ознакомлению читателя с основными требованиями к современным универсальным контроллерам систем малой автоматизации, методикой подбора современной элементной базы, схемотехникой и принципами построения основных функциональных узлов. Детально рассматриваются также специализированные локальные вычислительные сети, их структура, алгоритмы функционирования, схемотехника. Приводится упрощенная методика расчета производительности, фрагменты тестовых программ. Кроме того, приводятся несколько принципиальных схем универсальных и специализированных контроллеров с рабочими программами, а также базовая программа контроллера системы малой автоматизации.

Книга рассчитана на широкий круг инженерно-технических работников, а также студентов и аспирантов, интересующихся принципами построения систем малой автоматизации и последними достижениями современной микроконтроллерной техники.

Автор будет весьма признателен за все предложения и критические замечания, которые он просит направлять в адрес редакции.

1. Системы малой автоматизации

1.1. Предпосылки создания систем малой автоматизации

История развития современной вычислительной техники берет свое начало с середины 40-х годов прошлого века. По инициативе гарвардского математика Говарда Эйкена при финансовой поддержке президента компании IBM Томаса Уотсона в 1939 году начались работы по созданию первой вычислительной машины Mark-1, которая была официально передана Гарвардскому университету в 1944 году. В то же время был сформулирован ряд идей, которые и по настоящее время считаются концептуальными. К этим идеям следует отнести: концепцию хранения программы в памяти компьютера, предложенную в 1944 году американским инженером Джоном Эккертом; идеи по архитектуре электронных вычислительных машин, предложенные в 1946 году Джоном фон Нейманом; книгу американского ученого Норберта Винера «Кибернетика, или управление и связь у животных и машин», в которой изложены основные предпосылки развития теории автоматов, а также вводится понятие кибернетики как науки об управлении и передаче информации; книгу Клода Шеннона «Математическая теория передачи информации». Все эти события способствовали созданию в 1951 году под руководством Сергея Александровича Лебедева первой отечественной цифровой электронно-вычислительной машины МЭСМ (Малая Электронная Счетная Машина). В 1952 году была создана отечественная Быстродействующая Электронная Счетная Машина — БЭСМ [1—5].

Первые цифровые электронно-вычислительные машины имели центральное процессорное устройство (*CPU — Central Processor Unit*), которое управляло абсолютно всеми периферийными устройствами. По мере дальнейшего развития научной и инженерной мысли в области вычислительной техники, а также элементной базы, создавались все более совершенные электронные вычислительные машины (*CS — Computing Systems*). Они оснащались все большим набором периферийных устройств: всевозможных инженерных пультов, алфавитно-цифровых дисплеев, алфавитно-цифровых печатающих устройств, устройств ввода-вывода информации, устройств хранения информации на различных носителях (магнитных картах, лентах, дисках) и т. п. Периферийные устройства также все время совершенствовались. Постепенно управление ими все более усложнялось, что отвлекало все большие вычислительные мощности со стороны *CPU*. Со временем возникла ситуация, когда разработчики вычислительной техники поняли, что дальнейшее усложнение периферийных устройств будет отбирать все больше вычислительной мощности от *CPU* и что альтернативой этому будет оснащение периферийной техники собственными устройствами управления или локальными *CPU*, которые будут управлять непосредственно периферийным устройством. Таким образом, идея децентрализации управления перифе-

рийной техникой позволяла освободить значительную часть вычислительной мощности основного *CPU*.

Следует отметить, что в то время в СССР, как и в других странах, существовали два независимых направления в развитии средств вычислительной техники: гражданское и военное.

Первое направление было ориентировано на создание *CS* и их применение в различных областях народного хозяйства, для проведения различных научных и инженерных расчетов. В рамках этого направления работали многочисленные научно-исследовательские учреждения Академии наук и вузов, а также промышленные предприятия. В середине 60-х годов в русле этого направления началось создание *CS* для управления производственными и технологическими процессами в реальном времени, развитие которых было поручено Минприбору СССР [1]. Дальнейшее развитие элементной базы, совершенствование архитектуры средств вычислительной техники, научной и инженерной мысли привело к созданию управляющих вычислительных машин — *CCS (Control CS)*. В тоже время развитие научных исследований и появление ряда новых научно-исследовательских задач способствовало развитию идей децентрализации и созданию распределенных модульных систем автоматизации научных исследований и производства [6].

Второе направление не менее интенсивно и широко развивалось в организациях военно-промышленного комплекса. Оно практически не было, в силу известных причин, отражено в открытой печати. Уже в конце 50-х годов начались интенсивные работы по применению вычислительной техники в различных системах военного назначения: противовоздушной и противоракетной обороны, контроля космического пространства, управления полетами в авиации и космосе, системах управления разными родами войск [7]. Эти системы по понятным причинам были территориально рассредоточены, занимали очень большие площади, но в то же время требовали четкой согласованности действий и оперативной связи для обмена информацией. Одновременно начали интенсивно развиваться бортовые специализированные вычислительные машины — *SCS (Specialized Computing Systems)*. Они были предназначены для управления различной военной техникой: самолетами, танками, морскими кораблями и т. п. [8—10]. Многие бортовые системы вначале создавались как автономные, территориально рассредоточенные *CS* для управления различными подсистемами и агрегатами, а лишь затем, в ходе совершенствования техники, возникла необходимость в жесткой координации их действий и оперативному обмену данными между ними. Это привело к попыткам связи рассредоточенных *CS* в единую целостную систему с общим управлением. Таким образом, примерно к концу 60-х годов окончательно сформировались идеи и накопился некоторый начальный опыт создания рассредоточенных систем [11—13].

В далеком 1962 году в рамках работ, развернутых по инициативе президента США Эйзенхауэра совместно с военными ведомствами и ставивших своей целью обеспечение превосходства США в науке и техническом развитии (в ответ на запуск в СССР первого искусственного спутника Земли, 04.10.1965), было создано агентство *ARPA (Advanced Research Projects Agency)*. При этом

агентстве был создан новый департамент технологий обработки информации (*Information Processing Techniques Office, IPTO*), которому было поручено изучать и совершенствовать технологии контроля и управления. Первым директором IPTO был психолог, специалист по поведению человека, сотрудник Массачусетского технологического института Джозеф Карл Ликлайдер (*Joseph Carl Licklider*). В середине 1960-х он опубликовал свои идеи относительно будущей роли вычислителей, взаимодействующих друг с другом в среде передачи данных с множественным доступом. Считается, что именно эти идеи явились первым предвестником создания вычислительных сетей. Немного раньше, в 1961 году в Массачусетском технологическом институте была опубликована работа Леонарда Клейнрока (*Leonard Kleinrock*), будущего теоретика *Internet*. В этой работе впервые высказывалась идея технологии «коммутации пакетов». Смысл этой идеи заключался в том, что при передаче информации на неопределенно большое расстояние в течение неопределенно большого времени через неопределенное количество промежуточных узлов блок передаваемого сообщения должен быть заключен в капсулу, содержащую все необходимые сведения о сообщении для того, чтобы любой промежуточный узел мог определить его дальнейшее направление, а приемный узел — принять и проверить целостность. Эти идеи послужили, в некотором смысле, основой для создания и развития в дальнейшем сетевых технологий. В 1963 году в США был создан Институт инженеров по электротехнике и радиоэлектронике (*IEEE*), ставший впоследствии главным разработчиком массовых стандартов в области локальных сетей (*LAN — Local Area Network*). Первая в мире *LAN* была создана в 1967 г. Дональдом Дэвисом (*Donald Davies*) в Национальной физической лаборатории Великобритании (*British National Physics Laboratory*). Что было далее? Далее в мире начался процесс создания все новых и новых сетевых технологий, который часто называют «сетевым бумом» [14—18].

Появление на рубеже 70-х годов принципиально новой элементной базы — микропроцессоров послужило предпосылкой для создания и развития новых систем обработки данных — систем с распределенной структурой, называемых также децентрализованными или распределенными системами управления (*DCS — Distributed Control System*) [19—24]. Одновременно развивались и идеи создания различных сетей: в первую очередь глобальных, а затем и локальных. Схожесть целей и путей реализации *DCS* и *LAN*, а также их практически одновременное параллельное развитие часто приводило к некоторой путанице в терминологии. Только значительно позже, с развитием классификации и стандартизации, были определены основные отличия между этими понятиями.

Опыт создания периферийных микропроцессорных контроллеров, а также быстрое развитие архитектуры и возможностей микропроцессоров привели к ускоренному развитию *DCS*. Создавались как глобальные (например родом войск) и средние распределенные системы управления (распределенная система управления заводом), так и малые распределенные системы управления (автомобилем, отдельным конвейером, танком и т. п.). В 1975 году в ФИАН им. П. Н. Лебедева (Москва) была разработана первая в СССР локальная сеть [14]. В то же

время начали развиваться системы автоматизации научных исследований и системы управления технологическими процессами в реальном времени [6, 13—18].

К сожалению, так называемая «компьютерная революция» происходила достаточно медленно примерно до второй половины 70-х годов, что было обусловлено совершенствованием архитектуры и технологии производства микропроцессорной техники.

Одновременно с этим процессом происходило переосмысливание основных принципов построения буквально всех изделий, связанных с вычислительной техникой и, конечно же, эти процессы повлияли на идеи и принципы развития *DCS* и *LAN*.

Повышение степени интеграции привело к созданию новых мощных микропроцессоров и микроконтроллеров и систем на их основе — *MPS* (*Microprocessor Systems*), выполненных на одной или нескольких больших интегральных схемах (*ICs*), что окончательно сделало экономически выгодным развитие *DCS* и *LAN* [19—35].

Начиная с 80-х годов и по настоящее время темпы развития *DCS* постоянно возрастают. Этому способствует постоянное расширение сферы производства, совершенствование элементной базы при неуклонном снижении ее стоимости, развитие научно-технической мысли в областях архитектуры вычислительных систем и сетевых технологий, расширение областей автоматизации человеческой деятельности и т. п. В тоже время постепенно в мире снижается удельный вес гигантских промышленных предприятий и научно-исследовательских центров. Их заменяют экономически выгодные и мобильные (в научном и производственном отношении), а также менее энергоемкие малые промышленные фирмы и исследовательские центры.

В области развития *DCS* также происходят некоторые перемены. Такие традиционные области применения *DCS*, как малые системы автоматизации производства реального времени, малые специализированные распределенные системы управления, малые системы автоматизации научного эксперимента и малые информационно-измерительные системы, которые долгое время развивались независимо и параллельно, в последнее время все чаще проектируются по сходным принципам и, в некотором смысле, объединяются. Во многих развитых странах такие же системы применяются и для автоматизации домашнего хозяйства [36]. В специальной литературе такие малые распределенные системы управления реального времени часто называются *системами малой автоматизации* — *SAS* (*Small Automation Systems*).

В системах *SAS* используются все знания и практический опыт, накопленные при создании вышеперечисленных систем *DCS* и малых локальных сетей *LAN*. Современные *SAS* выполняются на самой современной элементной базе и работают в режиме реального времени. В то же время характерными основными чертами этих систем является низкая стоимость, простота реализации, высокие надежность и быстродействие. Основам схемотехники именно таких систем посвящена настоящая книга.

1.2. Основные понятия о распределенных системах управления

Для того чтобы сформулировать основные признаки распределенных систем управления *DCS*, следует более подробно остановиться на особенностях применения микропроцессоров (*MP* — *Microprocessor*), а позднее и микроконтроллеров (*MC* — *Microcontroller*) в различных системах.

Во-первых, отметим, что для большинства микропроцессоров и микроконтроллеров их вычислительная мощность не является основной чертой. Немаловажное значение для них имеют управляющие, коммутирующие и логические возможности. Именно они, в первую очередь, важны для организации эффективного взаимодействия *MP* и *MC* между собой (в сложных системах со многими *MP*) и с объектами управления (или измерения). Существует также некоторое обратное воздействие «решаемой задачи» на структуру и организацию *MP*.

Практически все современные *MC* кроме собственно вычислительного ядра оснащены дополнительными (периферийными) функциональными узлами, не свойственными ранним моделям, которые появились вследствие анализа наиболее часто встречающихся задач [12, 19, 20].

Во-вторых, снижение стоимости и повышение вычислительной мощности *MP* и *MC* привели к перераспределению ценовых акцентов в вычислительных системах. Ведь если раньше наиболее дорогостоящим в вычислительных системах было именно *CPU*, а периферийное оборудование стоило относительно недорого, то в последнее время именно периферийное оборудование имеет в вычислительных системах наибольший ценовой вес. Например, если ранее существовали тенденции к созданию распределенных вычислительных систем с целью повышения общей вычислительной мощности, то в настоящее время достаточно часто создаются распределенные системы с целью коллективного пользования дорогостоящим периферийным оборудованием. Все это, в конечном итоге, приводит к новым способам внутрисистемного взаимодействия и организации распределенных систем.

В-третьих, при создании распределенных систем разработчики стремятся разделить задачу на некие автономные, слабо взаимосвязанные фрагменты. Это существенно упрощает аппаратную и программную реализацию управления небольшими фрагментами объекта управления. С другой стороны, это приводит к тому, что *DCS* может содержать достаточно большое количество *MP* (или *MC*) — от десятков до нескольких сотен. Таким образом, эти системы являются мультимикропроцессорными (*MMPS* — *Multi Microprocessor Systems*). Наличие такого большого количества автономно работающих узлов предъявляет высокие требования к общей координации системы. Это означает, что в таких системах необходимо обеспечить достаточно простой и эффективный обмен данными, обнаружение и устранение конфликтных ситуаций при одновременной попытке нескольких устройств воспользоваться каким-либо общим ресурсом.

В-четвертых, низкая себестоимость и малые габариты *MPS*, в сочетании с достаточно большой вычислительной мощностью, позволяют устанавливать их

непосредственно возле управляемого объекта, создавая территориально достаточно большие (протяженные) системы. Для таких систем взаимодальность компонентов *DCS* может достигать от нескольких единиц до нескольких сот метров. Все это накладывает определенные требования на систему связи между компонентами. Иногда встречаются случаи, когда производительности одного *МР* или *МС* не хватает для решения локальной задачи управления объектом. В таких случаях непосредственно в рамках одной *MPS* может быть организована сосредоточенная многопроцессорная система, ориентированная на решение определенной задачи.

Таким образом, как было показано выше, важной особенностью децентрализованных систем управления является не столько рассредоточенное расположение аппаратуры, сколько разделение всей функции обработки и управления на несколько слабосвязанных функций и распределение этих функций между несколькими *MPS*.

Итак, основными причинами появления децентрализованных систем можно считать:

- снижение стоимости *MPS*;
- более простую аппаратную и программную реализацию отдельных функций *DCS*;
- интерактивный характер многих задач децентрализованных систем [23];
- предпочтительность расположения средств обработки данных на основе *MPS* непосредственно возле рассредоточенных в пространстве источников и приемников данных (объектов управления или измерений);
- вследствие недостаточной мощности *МР* (*МС*) может возникнуть необходимость в создании локальных централизованных *ММPS* или в обработке данных одного объекта с помощью нескольких децентрализованных *MPS*;
- необходимость коллективного использования дорогостоящего периферийного оборудования.

Из вышесказанного следует, что главенствующую роль при создании *DCS* играет организация взаимодействия и каналы связи между децентрализованными *MPS*.

Следует отметить, что само понятие «распределенная система управления», несмотря на почти 30-летний стаж существования, — недостаточно устоявшееся и порой противоречивое. Описанные в современной специальной литературе распределенные системы очень сильно отличаются друг от друга и по принципам организации, и по другим признакам. Как правило, такие системы создаются при наличии достаточно сложного объекта управления, как следствие, субъективно в качестве главных выделяются разные характеристики объекта и неоднозначно трактуются системы управления. Наиболее часто подчеркиваются следующие характеристики *DCS*: высокая вычислительная мощность, эффективность организации, быстрота реакции, высокая пропускная способность каналов связи, высокая надежность, живучесть, малая стоимость, модульная организация, легкость наращивания, гибкость структуры; совместное использование ресурсов; хорошее поведение при перегрузках и т. п.

На первом этапе создания *DCS* в научной литературе отмечались следующие характерные признаки:

1. Наличие в системе множества процессоров (*MP* или *MC*) или *MPS*, а также других элементов системы, которые собственно и образуют собой распределенные ресурсы системы.

2. Территориальное рассредоточение этих элементов системы (ресурсов).

3. Объединение элементов системы через общие линии связи (каналы) и протоколы.

4. Наличие разбиения системы на иерархические уровни. Наличие аппаратуры и программного обеспечения *SW (Software)* верхнего уровня, которые координируют работу всей системы. Наличие программного обеспечения собственно *MC (MP) — FW (Firmware)* в элементах системы нижнего уровня.

5. Наличие простой интерактивной системы взаимодействия пользователя с системой (наиболее часто — с элементом верхнего уровня), аналогичной взаимодействию с централизованной системой.

6. Время, требуемое на обмен сообщениями, должно быть одного порядка со временем между событиями процесса, и им нельзя пренебрегать.

Раньше считалось, что если время обмена сообщениями много меньше времени между событиями процесса, то первым можно пренебречь и, соответственно, признак 6 не удовлетворяется, следовательно, распределенной системы обработки данных нет [12]. Следует отметить, что на практике все перечисленные признаки удовлетворяются довольно редко, хотя системы и называются *DCS*. Например, в некоторых современных системах используются высокоскоростные каналы передачи данных, а интенсивность обмена между элементами несоизмеримо ниже, чем интервалы между событиями процесса вследствие высокой производительности *MPS* и значительного объема первичной обработки данных в них. В соответствии с приведенными признаками такая система не могла бы называться *DCS*, т. к. не выполняется признак 6, хотя на самом деле таковой является. Как будет показано ниже, в настоящее время существует более лояльный подход к определению *DCS*. В заключение следует отметить, что указанные выше признаки характерны не только для *DCS*, но и для многих других рассредоточенных систем, в том числе, например, и для глобальных сетей *CS*.

Основными отличиями *DCS*, например, от глобальных систем являются:

1. В состав глобальных сетей входят большие *CS*, а в состав *DCS*, как правило, входят только *MPS*.

2. Обмен сообщениями в глобальных сетях происходит в режиме коммутации пакетов с длинами сообщений — несколько тысяч байтов, а в *DCS* длина сообщений намного короче и редко достигает тысячи байт.

3. В *DCS* используются линии связи с длиной, редко превышающей несколько километров, а в глобальных сетях длина канала может достигать десятков и сотен километров.

4. Многие *DCS* имеют упрощенный протокол передачи и механизм обнаружения и устранения конфликтов.

5. Связь между элементами *DCS* обычно производится в режиме коммутации каналов.

Наиболее распространенные структуры. В современных DCS особенно важным является система передачи данных между отдельными MPS. Это обусловлено, в первую очередь, стоимостью аппаратуры и собственно каналов передачи данных. Наиболее распространенные варианты структур децентрализованных систем показаны на рис. 1.1 [12].

На рисунке квадратами обозначены MPS, а кружками — устройства управления передачей данных. Рис. 1.1 иллюстрирует соединения: а) «звезда»; б) замкнутое кольцо с централизованным управлением; в) замкнутое кольцо с децентрализованным управлением; д) разомкнутое кольцо (шина) с централизованным управлением; е) разомкнутое кольцо (шина) с децентрализованным управлением.

Топологическая децентрализация MPS предполагает пространственное распределение датчиков, регуляторов и других исполнительных устройств, а также MC или другой аппаратуры обработки данных. Для обеспечения оптимальной топологической децентрализации число и места установки периферийных узлов выбираются так, чтобы минимизировать суммарную длину кабеля и состав аппаратуры связи. Обычно координирующим элементом является один из MPS или персональный компьютер (PC — *Personal Computer*). Периферийные элементы называются обычно станциями (*Stations*).

Наиболее часто используются: звездная (радиальная), кольцевая и шинная организации, а также смешанная организация, где одновременно присутствуют различные типы связи.

В радиальной (звездоподобной) структуре (рис. 1.1, а) периферийные станции обмениваются друг с другом только через центральный или координирующий элемент MPS. Программное обеспечение станций весьма просто за счет упрощения его в координирующем элементе MPS. Кроме того, такая организация, особенно при большом количестве станций, значительно усложняет реализацию и стоимость аппаратуры связи, т. к. требует использования в центральном узле коммутаторов, концентраторов и т. п.

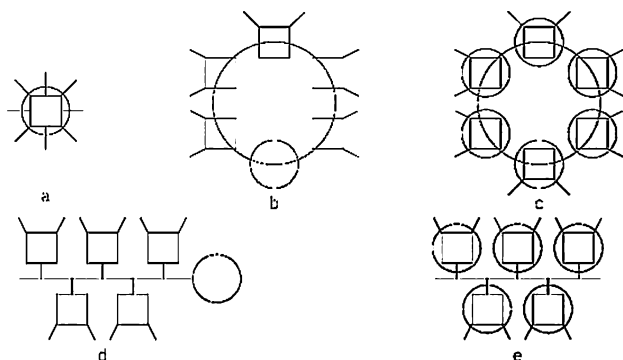


Рис. 1.1. Наиболее распространенные варианты соединения элементов децентрализованных систем

Кольцевая организация (рис. 1.1, б и рис. 1.1, с) упрощает взаимодействия между элементами станции и сокращает длину линии связи, но аппаратная реализация и программное обеспечение *FW* также достаточно сложны из-за особенностей протоколов взаимодействия.

Наибольшей популярностью в настоящее время пользуется соединение типа шина (разомкнутое кольцо). В специальной литературе опубликованы результаты анализа экономической целесообразности применения централизованных и децентрализованных систем управления шиной. Показано, что централизация систем управления экономически оправдана при большой степени территориальной сосредоточенности точек контроля и управления либо при малом их числе. В остальных случаях выгоднее децентрализация управления [12]. Кроме того, *DCS* с шинной организацией и централизованным доступом имеет минимальную длину линий связи и один координирующий *PC*, что очень выгодно при построении малых технологических и специализированных систем [14, 24—27].

Следует отметить, что в ранних *DCS* связь между станциями осуществлялась с помощью различных параллельных многопроводных интерфейсов. Такие системы связи имели высокую стоимость и по мере развития сетевых технологий и протоколов передачи данных были вытеснены системами передачи данных на основе последовательных интерфейсов. В настоящее время *DCS* практически всегда проектируются на основе *LAN* с использованием современных сетевых технологий и протоколов.

1.3. Основные понятия о локальных вычислительных сетях

Стандартизация. Локальная вычислительная сеть (*LAN*) представляет собой, в общем случае, коммуникационную систему, которая принадлежит одной организации и позволяет при помощи единой передающей среды общаться друг с другом однотипным или разнородным средствам вычислительной и микропроцессорной техники [18, 28]. Довольно большой промежуток времени, начиная с 1967 года по начало 80-х годов, *LAN* разрабатывались достаточно бессистемно, хотя отдельные фирмы и создавали ряд стандартов на разработанные ими сети. В 1980 году при *IEEE* был создан комитет *802* по стандартизации *LAN*. Стандартизацией сетей больших машин занимался в то время комитет *X3T9.5* при институте *ANSI*.

В результате работ по стандартизации были созданы ряд основополагающих стандартов, приведенных в табл. 1.1.

Таблица 1.1

	«Введение в LAN»
IEEE 802.1	
IEEE 802.2; ISO 8802.2	«Управление логическим звеном» (уровень данных)
IEEE 802.3; ISO 8802.3; ECMA-80/81/82	«Шина со случайным доступом» (физический уровень)
IEEE 802-4; ISO 8802.4; ECMA-00	«Шина с маркерным доступом» (физический уровень)
IEEE 802.5; ISO 8802.5; ECMA-89	«Кольцо с маркерным доступом» (физический уровень)
ISO 8802.7	«Кольцо с тактированным доступом» (физический уровень)

Однако на этом развитие новых типов *LAN* и соответствующих стандартов не остановилось и продолжается до сих пор. Кроме того, очень многие современные локальные сети разрабатываются с нестрогим соответствием приведенным стандартам.

Классификация LAN. К настоящему времени существует достаточно много вариантов классификации *LAN* [29—34]. В качестве классификационных признаков используются: сфера применения, назначение, размеры, трафик, топология, физическая среда, метод доступа к среде и т. д. Мы остановимся только на некоторых, наиболее интересных для нашей темы, классификационных признаках.

По размерам (количеству станций) *LAN* подразделяются на:

- малые *LAN* — до 16—20 станций;
- средние *LAN* — до 32—128 станций;
- большие *LAN* — до 500 станций;
- очень большие *LAN* — свыше 500 станций.

По признаку используемой физической среды *LAN* подразделяются на выделенные с:

- витой парой;
- многожильным кабелем;
- коаксиальным кабелем;
- силовым кабелем;
- волоконно-оптическим кабелем;
- радиоканалом;
- инфракрасным каналом и микроволновым каналом.

По признаку топологии *LAN* подразделяются на:

- шинную;
- кольцевую;
- звездообразную;
- петлевую;
- древовидную;
- смешанную;
- полносвязную.

Важнейшим классификационным признаком *LAN* является сфера применения (или функциональная ориентация). По этому признаку *LAN* подразделяются на сети для: автоматизации управления технологическими и другими производственными процессами; автоматизации делопроизводства; сбора и передачи данных; создания информационных систем и других автоматизированных систем; специальных применений; автоматизации научных исследований; автоматизации домашнего хозяйства и т. д.

От функционального назначения сети зависят алгоритмы ее работы, состав включаемого (поддерживаемого) оборудования, используемые интерфейсы, выбор среды передачи данных, форматы передаваемых данных, скорости передачи данных, метод доступа станций к сети и многое другое. Некоторые из вышеперечисленных функционально-ориентированных *LAN* должны работать в режиме

реального времени (т. е. должно гарантироваться время доступа к сети или время выполнения отдельных операций). Режим реального времени необходим при управлении технологическими или производственными процессами, в некоторых системах сбора данных, в системах для спецприменений, в системах автоматизации научных исследований. В других *LAN* временной параметр не является столь критичным, например в системах автоматизации делопроизводства, в информационных системах. Значительно зависит от функционального назначения и состав оборудования. Например, в управляющих системах редко возникает необходимость в подключении к сети стандартной компьютерной периферии и, наоборот, такое требование почти всегда присутствует при создании информационных систем или систем автоматизации делопроизводства.

Очевидно, что и длина передаваемых сообщений также сильно зависит от назначения системы, так, в управляющих системах длина сообщений обычно невелика, а в информационных — наоборот, значительна.

Большинство разработанных и стандартизованных типов сетей относятся к средним и большим. Это, как правило, информационные, делопроизводственные и другие сети общего назначения. Лишь небольшое число существующих типов сетей можно отнести к малым сетям (*SAN* — *Small Area Networks*). По приведенной выше классификации под этот термин подпадают малые (до 16—20 станций) и средние (до 32—128 станций) локальные сети.

Малые локальные сети характеризуются меньшей сложностью, стоимостью и иногда функциональной возможностью, чем обычные (большие) локальные сети. Малые локальные сети наиболее целесообразны для небольших производств, учреждений, предприятий, лабораторий, учебных заведений и контор, когда средние и большие сети избыточны, дороги и неэкономичны. Еще чаще они разрабатываются для автоматизации определенного типа территориально рассредоточенного оборудования (в производстве, научных исследованиях, домашнем хозяйстве, изделиях спецтехники, сложных измерительных системах и комплексах) [12—14, 17, 34—39]. Фактически именно *SAN*, как правило, работают в режиме реального времени, именно они являются не чем иным, как распределенными системами управления (*DCS*), именно такие системы в последнее время наиболее часто называют системами малой автоматизации.

Очевидной отличительной чертой всех *LAN* (включая *SAN*) является наличие непосредственной физической связи между абонентами сети. Другая картина наблюдается в глобальных сетях. По этой причине в глобальных сетях наиболее важной проблемой является проблема маршрутизации сообщений, а локальных сетях — проблема совместного использования общего канала передачи данных.

По методам доступа абонентов к общему каналу локальные сети разделяются на сети со случайным доступом, с детерминированным доступом, с централизованным управлением и с комбинированным доступами [24, 26, 30, 34].

В сетях со случайным доступом все абоненты равноправны и могут выходить на передачу в любое время. Это приводит к возможности появления конфликтов между абонентами при одновременной попытке нескольких из них обратиться к каналу (в силу ошибок программного обеспечения, аппаратных сбоев, помех, аварийных ситуаций). Иными словами, абоненты должны состязаться

за доступ к каналу. Поэтому метод случайного доступа иногда называют методом «состязаний». При этом методе возможно наложение в канале нескольких передач. В настоящее время существует целый ряд способов и алгоритмов минимизации конфликтных ситуаций (коллизий) и механизмов их устранения. Достоинством сетей этого типа является простота управления. Эти сети имеют достаточно широкое распространение. Недостатками сетей со случайным доступом являются: отсутствие гарантии времени доставки сообщения или команды, рост конфликтных ситуаций с увеличением интенсивности потока информации в канале и трудность реализации приоритетного доступа абонента.

От указанных недостатков свободны сети с детерминированным доступом к общему каналу. В этих сетях общий канал распределяется между абонентами в детерминированные моменты времени по некоторым алгоритмам централизованным или распределенным способом. Алгоритм обычно основывается на временном разделении возможности доступа каждого из абонентов сети, т. е. на формировании временных «окон». При этом в каждом временном «окне» канал используется только парой связывающихся друг с другом абонентов. К такому методу доступа относится широко распространенный на практике метод последовательных передач («эстафетный», «жезлов», «маркеров»), при котором доступ к общему каналу предоставляется только тому абоненту, у которого в данный момент находится «маркер». После завершения сеанса связи держатель «маркера» передает его следующей станции. Недостатком этого метода является сложность разработки эффективного алгоритма захвата канала и передачи «маркера». В системах реального времени маркерный метод доступа имеет преимущество перед методом «состязаний». Это связано с тем, что механизм обнаружения и устранения конфликтов вносит в LAN со случайным доступом дополнительные задержки, что в таких системах нежелательно.

Централизованный способ управления каналом используется преимущественно в SAN и некоторых специализированных системах. Он характерен тем, что в сети выделяется один главный узел, который часто называют «мастером» или «диспетчером». Остальные узлы сети имеют статус «ведомого» (пассивный статус). Как правило, они не имеют права по своей инициативе осуществлять доступ к сети, за редким исключением аварийных ситуаций. Диспетчер-узел управляет всеми передачами в сети и контролирует появление коллизий. Достоинствами сетей с централизованным управлением доступом является максимальное упрощение программного и аппаратного обеспечения ведомых узлов. Недостатком, соответственно, — усложнение программного обеспечения диспетчера.

В некоторых сетях используется комбинированный метод. Например, при малых потоках информации используется случайный доступ, а при больших — один из вариантов детерминированного доступа. В некоторых типах SAN, например командно-информационных, комбинируется метод централизованного управления (в фазе передачи команды) и случайный доступ.

В SAN наиболее часто используется канал со случайным доступом. Существует несколько методов (правил) управления каналом со случайным доступом:

- «асинхронный» базовый метод;
- с синхронизацией путем тактирования моментов начала передач;

- контроль (прослушивание) занятости канала перед началом передач, который иногда называют правилом «слушай, прежде чем говорить»;
- контроль (прослушивание) канала во время передачи, который также называют «слушай, пока говоришь».

Эти методы (правила) могут использоваться либо по отдельности, либо в комбинации.

Самым простым по реализации является базовый «асинхронный» метод, однако он обеспечивает самый низкий коэффициент использования канала, не превышающий 18 %. При этом методе узлы сети могут осуществлять доступ к сети в произвольные моменты времени, соответственно даже при небольшом количестве узлов очень велика вероятность наложения нескольких передач. Следует отметить, что информация в локальных сетях передается блоками фиксированной или переменной длины, которые имеют детерминированную структуру и наиболее часто называются фреймами (*frame*) или кадрами. Принимающий узел проверяет принятый фрейм обычно по контролирующей последовательности или контрольной сумме, передаваемым в конце фрейма. В случае безошибочного приема фрейма принимающий узел отправляет передающему узлу подтверждение правильности приема в виде фрейма «квитанции». В случае наличия искажений данных в принятом фрейме передается фрейм ошибки. Время передачи ответного фрейма обычно ограничено. Если передающий узел не получил ответный фрейм в отведенное время, он производит еще некоторое количество попыток передачи фреймов с данными. В таких сетях каждый узел имеет свой адрес, для него определено время задержки передачи ответа и количество попыток. Если ни одна из отведенных попыток не завершилась успешно, виртуальный канал объявляется дефектным и закрывается [26].

Второй метод позволяет повысить коэффициент использования примерно в два раза и достичь 37 %. Метод заключается в том, что в канал через определенные равные промежутки времени передаются тактовые импульсы, а любой узел может начинать передачу только в момент появления этого импульса. Во всем остальном принцип обмена по сети такой же, как у предыдущего метода.

Использование третьего правила «слушай, прежде чем говорить» позволяет достичь примерно 80 %. Метод (правило) основывается на том, что узел, желающий осуществить доступ к каналу, сперва контролирует наличие передачи в канале, и если «чужая» передача обнаружена, ожидает ее завершения. Конфликт фреймов в этом случае может произойти только тогда, если два (или несколько) узлов одновременно, не обнаружив передачи в канале, начнут свои передачи. Недостатком этого метода является то, что если несколько узлов начали свои передачи, они их продолжают до конца, т. к. они не контролируют искажения информации во время передачи.

Этот недостаток устраняется при четвертом методе, который отличается от третьего тем, что собственная передача фрейма контролируется самим узлом-передатчиком в момент передачи. Если узел обнаруживает искажение передачи, он ее прерывает и пытается повторить снова.

Комбинация двух последних методов чрезвычайно эффективна и позволяет использовать 95—98 % пропускной способности канала.

Многие фирмы, разработавшие *LAN*, предлагают свои спецификации протоколов сети в качестве стандартов.

Протоколы сети указывают, что и как должны делать узлы сети для осуществления связи. В частности, протоколы определяют такую важную характеристику сети, как формат кадра передаваемого сообщения. Обычно фрейм имеет преамбулу (признак начала фрейма) и ряд полей (байтов, слов и т. п.), содержащих информацию об адресе передатчика, адресе приемника, длине фрейма, иногда скорости передачи, собственно данных и контрольной сумме. Длины фреймов у различных сетей сильно отличаются. От длины фрейма и его структуры достаточно сильно зависит такой важный параметр, как максимальная производительность сети, под которой обычно понимается предельно возможное количество полных циклов обмена в сети без учета возможных коллизий в единицу времени.

В табл. 1.2 приведены форматы фреймов некоторых из первых известных *LAN*.

Таблица 1.2

Название сети	Преамбула, байт	Адрес приемника, байт	Адрес источника, байт	Поле команды, байт	Поле типа фрейма, байт	Поле длины фрейма, байт	Поле данных, байт	Контрольная сумма, байт
NBS (1975)	1	1	8	1–2	–	*	0–128	2
Ethernet (1976)	1	1	8	–	–	–	0–500	2
Xerox (1980)	4	6	48	2	*	–	46–1500	2
DIX (1980)	8	6	48	2	–	*	46–1500	4
Proway (1981)	1	1	8	1	*	+	1–20	2
IEEE Proj 802 (1981)	1–4	1–2	8–56	1–7	1–7	*	0–20	4
DIX (1981)	4	8	64	–	+	*	46–1024	4

Приведенный список постоянно пополняется. Разрабатываются все новые и новые сети. К настоящему моменту в мире насчитывается уже несколько тысяч типов *LAN* и интерфейсов *DCS*. Достаточно глубоко описаны такие *LAN*, как *ETHERNET*, *RING*, *FIBRENET*, *FIELDBUS*, *PROFIBUS*, *LONWORKS*. Также хорошо описаны интерфейсы распределенных систем управления *DCS*, таких как: *PROWAY*, *IPM*, *BITBUS*, *MAP*, *MIL-1553*, *MIL-1773*, *RS422*, *RS485*, *CAN*.

1.4. Основные понятия о системах малой автоматизации

В предыдущих разделах мы ознакомились с основными понятиями о распределенных системах управления *DCS* и малых локальных сетях *LAN*. Эти понятия достаточно близки, т. к. практически все современные *DCS* реализуются на основе малых и средних *LAN*, соответственно они имеют перекрывающиеся признаки и области применения.

Системами малой автоматизации — *SAS* называются малые распределенные системы управления или контроля (*DCS*), реализованные на основе специализированных малых и средних локальных сетей (*SAN*), ориентированные на работу в режиме реального времени (*Real Time*). Первостепенными качествами систем малой автоматизации считаются низкая себестоимость, обеспечение режима реального времени, относительно простое проектирование, модернизация и эксплуатация, относительно простое программное и аппаратное обеспечение в сочетании с достаточно высокой надежностью.

Эти системы широко применяются для автоматизации небольших технологических, измерительных, исследовательских систем (обычно территориально расположенных в пределах одного помещения), а в последнее время и для автоматизации домашнего хозяйства.

Рассмотрим основные факторы, способствующие развитию этих систем:

1. Постоянное снижение цен на персональные компьютеры (*PC*) при значительном повышении их вычислительной мощности и надежности привело к тому, что стало экономически выгодным использование *PC* на верхнем уровне управления даже очень простых *SAS*. При этом оптимально решаются проблемы визуализации, хранения и документирования данных системы.

2. Быстрое развитие высокопроизводительных и дешевых *x86*-совместимых микроконтроллеров фирм *Cygnal* и *Atmel* и семейства *AVR* фирмы *Atmel*. Эти микроконтроллеры характеризуется высокой производительностью, соизмеримой с производительностью *i386*-компьютеров, а некоторые из новых микроконтроллеров фирм *Cygnal* (семейства *F12x*) даже достигают производительности *5x86*- и *PII*-компьютеров. Эти микроконтроллеры имеют очень мощный набор периферийных узлов, большие объемы памяти программ, они ориентированы на работу в сложных условиях эксплуатации, имеют высокопроизводительные последовательные каналы передачи данных *UART*. Многие из них имеют также встроенные многоканальные аналого-цифровые преобразователи *ADC* (*Analog-to-Digital Convertor*) и несколько встроенных цифроаналоговых преобразователей *DAC* (*Digital-to-Analog Convertor*), аналоговые компараторы *CPT*, охранные таймеры *WDT* (*WatchDog Timer*), последовательные интерфейсы *SPI*, *SMBus* (*I2C*), *CAN* и т. п. Перечисленные достоинства современных микроконтроллеров позволяют создавать на их базе мощные малогабаритные универсальные локальные контроллеры управления и измерения (которые часто называют универсальными технологическими контроллерами), которые при этом имеют достаточно низкую стоимость и относительно простую аппаратную реализацию.

3. Наблюдаемая в последнее десятилетие глобальная тенденция к реструктуризации крупных производственных и научных учреждений и созданию на их основе малых производственных фирм и научно-исследовательских центров повышает актуальность создания малых систем автоматизации. А снижение или полное прекращение во многих странах объемов централизованного финансирования автоматизации производства и научных исследований приводит к тому, что *SAS*, обладающие более низкой первоначальной стоимостью и высокой гибкостью, чем большие универсальные *DCS*, становятся экономически все более выгодными и оправданными.

4. Развитие сетевых технологий и создание новых типов *SAN*, обладающих низкой стоимостью при высоких качественных показателях, специально ориентированных на работу в составе *SAS*, также способствует их быстрому развитию. Наиболее часто в современных *SAS* используются так называемые командно-информационные сети — *CI-LAN (Command — Information LAN)*. Мы подробно ознакомимся с ними в одной из следующих глав, а сейчас только поясним смысл названия. Сети *CI-LAN* характеризуются тем, что имеют в своем составе только один персональный компьютер (или мощный специализированный контроллер, оснащенный полным комплектом периферийного оборудования), и некоторое количество универсальных (или специализированных) контроллеров — станций. Сеть работает по принципу опроса. Компьютер является инициатором всех обменов в сети, т. е. «ведущим» узлом. Он передает всем станциям фрейм команды, содержащий, в том числе, и адрес одной из станций. Все станции пассивны, т. е. работают в режиме «ведомого». Они не имеют права инициализировать обмены в сети (за исключением аварийных ситуаций). Они принимают и анализируют фрейм команды, но только одна из них, адрес которой совпал с адресом фрейма, выполняет предписанные командным фреймом действия и либо сообщает компьютеру о выполнении команды, либо передает в него полученные данные. Таким образом, полный цикл обмена состоит из команды и ответной информации, отсюда и термин — командно-информационные сети. Этот тип *SAN* обладает всеми требуемыми качествами для создания малых систем автоматизации и находит в последнее время широкое применение.

Типовая структура современных систем малой автоматизации, построенных с использованием *CI-LAN*, приведена на рис. 1.2.

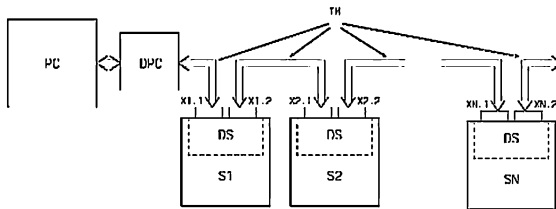


Рис. 1.2. Типовая структура современных систем малой автоматизации

Единственный в *SAS* персональный компьютер (*PC*) выполняет роль единственного в системе генератора заданий и получателя информации. Только он имеет право инициировать обмены информацией по сети. Кроме того, он получает всю предварительно обработанную информацию от N периферийных станций $S1, S2, \dots, SN$ (*Station 1...N*), которые не имеют право самостоятельного доступа к сети. Они только имеют право отвечать на поступившую команду. Исключением может быть случай аварийной ситуации в станции (станциях) или в обслуживаемой подсистеме объекта регулирования. Станции между собой соединяются с помощью кабелей с разъемами (обычно типа DB9) через параллельно соединенные пары разъемов $X1, X2, \dots, XN$ на каждой из станций $S1,$

S_2, \dots, S_N . Каждая из станций в сети имеет индивидуальный номер, устанавливаемый на модификаторах (переключателях или перемычках) или записанный во Flash-памяти микроконтроллера. Соединение персонального компьютера может производиться к одной из станций непосредственно или через специальный узел диспетчера персонального компьютера *DPC* (*PC Dispatcher*) в зависимости от используемого количества станций N , необходимости преобразования интерфейсов (выходного интерфейса персонального компьютера в интерфейс сети) и оптической развязки. Кабельная часть сети называется сетевой средой — *TM* (*Transmission Medium*). В каждой станции имеется узел, согласующий интерфейс микроконтроллера станции (обычно *UART*) с сетевой средой. Он обычно называется диспетчером станции *DS* (*Station Dispatcher*) или *MGT* (*Station Management*).

Современные системы малой автоматизации *SAS* по количеству используемых станций подразделяются на три группы: малые (до 16—32 станций), средние (до 128—256 станций) и большие (свыше 256 станций) *SAS*. Большие *SAS* в виде сети одного уровня выполняются редко, ввиду ограниченной нагрузочной способности существующих последовательных интерфейсов. Чаще всего большие *SAS* либо являются сетью одного уровня с комбинацией интерфейсов, либо выполняются в виде многоуровневых сетей.

Условно структуру *SAS* можно разбить на два уровня. Первый верхний уровень — уровень организации командно-информационных сетей *CI-LAN*. Второй нижний уровень — уровень организации станций, в качестве которых используются так называемые универсальные технологические контроллеры. Каждому из этих уровней будет посвящена отдельная глава настоящей книги. Но прежде чем рассмотреть более подробно эти вопросы, давайте познакомимся с основными тенденциями развития и требованиями, которые предъявляются к современным *SAS*.

Практически во всех работах, посвященных архитектуре современных малых систем автоматизации производства и научных исследований, в качестве основных отмечаются тенденции *децентрализации, минимальной стоимости, простоты установки и эксплуатации, максимальной эффективности*. Рассмотрим эти понятия.

Современное понятие *децентрализации* системы предполагает и территориальное рассредоточение вычислительной системы, и разделение функции системы (измерений, первичной обработки и управления) на ряд слабосвязанных функций, и распределение этих функций между несколькими децентрализованными вычислителями. Таким образом, современные *децентрализованные* системы являются системами с множеством потоков команд и данных *MIMD* (*Multiply Instruction Multiply Data*). С другой стороны, каждая из станций (сосредоточенный вычислитель) также может выполнять несколько слабосвязанных функций одновременно и состоять из нескольких слабосвязанных вычислителей, т. е. быть *мультимикроспроцессорной* (*MMPs*).

Создание *децентрализованных* систем автоматизации производства и научных исследований в условиях малых производственных фирм и научно-исследовательских центров определяет в качестве одного из основных требований — требование *минимальной стоимости*.

Системы, ориентированные на работу в условиях малых производственных фирм и научно-исследовательских центров, должны удовлетворять требованиям *простоты установки и эксплуатации*, а также обеспечивать достаточно простую *модернизацию*, т. е. обновление.

Под понятием *максимальной эффективности*, как правило, понимается соотношение производительности/цена, однако в последнее время это понятие часто объединяется с понятием *минимальной избыточности* системы. При этом под *избыточностью* понимают как *аппаратную избыточность* (неиспользуемые узлы аппаратуры или излишние сигналы управления), так и *временную избыточность* (например, передача излишней информации по сетям).

Теперь рассмотрим основные свойства и тенденции развития современных сосредоточенных систем SAS, т. е. сетевых станций (или универсальных технологических контроллеров).

Важнейшим принципом построения современных сложных универсальных микроконтроллерных (микропроцессорных) систем является *магистрально-модульный принцип*.

Модульность — свойство территориально сосредоточенных элементов системы (станций), означающее конструктивное выполнение всех узлов в виде набора функционально законченных автономных модулей, имеющих одинаковый интерфейс с общей *магистралью или шиной*.

Понятие *модульности* системы тесно связано с понятием *магистральной (или шины)* системы, т. е. подразумевается, что все модули связаны между собой через одинаковые разъемы с параллельно соединенными контактами — *магистраль (шину)*. Каждый контакт и соединяющая их линия имеют определенное назначение и наименование. Временные и электрические параметры сигналов обычно также определены правилами, созданными для конкретного типа *магистральной*.

К настоящему времени создано более 250 различных магистрально-модульных систем [11, 16, 32, 33, 42—48], наиболее известные из которых *EUROBUS*, *VME-BUS*, *FASTBUS*, *MULTIBUS-I/II*, *P-896*, *CAMAC*, однако к настоящему времени часть из них уже морально устарела, другие оказались избыточными и чрезвычайно дорогими. К тому же ставшие *de facto* во всем мире промышленным стандартом микроконтроллеры семейства MCS-51 и аппаратно совместимые с ним AVR, а также микропроцессоры MCS-86/88 и MCS-80/85, которые наиболее часто используются сейчас для автоматизации производства и научных исследований, плохо совместимы с большинством из них. Как правило, в современных системах наиболее часто используются модернизированные интерфейсы *MULTIBUS-I/II*, *I41* [45, 48]. Они наиболее просты и наиболее подходят для построения современных микроконтроллерных систем. Модернизация интерфейсов магистральной наиболее часто касается исключения или замены линий (сигналов), предназначенных для многопроцессорного арбитража, питания и т. д.

Модульность позволяет легко заменять вышедшие из строя модули, т. е. повышается *ремонтпригодность* системы.

Гибкость — свойство *модульных* систем легко изменять структуру изделий за счет различного сочетания модулей. Кроме того, под *гибкостью* системы понимается возможность осуществления модернизации системы за счет независимой модернизации каждого из модулей. Облегчается разработка новых систем и увеличивается срок морального старения за счет применения комбинации новых и уже существующих модулей, повышаются регулярность структуры и, как следствие, контрольно-диагностические качества, обеспечивается простота переориентации системы на другой объект регулирования или исследования.

Наращиваемость — свойство системы, заключающееся в возможности простого увеличения или уменьшения количества модулей в станции или станциях в сети.

Многофункциональность — свойство станции, означающее, что она не имеет жесткой специализации и привязки к объекту регулирования или исследования и, соответственно, может выполнять разные функции.

«*Интеллектуальность*» — свойство большинства модулей современных станций, связанное с наличием в них микроконтроллеров или микропроцессоров, работающих по своей индивидуальной программе, уровень интеллектуальности можно легко увеличивать в уже работающей системе путем модификации программного обеспечения модулей.

Конструктивное единство — понятие, означающее конструктивное единоеобразное исполнение всех модулей станции. Доминирующим стандартом в настоящее время является *EUROCARD (IEC48D, 1980)*, позволяющий создавать ряд вариантов плат как комбинацию размеров ширины (160 и 220 мм) и высоты (100 и 233,53 мм). Наиболее часто используются платы E2 с размерами 220 × 233,53 мм² (*MULTIBUS-II*) и E4 с размерами 100 × 220 мм² (*EUROBUS, ESONE*), а также 100 × 100 мм².

Мультимагистральность — понятие, означающее использование в одной станции одновременно несколько различных магистралей. Такое техническое решение широко используется и в современных РС, и во многих магистрально-модульных системах. Наиболее широкое распространение в *SAS* получили двухшинные системы, на каждой из шин которых расположены слабосвязанные микропроцессоры и микроконтроллеры различной мощности и разрядности, причем магистрали связаны через общий двухходовый ресурс, чаще всего оперативную память.

Безусловным принципом при построении проблемно-ориентированных систем является принцип *достаточности производительности*. Достижение необходимой производительности обеспечивается, как правило, созданием многопроцессорных систем в одной станции, причем кроме обеспечения достаточной производительности за счет параллельного выполнения независимых задач или частей одной задачи достигается повышение надежности путем обеспечения свойства «*постепенной деградации*» системы при отказах отдельных процессоров за счет перераспределения задач, возрастает также *гибкость* системы.

Совместимость ПО — требование ко всем микропроцессорам и контроллерам системы, соблюдение которого позволяет использовать богатый арсенал созданных прикладных программ. В настоящее время, когда в одной системе могут

одновременно присутствовать вычислительные элементы, построенные на различных семействах процессоров (*Pentium*, *MCS-86/88*, *MCS-80/85*, *MCS-51*, *AVR* и т. д.), требование реализуется за счет написания программного обеспечения всех элементов системы на языке высокого уровня (как правило, языке «С»).

Развитие *магистрально-модульных* систем предполагает обеспечение принципа *максимальной универсализации*, состоящего в том, что каждый вводимый в систему модуль должен обеспечивать построение системы, удовлетворяющей требованиям максимального числа применений.

Кроме того, очевидно выполнение принципа построения магистральных систем — *принципа асинхронности и минимальности количества сигналов управления*, упрощающего протокол обмена между модулями, повышающего надежность связи и быстродействие обмена.

При построении распределенных систем с небольшими линейными размерами, измеряющих и обрабатывающих большие массивы информации, часто необходимо обеспечивать местную *визуализацию* оперативной информации, что значительно упрощает работу с системой.

Требования к сетевой архитектуре, организации, среде и протоколам передачи данных также обусловлены современной технической и экономической ситуацией.

Быстрое снижение цен на персональные компьютеры при значительном повышении их вычислительной мощности и надежности, то обстоятельство, что самыми дорогостоящими в *PC* по-прежнему остаются периферийные устройства, а также требование минимальной стоимости системы предопределяют еще одну важную особенность современных малых систем автоматизации производства и научных исследований — *информационно-командный принцип построения LAN*.

В соответствии с этим принципом единственный *PC*, включенный в *LAN*, играет роль генератора заданий и получателя данных, осуществляет вторичную обработку, систематизацию, сохранение и визуализацию данных. Периферийные станции осуществляют съем аналоговой и дискретной информации с различных датчиков, ее предварительную обработку, управляют различными внешними объектами, производят тестирование своих подсистем и передают полученную информацию в *PC*.

С целью снижения стоимости системы и обеспечения взаимозаменяемости используются *PC* в стандартной конфигурации, а связь осуществляется через свободный интерфейс *RS232C* (*COM1* или *COM2*), имеющийся на любом стандартном компьютере.

Еще одним требованием к современным *LAN*, работающим в условиях сильных электромагнитных и электростатических полей, является наличие гальванической (чаще всего оптической) развязки на напряжения более 500 В всех периферийных станций.

С целью обеспечения минимизации сетевого *фрейма* (кадра передачи данных) и нагрузки на сетевой интерфейс разработчики прибегают к ограничению количества периферийных станций сети и количества команд одной станции. С другой стороны, с целью упрощения алгоритмов работы одной станции иногда

разработчики предпочитают разделять одну локальную станцию на две и более, территориально расположенные в одном месте.

Наиболее дорогостоящим и трудоемким элементом современных *LAN* является сетевая среда, т. е. кабель, связывающий станции между собой и с *PC*. Поэтому в системах, предполагающих частую модернизацию, желательно использовать стандартные кабели с разъемами. При этом желательно иметь возможность оперативной замены кабеля и учитывать, что его длина влияет на скорость передачи.

Описанные понятия и тенденции широко применяются при проектировании современных *SAS*. Более подробно вопросы проектирования *SAS* мы рассмотрим в следующих главах.

2. Командно-информационные сети

2.1. Основные понятия

Командно-информационные сети широко используются в современных системах малой автоматизации SAS.

Командно-информационными сетями (*CI-LAN* — *Command-Information LAN*) называются малые и средние специализированные локальные сети с шинной топологией и централизованным управлением, ориентированные на работу в режиме реального времени.

Все узлы сети называются станциями (*Station*). Централизованное управление шиной осуществляется ведущей станцией, в роли которой, как правило, выступает единственный в сети персональный компьютер (*PC*) или мощный специализированный контроллер, оснащенный полным комплектом периферийного оборудования. Все остальные станции сети называются периферийными станциями (*PS* — *Peripheral Stations*).

Типовая структура *CI-LAN* приведена на рис. 1.2. Единственный персональный компьютер сети является генератором заданий и получателем информации. Только он имеет право инициировать обмены информацией по сети. Он работает под управлением основной программы, которая в заданные моменты времени выдает определенные команды периферийным станциям. Команды от ведущей станции и ответы от периферийных станций передаются в виде фреймов (*Frame*) — детерминированной последовательности байтов, содержащих байт(ы) синхронизации — преамбулу (*Preamble*), байты адреса станции, команды, данных и контроля. Структуры фреймов будут рассмотрены позднее. Каждая из N периферийных станций S_1, S_2, \dots, S_N (*Station 1...N*) имеет уникальный адрес, устанавливаемый модификатором адреса (переключатель или перемычки) или записанный во Flash-памяти микроконтроллера. При передаче очередной команды (в виде фрейма команды) ведущей станцией (далее просто *PC*), все периферийные станции принимают фрейм этой команды и проверяют достоверность передачи по байту (байтам) контроля. В качестве байтов контроля обычно используется контрольная сумма — сумма всех байтов фрейма, начиная с преамбулы и заканчивая байтом, предшествующим байту контроля. В случае несовпадения переданной и рассчитанной в периферийных станциях контрольной суммы станции игнорируют полученный фрейм команды и ожидают поступления очередного фрейма команды. При валидной (правильной) контрольной сумме принятого командного фрейма все станции анализируют байт адреса команды, сравнивая его со своим уникальным адресом. Одна из станций, уникальный адрес которой совпал с адресом, установленным в фрейме команды, распознает команду, исполняет ее, а остальные периферийные станции, уникальные адреса которых отличаются от переданного в фрейме адреса, игнорируют полученный фрейм и ожидают приема очередного командного фрейма. Команда может быть

различного уровня сложности. Простые команды могут требовать от периферийной станции элементарных действий, например, установить одну из линий порта ввода/вывода в определенное логическое состояние. Более сложные команды могут заставить периферийную станцию выполнить достаточно сложную процедуру получения и предварительной обработки данных. Например, произвести опрос одного из каналов аналого-цифрового преобразователя, при этом предварительно настроив канал, предварительный усилитель, счетчик усреднения, произвести несколько опросов, усреднить их, масштабировать, преобразовать в требуемую форму, вывести информацию на местную индикацию и подготовить информацию для передачи в *PC*. После завершения выполнения команды адресуемая станция формирует фрейм ответа. Он может содержать либо информацию об успешном выполнении простой команды, либо полученные и предварительно обработанные данные в случае более сложной операции, либо информацию об ошибках и сбоях. В любом случае, в фрейме ответа содержится та или иная информация. Т.е. передаваемые *PC* данные — всегда команда, а получаемые — всегда информация. Отсюда и произошло название этих сетей — командно-информационные. Получив фрейм ответа от периферийной станции, *PC* производит в случае необходимости последующую обработку данных, систематизирует, визуализирует и сохраняет их, а затем, в соответствии с главной программой сети, производит выдачу очередной команды.

Обычно в *CI-LAN* периферийные станции не имеют право самостоятельного доступа к сети. Они только отвечают на поступившую команду. Исключением может быть случай аварийной ситуации в станции (станциях) или в обслуживаемой подсистеме объекта регулирования. В этом случае в некоторых сетях допускается передача в *PC* сигнала аварии. Однако такая возможность значительно усложняет алгоритм работы системы. Чаще в *CI-LAN* используется периодический опрос всех станций с целью определения состояния каждой из них.

Станции между собой соединяются с помощью кабелей с разъемами (обычно типа DB9) через параллельно соединенные пары разъемов X_1, X_2, \dots, X_N на каждой из станций S_1, S_2, \dots, S_N . Соединение персонального компьютера может производиться к одной из станций непосредственно или через специальный узел диспетчера персонального компьютера *DPC* (*PC Dispatcher*), в зависимости от используемого количества станций N , необходимости преобразования интерфейсов (выходного интерфейса персонального компьютера в интерфейс сети) и оптической развязки. Кабельная часть сети называется сетевой средой — *TM* (*Transmission Medium*). В каждой станции имеется узел, согласующий интерфейс микроконтроллера станции (обычно UART) с сетевой средой. Он обычно называется диспетчером станции *DS* (*Station Dispatcher*) или *MGT* (*Station Management*).

Количество станций в современных *CI-LAN* одного уровня обычно не превышает 256. При этом, если количество станций в сети менее 16—32, такие *CI-LAN* называют малыми, в противном случае — средними.

Рассмотрим основные требования, предъявляемые к *CI-LAN*:

- с целью снижения стоимости системы и обеспечения взаимозаменяемости в составе *CI-LAN* используются *PC* в стандартной конфигурации с двумя

портами RS232C, один из которых, как правило, занят, а связь осуществляется через второй свободный порт;

- наиболее дорогостоящим и трудоемким элементом современных LAN является кабель передачи данных — сетевая среда (ТМ), поэтому в CI-LAN, предполагающих частую модернизацию, желательно использовать либо стандартные компьютерные кабели с двумя разъемами типа DB9, либо «витую пару», либо 4-жильные телефонные экранированные кабели;
- общая длина сетевой среды влияет на скорость передачи данных, поэтому желательно ограничивать длину LAN до минимума;
- важнейшим требованием к CI-LAN, работающим в условиях сильных электромагнитных и электростатических полей или в лабораторных условиях, является наличие гальванической развязки периферийных станций от сетевой среды (а иногда и PC от сетевой среды). В современных CI-LAN гальваническая развязка достигается с помощью элементов с оптической изоляцией — оптронов;
- с целью снижения стоимости и объема аппаратных затрат в CI-LAN в основном используются два типа интерфейсов: модифицированный интерфейс RS232C и интерфейс RS485.

Диспетчер PC (DPC) — узел, согласующий порт RS232C персонального компьютера с сетевой средой. Если CI-LAN строится на базе интерфейса RS485, то DPC оптически изолирует порт RS232C персонального компьютера и преобразует его в интерфейс RS485. В случае построения CI-LAN на базе модифицированного интерфейса RS232C узел не является обязательным и фактически оптически изолирует некоторые сигналы интерфейса RS232C.

Диспетчер станции DS преобразует сигналы интерфейса RS232C или RS485 во внутренние сигналы станции. Как правило, каждый DS имеет два одинаковых разъема (Xi.1 и Xi.2) DB9F, соединенных параллельно.

Сетевая среда в CI-LAN выполняется в виде кабелей с однотипными разъемами DB9V одинаковой либо разной длины, соединяемых последовательно. Это позволяет легко и оперативно модифицировать сеть. Малые CI-LAN можно реализовать на базе интерфейсов RS232C и RS485, средние — только на базе интерфейса RS485. Общая длина малых CI-LAN обычно не превышает 20 м, в то время как средние CI-LAN могут иметь общую длину шины до 1 км и более. Если в CI-LAN используется RS232C, то сетевая среда чаще всего состоит из стандартных компьютерных кабелей с двумя разъемами DB9V (длиной 1 м или 1,2 м). Если в CI-LAN используется RS485, то сетевая среда чаще всего выполняется из четырехжильного телефонного провода с общим экраном и изоляцией, оснащенного двумя разъемами DB9V.

Отметим также некоторые особенности работы сетей на базе интерфейсов RS232C и RS485. Напомним, что интерфейс RS232C имеет отдельные линии передачи TxD и приема RxD данных. Поскольку команды в CI-LAN передает только PC, он монополюсно использует свою линию TxD и, следовательно, на этой линии конфликты от одновременного включения нескольких передатчиков невозможны. При этом PC не имеет возможности контролировать передачу. Линия RxD используется всеми контроллерами, которые подключены к ней своими вы-

ходами TxD . В случае нормальной работы аппаратуры и программного обеспечения станций конфликтов на линии быть не должно, но при сбоях возможны конфликты вследствие одновременного подключения к линии нескольких передатчиков. Таким образом, линия RxD является линией со случайным (со стороны контроллеров) доступом и программным (со стороны PC) анализом конфликтов.

При использовании интерфейса $RS485$ данные передаются по двум проводам как в прямом, так и в обратном направлении, т. е. конфликты возможны как при передаче команд, так и приеме данных. За всем конфликтами в сети следит PC . В этом случае вся сеть является сетью со случайным доступом и программным анализом конфликтов. Кроме того, все участники такой сети (и PC , и PS) должны обеспечивать отключение своих передатчиков в перерывах между передачами и анализировать незанятость сети перед попыткой передачи и в процессе передачи.

Таким образом, информационно-командный принцип построения LAN предполагает следующие основные режимы обмена данными:

- получение любой из периферийных станций (PS) фрейма команды от PC ;
- передача в PC результатов измерений от любой станции PS (в ответ на принятую команду от PC);
- выполнение любой станцией самотестирования по команде PC с передачей результатов тестирования обратно в PC ;
- желателен аппаратный сброс всех станций по инициативе PC ;
- возможна передача в PC информации об аварийных ситуациях по инициативе любой станции PS .

2.2. Диспетчеры персонального компьютера

На диспетчеры персонального компьютера (DPC) возлагаются две основные функции: сформировать сигналы среды передачи данных (TM) и обеспечить гальваническую развязку PC . В этом разделе приведены три схемы DPC для $CI-LAN$ на базе интерфейсов $RS232C$ и $RS485$.

На рис. 2.1 представлена схема DPC для сети на базе интерфейса $RS232C$.

Входная часть схемы DPC (слева) гальванически развязана от выходной части (справа) с помощью оптронов $4N35$. Питание и управление входной части осуществляется от выходных линий $RS232C$ порта персонального компьютера. При этом предполагается, что в рабочем режиме (приема и передачи данных) на линии RTS должно быть установлено положительное напряжение, а на линии DTR — отрицательное напряжение. Величины этих напряжений определяются исполнением материнской платы персонального компьютера и могут составлять от 7 до 12 В соответствующей полярности. Положительное напряжение с линии RTS поступает на коллекторы транзисторов оптронов $D1$ и $D5$. Отрицательное напряжение с линии DTR через резистор $R5$ и диод $VD1$ создает на конденсаторе $C1$ отрицательное напряжение питания. Передача данных от компьютера осуществляется по линии TxD через резистор $R2$ и светодиод оптрона $D2$. Прием данных осуществляется по линии RxD с эмиттера транзистора оптрона $D1$. Со-

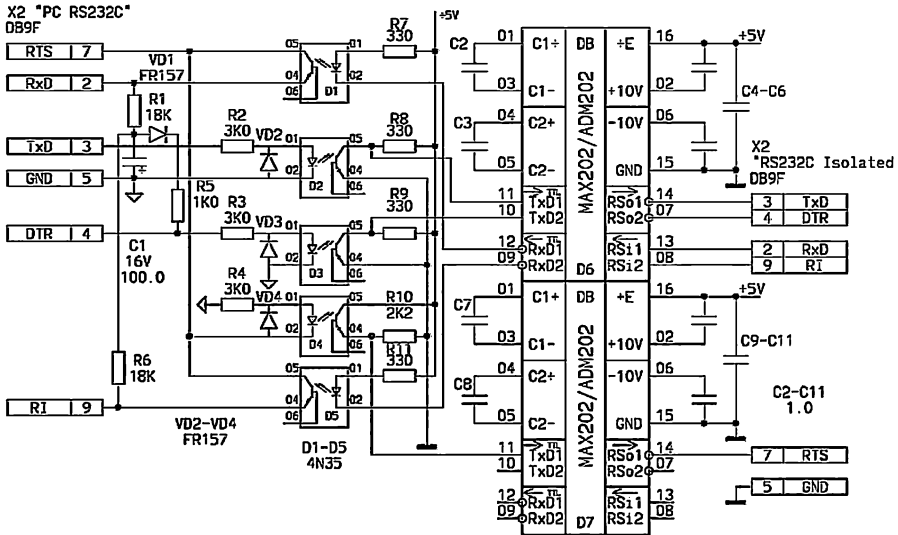


Рис. 2.1. Диспетчер персонального компьютера для сети на базе интерфейса RS232C

стояния линий *DTR* и *RTS* передаются в выходную часть через оптроны *D3* и *D4*, а состояние линии *RI* с выходной части передается в персональный компьютер через оптрон *D5*. Предполагается, что в составе *DPC* имеется стабилизированный источник питания на напряжение +5 В и рабочий ток менее 30 мА (на схеме не показан). Этот источник осуществляет питание выходных микросхем *D6* и *D7*. При использовании оптронов типа *4N35* напряжение изоляции цепей *PC* от выходной части составляет более 2,5 кВ. В выходной части в качестве формирователей уровней интерфейса *RS232C* использованы самые доступные микросхемы *MAX202* (*ADM202*, *MAX232*). Возможно, конечно, использовать и другие формирователи уровней фирмы *MAXIM* (*MAX235-MAX249*), однако это увеличит стоимость изделия, сложность и площадь печатной платы. Конденсаторы *C2—C11* должны быть не менее 1,0 мкФ (хотя изготовителем микросхем рекомендуется величина 0,1 мкФ). Это обеспечивает повышенную устойчивость работы выходных микросхем в аварийных режимах работы сети, а также при отладке контроллеров. Диоды *VD2—VD4* рекомендуется устанавливать для повышения надежности (предотвращения пробоя светодиода оптронов).

Приведенная схема *DPC* обеспечивает изоляцию и повторение состояний сигналов порта *RS232C* в статике. Однако следует заметить, что следствием оптической развязки с помощью оптронов является задержка времени переключения всех сигналов на величину более 7 мкс. Кроме того, несмотря на то что номиналы резисторов входной части оптимальны с точки зрения быстродействия и нагрузки на выходной порт персонального компьютера, максимально допустимая скорость передачи данных через описанный *DPC* составляет 28800. На больших скоростях передачи резко возрастает количество ошибок.

Как уже отмечалось выше, некоторые *CI-LAN* на базе интерфейса *RS232C* могут работать вообще без использования *DPC*, однако приведенный диспетчер всегда рекомендуется использовать в процессе отладки, чтобы защитить ваш компьютер. Для малоопытных специалистов следует напомнить, что во многих персональных компьютерах земля находится под потенциалом половины сетевого напряжения и подключение к работающему незаземленному компьютеру земли осциллографа или ее обрыв может привести к катастрофическим последствиям. При использовании описанного диспетчера *DPC* отладка и *LAN*, и отдельных контроллеров, подключенных к компьютеру, становится безопасной!

На рис. 2.2 представлена схема *DPC* для сети на базе интерфейса *RS485*. Входная часть *DPC* изображена справа и подключается к *PC* с помощью разъема *X2*. Она аналогична входной части описанной выше схемы. Однако в связи с особенностями построения *CI-LAN* на базе интерфейса *RS485* в этом *DPC* реализованы дополнительные функции: программно-аппаратного сброса всех контроллеров сети и отключения выхода диспетчера. Особенностью входной части *DPC* является то, что интерфейс *RS232C* персонального компьютера имеет только две выходные линии *RTS* и *DTR*, которые уже используются для питания входной части (как и в схеме, приведенной на рис. 2.1), при этом их уровни соответственно равны $RTS = H$ (*high*) и $DTR = L$ (*low*).

Учитывая то, что в соответствии с алгоритмом работы *CI-LAN* (описание будет приведено ниже) передатчик *PC* закрывается на достаточно короткие промежутки времени, отключение выхода передатчика осуществляется при переводе сигнала *DTR* в состояние Н. При этом положительное напряжение с линии *DTR* через резистор *R10* включает светодиод нижней оптопары *D3*, а коллектор транзистора этой оптопары переводит вход управления драйвера *DE* в состояние L, что, соответственно, переводит выходы передатчика в высокоимпедансное состояние.

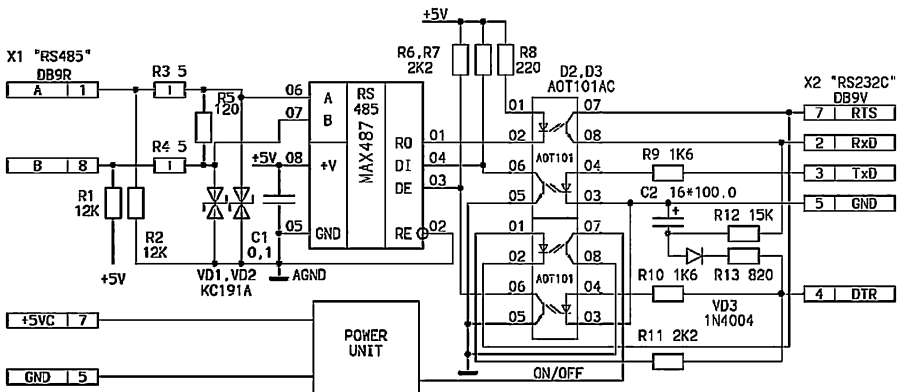


Рис. 2.2. Диспетчер персонального компьютера для сети на базе интерфейса *RS485*

Состояние $RTS = L$ и $DTR = H$ приводит к включению светодиода верхней оптопары $D3$. Коллектор ее транзистора переводит при этом в состояние L вход управляемого стабилизатора $POWER UNIT$. Его схема не приведена, однако она тривиальна. Этот стабилизатор может быть собран либо по стандартной схеме включения $KP142EH12A$, либо с использованием более мощных стабилизаторов $SD1083$ (7,5 А) или $SD1084$ (5 А) (отечественный аналог — $KP142EH22$), имеющих такой же корпус и такую же схему включения. Управляющий вход подключается к выводу 1 стабилизатора. Функции управляемого стабилизатора зависят от назначения и количества станций в $CI-LAN$. Обычно этот стабилизатор делается маломощным на напряжение +5 В (до 200 мА) и используется только для сброса всех контроллеров сети, при этом выходное напряжение поступает на входы внешнего сброса супервизоров питания контроллеров. Однако в некоторых сетях стабилизатор выполняют достаточно мощным, до 3 А на напряжение около 8 В. При этом выходное напряжение этого стабилизатора кроме функции сброса осуществляет еще и питание всех контроллеров сети. Очевидно, это возможно только при специальной схемотехнике периферийных станций и их ограниченном количестве.

Выходная часть DPC может быть реализована либо на микросхеме $MAX485$ (или ее аналогах), при этом DPC обеспечивает работу до 32 PS в составе сети, либо на микросхеме $MAX487$, — при этом DPC обеспечивает работу в сети до 128 станций. Естественно, что в этом случае и во всех контроллерах должны стоять такие же микросхемы драйверов.

Резистор $R5$ должен иметь сопротивление 120 Ом. Он должен устанавливаться в $RS485$ -сетях на двух концах TM . Т. к. согласно топологии $CI-LAN$ диспетчер PC всегда устанавливается на одном из концов TM , резистор $R5$ должен быть установлен всегда. Резисторы $R1$, $R2$ служат для «привязки» уровней линий «А» (к земле) и «В» (к питанию) для уменьшения времени переходных процессов в случае, когда все передатчики сети закрыты. Резисторы $R3$, $R4$ и стабилитроны $VD1$, $VD2$ являются простейшей защитой от импульсных помех, возникающих в достаточно протяженных сетях или в сетях, работающих в условиях мощных электромагнитных помех. В особо надежных сетях следует использовать более дорогостоящие и дефицитные ограничительные диоды — сапрессоры, способные при превышении на них допустимого напряжения пропускать через себя за несколько нс токи до сотен кА. Следует также сказать, что описанные методы защиты линий $CI-LAN$ могут использоваться и в сетях на базе интерфейса $RS232C$.

Описанные выше диспетчеры DPC ориентированы на использование только с одним из выходных сетевых интерфейсов. Кроме того, примененные для оптической изоляции отечественные оптрона имеют достаточно низкое быстродействие. Известно, что обеспечение высокого быстродействия собственно DPC резко сокращает вероятность возникновения ошибок рассогласования при работе сетей.

Рассмотрим характеристики наиболее известных отечественных и зарубежных оптронов широкого применения, приведенные в табл. 2.1.

Таблица 2.1

Тип	Ток светодиода, мА	Макс. Узк, В	Макс. Унас, В	Время переключения ON/OFF, мкс	Напряжение изоляции, кВ
30T122	<15		1,5	100	0,1
30T127	<15		1,5	100	0,5
30T13	<10		1,5		1,0
30T131A	<30		1,5	>20*	1,0
30T135	<20		1,5	>20*	1,0
30T138	<25		0,3		1,0
30T142	<30		1,5		10,0
30T144A	<25		0,3		3,0
30T146	<30		0,4		
30T150A	<20		1,5		3,0
4N25	<12	30	0,5	1,2/1,3	5,3
4N26	<12	30	0,5	1,2/1,3	5,3
4N27	<12	30	0,5	1,2/1,3	5,3
4N28	<12	30	0,5	1,2/1,3	5,3
4N35	<12	30	0,3	10/10	5,3
4N36	<12	30	0,3	10/10	5,3
4N37	<12	30	0,3	10/10	5,3
4N38	<12	80	1,0	–	5,3
6N135	16	15	0,4	1,5/1,5	2,5
6N136	16	15	0,4	0,8/0,8	2,5
6N137	5	7	0,6	0,075/0,075	2,5
6N138	1,6	7	0,4	10/35	2,5
6N139	0,5	18	0,4	1/7	2,5
CNG35	<12	30	0,4	20/20	5,3
CNG36	<12	30	0,4	20/20	5,3
CNW11AV-1	<12	70	0,4	15/15	4,0
CNW11AV-2	<12	70	0,4	15/15	4,0
CNW11AV-3	<12	70	0,4	15/15	4,0
CNW135	16	30	0,4	1,5/1,5	5,0
CNW136	16	30	0,4	0,8/0,8	5,0
CNW138	1,6	7	0,4	10/35	5,0
CNW139	0,5	18	0,4	1/7	5,0
CNW4502	16	30	0,4	0,8/0,8	5,0
CNW82		50	0,4	–	5,9

Продолжение табл. 2,1

Тип	Ток светодиода, мА	Макс. Узк, В	Макс. Унас, В	Время переключения ON/OFF, мкс	Напряжение изоляции, кВ
CNW83	<16	50	0,4	–	5,9
CNW84		80	0,4	–	5,9
CNW85	<16	80	0,4	–	5,9
CNX35U	<16	30	0,4	20/20	5,3
CNX36U	<16	30	0,4	20/20	5,3
CNX38U	<16	80	0,4	20/20	5,3
CNX82A		50	0,4	–	5,3
CNX83A	<16	50	0,4	3/3	5,3
CNY17	<12	70	0,4	10/10	5,3
CNY17(G)F-(1-4)		70	0,4	10/10	5,3
H11A(1-5)	<12	30	0,4	–	5,3
H11AG1		30	0,4	–	5,3
H11AG2		30	0,4	–	5,3
H11AG3		30	0,4	–	5,3
H11D(1-4)		300	0,4	–	5,3
HCPL-2503	16	15	0,4	0,8/0,8	2,5
HCPL-2530	16	30	0,5	1,5/1,5	2,5
HCPL-2531	16	30	0,5	0,8/0,8	2,5
HCPL-2601	5	7	0,6	0,075/0,075	2,5
HCPL-2611	5	7	0,6	0,075/0,075	2,5
HCPL-2630	5	7	0,6	0,075/0,075	2,5
HCPL-2631	5	7	0,6	0,075/0,075	2,5
HCPL-2730	1,6	7	0,4	20/35	2,5
HCPL-2731	0,5	18	0,4	2/10	2,5
HCPL-4502	16	15	0,4	0,8/0,8	2,5
IL1	<12	50	0,25	–	5,3
IL2	<12	70	0,25	–	5,3
IL201	<12	70	0,4	–	5,3
IL202	<12	70	0,4	–	5,3
IL203	<12	70	0,4	–	5,3
IL5	<12	70	0,25	–	5,3
IL74	<12	20	0,3	3/3	5,3
MCT2		30	0,4	1,2/1,3	5,3
MCT210		30	0,4	–	5,3

Продолжение табл. 2.1

Тип	Ток светодиода, мА	Макс. Узк, В	Макс. Унас, В	Время переключения ON/OFF, мкс	Напряжение изоляции, кВ
МСТ2200		30	0,4	10/10	5,3
МСТ2201		30	0,4	10/10	5,3
МСТ2202		30	0,4	10/10	5,3
МСТ270		30	0,4	10/10	5,3
МСТ271		30	0,4	7/7	5,3
МСТ272		30	0,4	10/10	5,3
МСТ273		30	0,4	20/20	5,3
МСТ274		30	0,4	25/25	5,3
МСТ275		30	0,4	15/15	5,3
МСТ276		30	0,4	3,5/3,5	5,3
МСТ277		30	0,4	15/15	5,3
МСТ2Е		30	0,4	1,2/1,3	5,3
МСТ5200		30	0,4	12/20	5,3
МСТ5201		30	0,4	30/13	5,3
МСТ5210		30	0,4	–	5,3
МСТ5211		30	0,4	–	5,3
МСТ6	<10	30	0,4	–	2,5
МСТ61	<10	30	0,4	–	2,5
МСТ62	<10	30	0,4	–	2,5
МСТ9001	<10	55	0,4	–	2,5
МОС8100		30	0,5	20/20	7,5
МОС8111		30	0,4	10/10	5,3
МОС8112		30	0,4	10/10	5,3
МОС8113		30	0,4	10/10	5,3
МОС8204		400	0,4	–	7,5
МОС8205		400	0,4	–	7,5
МОС8206		400	0,4	–	7,5
РС111L	<12	35	0,2	4/3	5,0
РС112L	<12	70	0,2	4/3	5,0
PS2501-2	<10	80	–	3,5	5,0
PS2511-2	<10	40	–	3,5	5,0
PS2561-2	<10	50	–	3,5	5,0
PS2601		50	–	3,5	5,0
PS2651		50	–	3,5	5,0

Продолжение табл. 2.1

Тип	Ток светодиода, мА	Макс. Узк, В	Макс. Унас, В	Время переключения ON/OFF, мкс	Напряжение изоляции, кВ
PS2701-2	<10	80	–	3,5	3,75
SFH600-(0-4)		70	0,4	3/3	5,3
SFH601-(0-4)		100	0,4	3/2,3	5,3
SFH608-(2-5)		55	0,4	8/7,5	5,3
SFH640-(1-5)		300	0,4	5/6	7,5
SL5500		30	0,4	20/50	5,3
SL5501		30	0,4	20/50	5,3
SL5504		80	0,4	50/150	5,3
SL5511		30	0,4	20/50	5,3
SL5582		50	0,4	20/50	5,3
SL5582W		50	0,4	20/50	5,9
SL5583		50	0,4	20/50	5,3
SL5583W		50	0,4	20/50	5,3
TIL11	<12	30	0,4	10/10	5,3
TIL111	<12	30	0,4	5/5	1,5
TIL112	<12	20	0,5	2/2	1,5
TIL117	<12	30	0,4	5/5	2,5
TIL126	<12	30	0,4	2/2	1,5
AOT101	<20		0,4	10	1,5
AOT102	<40		4	5	0,5
AOT110	<40		1,5	100	0,5
AOT123	<30		0,3	2	0,1
AOT126	<30		0,3	2	1,0
AOT128	<40		0,3	5	1,5
AOT137	<30		0,4		
AOT147	<30		0,4		8,0
AOT151A	<30		0,4		8,0
КОЛ201A	<10			1	0,5

Анализ приведенной таблицы показывает, что практически все отечественные и большинство зарубежных оптронов имеют время включения/выключения больше 1 мкс, что с трудом позволяет обеспечивать устойчивую работу на скоростях 19200 и 28800. Кроме того, параметры оптронов сильно зависят от партии и производителя. Как показали исследования, проведенные автором, параметры быстродействия многих отечественных оптронов, да и некоторых импорт-

ных, значительно завышены. Фактически все оптроны по быстродействию можно разделить на 3 группы: с высоким быстродействием (6N137, HCPL-2601, HCPL-2611, HCPL-2630, HCPL-2631), со средним быстродействием (6N136, CNW4502, HCPL-4502) и с низким быстродействием, к которым следует отнести все остальные оптроны.

Оптроны с высоким быстродействием достаточно дороги и трудно доставаемы. Из оптронов средней группы наиболее доступен оптрон 6N136. Его испытания показали, что он устойчиво работает даже на скорости 115200, что является пределом скорости большинства COM-портов современных компьютеров. В связи с этим этот тип оптрона и был признан автором в качестве оптимального для создания быстродействующего диспетчера DPC, принципиальная схема которого приведена на рис. 2.3.

Устройство содержит: входной приемопередатчик RS232C, выполненный на микросхеме D1 (MAX202, AMD202, MAX232 и т. д.), обеспечивающий связь с персональным компьютером; узел оптической развязки D2—D5 (6N136), осуществляющий развязку входных цепей, соединенных с персональным компьютером от выходных цепей, связанных гальванически с локальной сетью или периферийным устройством; выходной приемопередатчик RS232C на микросхеме D6 (MAX202, AMD202, MAX232, и т. д.), осуществляющий связь с выходным RS232C-каналом; а также выходной приемопередатчик RS485 D7 (MAX485, MAX487). Кроме этого, в состав устройства входят два изолированных друг от друга линейных стабилизатора напряжения на +5 В для изолированного питания входной и выходной части, реализованных на микросхемах A1, A2 (78L05).

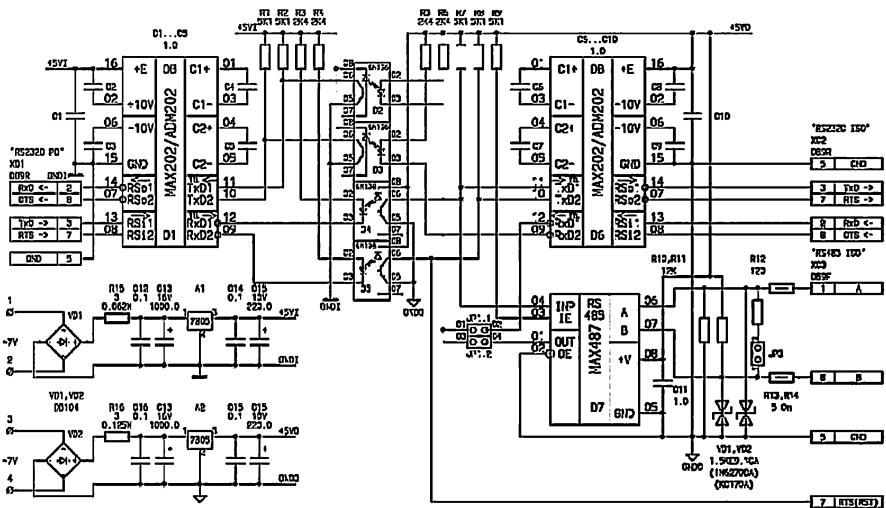


Рис. 2.3. Быстродействующий диспетчер персонального компьютера для сетей на базе интерфейсов RS232C и RS485

Устройство обеспечивает умощнение с оптической развязкой двух каналов интерфейса *RS232C* персонального компьютера: первый канал составляют штатные линии приемопередатчиков (*RxD* и *TxD*), второй канал составляют вспомогательные линии интерфейса (*CTS* и *RTS*), которые можно использовать по прямому назначению для организации обмена или как дополнительный канал приемо-передачи с программной эмуляцией обмена. Узел оптической развязки выполнен по классической схеме включения оптронов *6N136*. В случае, если высокое быстродействие дополнительного канала не требуется, можно заменить *D3* и *D5* на более распространенные и дешевые оптроны *4N35*. На рис. 2.4 показано расположение выводов оптронов *6N136* и *4N35*. Из сопоставления разводки выводов видно, что оптрон *6N136* может быть заменен на оптрон *4N35* путем установки последнего в колодку *DIP8* так, чтобы первый вывод оптрона *4N35* вошел во второй вход колодки. Очевидно, что при такой замене быстродействие второго канала заметно упадет. Оптроны *6N136* обеспечивают устойчивую передачу на скоростях до 115200 включительно, в то время как оптроны *4N35* с трудом работают на скорости 38400, а устойчиво ведут себя только на скорости 28800.

Напряжение изоляции входной и выходной частей изделия составляет 2,5 кВ при установке хотя бы одного оптрона *6N136* и 5,3 кВ при установке всех оптронов *4N35*. Следует отметить, что напряжение изоляции определяется также используемым трансформатором. В устройстве рекомендуется использовать любой из трансформаторов ТПП213—ТПП218, ТПП222—ТПП225 (220/127-50), которые обеспечивают напряжение изоляции между обмотками выше 7 кВ.

Особенностью выходной части является то, что передача информации от *DPC* к периферийному устройству (устройствам) не коммутируется, т. е. ведется одновременно по выходным основному каналу *RS232* и каналу *RS485*. Прием информации коммутируемый. Выбор канала поступления данных от периферийных устройств осуществляется с помощью установки перемычек *JP1* или *JP2*. К штырям *JP1/2* может быть подключен малогабаритный тумблер, установлен-

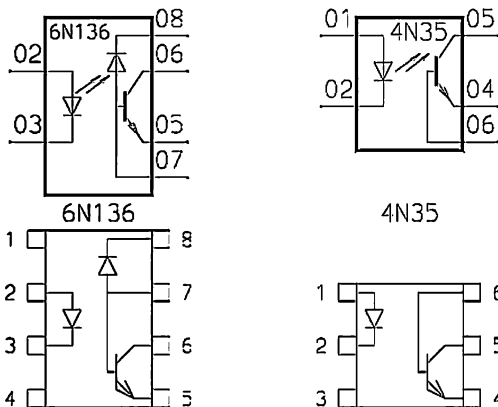


Рис. 2.4. Расположение выводов оптронов *6N136* и *4N35*

ный на передней панели и осуществляющий переключение выходных каналов RS232—RS485.

Канал интерфейса RS485 имеет защиту от перенапряжений с помощью сапрессоров VD1, VD2 (1.5KE9.1CA или 1N6270CA) на напряжение 7,2 В. Сапрессоры — это быстродействующие мощные стабилитроны (индекс CA указывает, что они двунаправленные), обеспечивающие срабатывание при напряжении выше 7,2 В и замыкающие на себя ток до 40 А. В лабораторных условиях, если подключаемая линия не очень длинная (до 10 м), их можно с успехом заменить на отечественные стабилитроны КС170/175/191А. Ограничение входного тока и роль предохранителей на линиях А и В интерфейса выполняют резисторы R13, R14 величиной 3—5 Ом. Резистор R12 величиной 120 Ом устанавливается на концах линии (концевых станциях сети) для согласования. Резисторы R10, R11 подтягивают свободно висящие линии и улучшают динамические характеристики сети. На разъеме интерфейса RS485 выведена также изолированная линия сигнала (RTS/RST), которая может служить сигналом сброса одновременно всех станций сети RS485 для их последующей инициализации.

Входная часть устройства потребляет не более 10 мА, выходная — не более 15 мА. Поэтому в стабилизаторах используются микросхемы A1, A2 78L05. Резисторы R15 и R16 выполняют роль предохранителей (ваттность не более 0,062 Вт). В особо ответственных изделиях следует устанавливать импортные самовосстанавливающиеся предохранители на ток 50 мА. Входное напряжение для стабилизатора может быть от 7 до 12 В. В устройстве могут быть использованы диодные мосты DB10х с последними цифрами от 1 (100 В) до 7 (1 кВ), выполненные в корпусе DIP6 или любые другие малогабаритные мосты. Устройство собрано на двухсторонней печатной плате с размерами 90 × 62 мм² [49].

В заключение раздела отметим, что приведенные диспетчеры персонального компьютера (DPC) для управления доступом к каналу используют самый простой активный способ управления. На самом деле в настоящее время существует несколько разнообразных вариантов управления каналом сети RS485, которые мы рассмотрим в следующем разделе. Кроме того, поскольку не существует стандартов на управляющие сигналы интерфейса RS485, в различных изделиях для этого используются различные сигналы (DTR или RTS) в различной полярности. Конечно, возможно создание универсального диспетчера, и реализующего все возможные варианты управления каналом, и осуществляющего оптическую изоляцию всех линий входного интерфейса RS232C, и переводящего выключенные выходные интерфейсы RS232C или RS485 в высокоимпедансное состояние, однако такой диспетчер будет достаточно сложным и дорогостоящим изделием с ограниченной сферой применения.

2.3. Способы управления доступом к каналу сети RS485

Вообще говоря, сети на основе интерфейса RS485/RS422 могут работать в дуплексном (RS422) и полудуплексном (RS485) режимах. Дуплексный режим требует для передачи данных четыре линии связи (две линии для передачи и две

для приема), а полудуплексный — две (мультиплексированные двунаправленные для приема и передачи). Кроме того, сеть с дуплексным режимом обмена имеет примерно в два раза большие аппаратные затраты интерфейсных узлов. Два перечисленных недостатка привели к тому, что полдуплексный режим используется достаточно редко. В связи с этим, в рамках настоящего раздела мы рассмотрим только особенности управления каналом полдуплексных сетей.

Типовая структура сети на основе интерфейса *RS485* представлена на рис. 1.2. Структура содержит N станций (*Station 1, 2, ..., N*). Каждая станция имеет узел управления доступом к каналу сети, называемый станционным диспетчером (*DS или MGT*). Станционный диспетчер соединяется своими выходами «А» и «В» с одноименными линиями связи. На концах линий связи в обязательном порядке должны быть расположены согласующие резисторы R_c , равные 120 Ом.

Таким образом, станционный диспетчер собственно и осуществляет управление доступом к каналу сети. Обобщенная структура диспетчера станции представлена на рис. 2.5.

Термин «*обобщенная структура*» означает, что эта структура — собирательная, составлена на основе анализа многочисленных структур различных сетей, описанных в специальной литературе. С левой стороны рисунка показан микроконтроллер или микропроцессор (*MCU — MicroController Unit*). Подразумевается, что в большинстве случаев они являются ядром современных сетевых станций. На этом и последующих рисунках непрерывными линиями показаны основные функциональные связи, а пунктирными — необязательные связи. В диспетчерах сигналы подразделяются на станционные, т. е. идущие вовнутрь станции (показаны слева, между диспетчером и микроконтроллером), и сетевые, т. е. идущие от станции к каналу сети (показаны справа от диспетчера).

Станционные сигналы подразделяются на основные и дополнительные. Основными сигналами являются: *RxD* — входная (по отношению к микроконтроллеру и станции в целом) линия данных последовательного порта и *TxD* — выходная линия данных последовательного порта. Очевидно, что основные линии служат для передачи данных между микроконтроллером и диспетчером. Дополнительными сигналами являются: *Control* — сигнал или сигналы управления диспетчером и *RST(/)* — сигнал сброса микроконтроллера (инвертированный или неинвертированный).

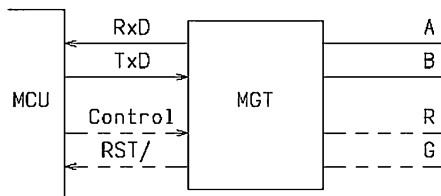


Рис. 2.5. Обобщенная структура диспетчера станции

Сетевые сигналы также подразделяются на основные и дополнительные. Основные сигнальные линии — информационные линии *A* и *B*. Дополнительные необязательные линии: *G* (*GND*) — общая линия или земля и *R* (*RST/*) — необязательная линия сброса всех или части станций. Линия *G* необязательная, но используется достаточно часто для связи земель станций. Иногда в качестве линии *G* используется экранная оплетка основных сигнальных проводов. Линия *R* используется лишь в тех случаях, когда необходимо обеспечить сброс части или всех станций сети от одной из них. Вообще говоря, использование этой линии повышает надежность функционирования сети, поскольку позволяет нормально функционирующим станциям сбросить «зависшие станции». Эта возможность особенно важна в специализированных системах реального времени, распределенных системах управления и в лабораторных (исследовательских) сетях.

Существует несколько способов управления доступом к каналу сетей на базе интерфейса *RS485*, работающих в полудуплексном режиме:

- активный способ управления;
- пассивный способ управления;
- управляемый пассивный способ управления;
- способ управления с автоопределением скорости;
- способ управления с управляемым контроллером;
- способ управления с интеллектуальным формирователем фрейма.

В соответствии с этими способами особым образом реализуются и станционные диспетчеры. Рассмотрим структуры диспетчеров, реализующих эти способы управления. При этом в качестве базовых будем использовать широко распространенные полудуплексные микросхемы интерфейса *RS485* — *MAX485* (*MAX487*, *ADM485*).

На рис. 2.6 показана обязательная часть станционного диспетчера с активным управлением доступом. Достоинством этого диспетчера (и способа управления) являются минимальные аппаратные затраты. Главными недостатками являются: необходимость формирования дополнительных сигналов управления (*Control*), необходимость занятия дополнительных линий (ввода/вывода) микроконтроллера и более сложное программное обеспечение станции (например, по отношению к диспетчеру с пассивным управлением). Вообще говоря, в описываемых сетях наиболее часто используется режим, в котором входы интер-

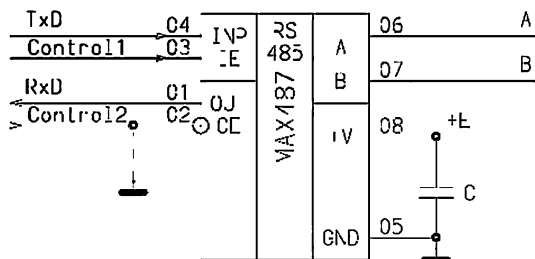


Рис. 2.6. Основная часть диспетчера с активным управлением доступом

фейса *RS485* всегда открыты (станции принимают все передачи по линии), т. е. сигнал *Control2* отсутствует, а вывод *02* микросхемы соединен с землей. Таким образом, в таких режимах остается только один сигнал управления — *Control1*. Проблема еще больше усложняется, если в качестве ядра станции используется не микроконтроллер, а персональный компьютер, работающий в операционной среде Windows (вследствие многозадачности системы, приоритетности задач и виртуализации периферии).

На рис. 2.7 схематично показана дополнительная необязательная часть диспетчера. Эта часть может использоваться с диспетчерами любых типов. Она содержит резистор *R*, одним выводом подключенный к питанию контроллера, а вторым через ключ *S* к общему проводу. Точка между резистором *R* и ключом *S* соединяется с дополнительной линией *R* — сброса сети, при этом обязательно должна присутствовать вторая дополнительная линия *G* — общий провод. Кроме того, внутри станций, которые должны сбрасываться от сигнала сброса, эта точка должна быть подключена ко входу сброса микроконтроллера непосредственно, через инвертор или супервизор питания. При подаче сигнала управления *Control3* на любой из станций открывается ключ *S* и линия *R* оказывается в состоянии логического нуля. Этот сигнал поступает в каждой станции на соответствующий узел (чаще всего супервизор питания), который вырабатывает сигнал сброса *RST* микроконтроллера необходимой длительности и полярности и вызывает его инициализацию.

Недостатки диспетчера с активным управлением доступом к каналу привели к тому, что большинство фирм-производителей сетевого оборудования или промышленных контроллеров начали активно использовать пассивный способ управления и соответствующую структуру диспетчера, приведенную на рис. 2.8.

Принцип пассивного способа управления заключается в том, что стационарный входной сигнал диспетчера передачи данных *TxD* подается не только на микросхему интерфейса *MAX485*, но и на вход формирователя положительных импульсов, срабатывающий по отрицательному (заднему) фронту импульса (перепаду с высокого в низкий логический уровень). При такой схеме начало передачи — первый же стартовый импульс сигнала *TxD* вызовет срабатывание формирователя *F*, который выходным положительным импульсом заданной длительности откроет выход передатчика *RS485* — вывод *03*. Достоинства такого диспетчера: отсутствие дополнительных линий управления и упрощение (по от-

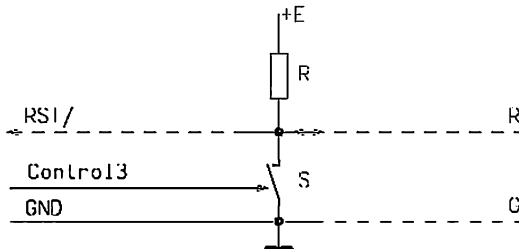


Рис. 2.7. Структура узла сброса сети

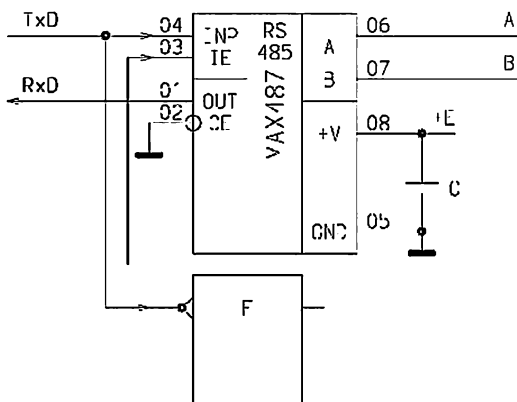


Рис. 2.8. Диспетчер с пассивным управлением доступом к каналу

ношению к активному способу управления) программного обеспечения. Недостатком описанного диспетчера является то, что используемый формирователь имеет определенную длительность выходного импульса и перезапускает этот импульс при любом отрицательном фронте импульса на входе. Это означает, что время, в течение которого выход передатчика будет открытым, зависит от содержимого последнего байта передачи. Это в свою очередь означает, что для избежания конфликтов передач различных станций необходимо обеспечить паузу после каждой передачи как минимум на установленную длительность импульса формирователя, что естественно снижает максимальную производительность сети. К тому же эту паузу необходимо устанавливать с некоторым запасом, т. к. обычно используются простейшие схемы формирователей, например на *74НС123*, не отличающиеся высокой температурной стабильностью длительности выходного импульса. Положение еще более усложняется, если сеть должна функционировать на различных скоростях, при этом длительность паузы выбирается равной длительности передачи байта на наименьшей скорости. На высокой скорости при таком решении потери производительности значительно возрастают. Показателен простой пример. Допустим, что сеть должна работать на скоростях от *2400* до *28800*. Если при этом длительность паузы выбрать исходя из времени передачи одного байта на скорости *2400*, то на скорости *28800* (которая имеет в 12 раз меньшее время передачи байта) мы будем терять после каждой передачи лишние 12 байт! Конечно, это весьма расточительно и недопустимо!

Для устранения этого недостатка в сетях с несколькими возможными скоростями передачи иногда используются диспетчеры с управляемым пассивным режимом (см. рис. 2.9).

Этот диспетчер отличается от предыдущего тем, что на формирователь подаются дополнительные управляющие сигналы, устанавливающие необходимую длительность импульса в зависимости от установленной скорости передачи. Достоинства этого диспетчера понятны из описания недостатков предыдущего, а

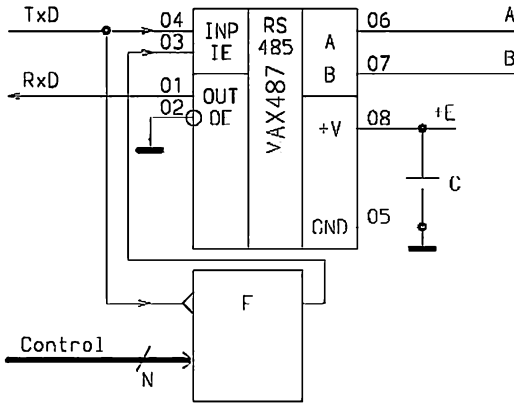


Рис. 2.9. Диспетчер с управляемым пассивным режимом

вот недостатки у него тоже есть. Во-первых, необходимо иметь как минимум одну, а чаще — несколько линий управления. Во-вторых, усложняется программное обеспечение.

На рис. 2.10 показан интеллектуальный диспетчер с автоопределением скорости. Структура этого диспетчера очень похожа на структуру диспетчера с пассивным управлением доступом. Разница состоит лишь в том, что вместо формирователя *F* используется дополнительный микроконтроллер (*MCU*), играющий роль формирователя. Вообще говоря, под этим названием встречаются два различных варианта реализации диспетчера и, соответственно, два способа управления каналом: авто-определение скорости и предварительная настройка скорости.

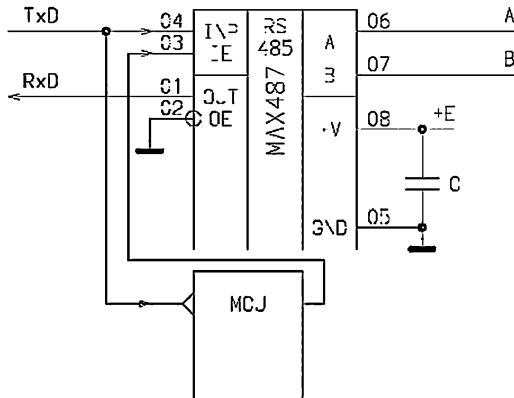


Рис. 2.10. Интеллектуальный диспетчер с автоопределением скорости

Диспетчер с автоопределением скорости передачи содержит вместо формирователя дополнительный микроконтроллер, который запускается по отрицательному фронту первого стартового бита (по прерыванию) и рассчитывает скорость передачи, а точнее, длительность передачи байта. Таким образом, производится автоматическая настройка диспетчера на текущую скорость передачи. Такое решение встречается достаточно редко в дорогих и ответственных сетях, хотя в последнее время стоимость отдельных микроконтроллеров и приближается к стоимости дискретных микросхем. Достоинствами диспетчера с автоматическим определением скорости являются простота эксплуатации и малые потери времени работы сети на паузы между передачами. Недостатками являются сложность разработки и более высокая стоимость.

Диспетчер с предварительной настройкой скорости имеет такую же структуру, но принцип работы его полностью отличается. Обычно в локальных сетях все станции имеют свой адрес. Интерфейсные микросхемы *RS485* по нагрузочной способности обычно обеспечивают соединение до 32 (*MAX485*, *ADM485*), 128 (*MAX487*) или 256 (*MAX3080—MAX3089*) станций. При реализации способа управления с предварительной настройкой один из этих адресов присваивается всем микроконтроллерам всех станций. В этом случае передаваемая по сети команда настройки на заданную скорость воспринимается всеми дополнительными контроллерами всех станций сети и устанавливает длительность передачи одного байта на заданной скорости. Далее дополнительный контроллер отслеживает начало передачи и поддерживает выход передатчика в открытом состоянии в течение всей передачи. Достоинствами диспетчера с предварительной настройкой являются простота эксплуатации, малые потери времени работы сети на паузы между передачами (т. е. более высокая предельная производительность сети), более надежное функционирование и более простая реализация. Недостатками являются более высокая стоимость (по сравнению с «пассивным» диспетчером) и необходимость занятия одного адреса станции.

Встречается также так называемый интеллектуальный диспетчер с управлением (см. рис. 2.11), в котором также используется дополнительный микроконтроллер, но скорость ему задается по дополнительным линиям управления. Его структура очень похожа на структуру диспетчера с управляемым пассивным режимом. Достоинства такого структурного решения в простоте управления, реализации и малых потерях времени на паузы. Дополнительный микроконтроллер первый раз запускается по отрицательному фронту (по прерыванию), а затем отслеживает передачу по стартовым и стоповым битам и сразу же закрывает доступ к каналу после завершения передачи. Недостатки заключаются в необходимости выделения дополнительных линий управления и более высокой стоимости.

Анализ использования интеллектуальных диспетчеров позволил сформировать структуру интеллектуального диспетчера с формированием фрейма. Напомним, что данные в *LAN* передаются в виде фреймов (*frame*) или кадров — последовательности байтов, содержащей преамбулу (байт или байты синхронизации), байт адреса станции-получателя данных, байт команды, передаваемые данные, иногда адрес станции-передатчика, контрольный байт (обычно кон-

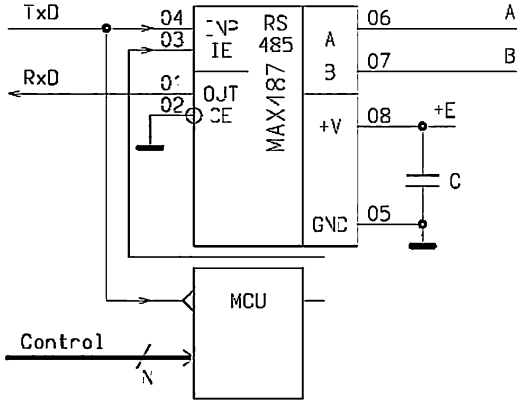


Рис. 2.11. Интеллектуальный диспетчер с управлением

трольная сумма). На формирование фрейма тратятся различные ресурсы основного микроконтроллера станции (память, время, линии ввода/вывода для подключения задатчика адреса станции и т. п.). Идея заключается в том, что раз уж в интеллектуальных диспетчерах используется дополнительный микроконтроллер, то почему бы на него не возложить все функции по формированию и передаче фрейма, а также по получению входных фреймов, анализу ошибочных ситуаций и их устранению. Структура интеллектуального диспетчера с формированием фрейма показана на рис. 2.12.

Основной микроконтроллер может обмениваться с дополнительными данными через привычный последовательный канал или через любой другой имеющийся в обоих микроконтроллерах канал, например *SPI (Serial Peripheral Interface)*. Передаваться между ними может только полезная информация: команда и данные. Адрес приемника может передаваться в параллельном виде. Адрес своей станции дополнительный микроконтроллер может определять от подключенного к нему модификатора. Дополнительный микроконтроллер в этом случае может сам формировать фрейм и осуществлять передачу, получать вход-

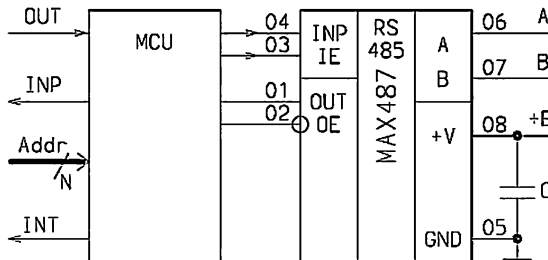


Рис. 2.12. Интеллектуальный диспетчер с формированием фрейма

ные фреймы, расшифровывать их и выдавать основному микроконтроллеру прерывание, свидетельствующее о поступлении неповрежденного фрейма и готовности передачи в основной микроконтроллер только полезных данных. Такое решение не увеличивает аппаратные затраты по сравнению с остальными интеллектуальными диспетчерами, но значительно освобождает вычислительные мощности основного микроконтроллера станции.

Описанные в настоящем разделе способы управления доступом к каналу сети RS485 и структуры диспетчеров, реализующие эти способы, позволяют разработчикам выбрать наиболее приемлемый и оправданный вариант для своей реализации локальной сети. В заключение хочется отметить, что в настоящее время наиболее часто в промышленных контроллерах и сетевом оборудовании встречаются диспетчеры с пассивным управлением (~60 %). Второе место по применяемости занимают диспетчеры с активным управлением, наиболее часто используемые в единичных или мелкосерийных изделиях (~30 %). Третье место занимают интеллектуальные диспетчеры с формированием фрейма, причем доля их применяемости постоянно растет. Остальные способы используются примерно в 3—4 % случаев. Вероятнее всего, что по мере развития рынка дешевых малогабаритных и высокоскоростных микроконтроллеров в ближайшее время именно интеллектуальные диспетчеры с формированием фрейма будут все более укреплять свои позиции.

2.4. Диспетчеры периферийных станций

2.4.1. Диспетчер станции *CI-LAN «SISNET»*

Целью разработки *CI-LAN «SISNET» (Small Inform Service NET)* являлось создание малых распределенных систем, состоящих из *PC* и нескольких контроллеров и при этом обеспечение минимальных аппаратных затрат на ее создание [38—42, 50, 51]. Топология этой сети отличается от стандартной топологии *CI-LAN*. В описываемой сети отсутствует диспетчер *PC* и сетевая среда (*TM*) непосредственно подключается к интерфейсу RS232C персонального компьютера. Это накладывает ограничение и на количество станций (не более 16), и на общую длину сетевой среды (до 20 м). Второй особенностью этой сети является то, что диспетчер станции не содержит специализированных микросхем драйверов RS232C (например *MAX232* или аналогичных). В качестве сетевой среды используются стандартные 9-проводные компьютерные кабели длиной 1 или 1,2 м, снабженные двумя одинаковыми разъемами *DB9*. Диспетчер станции *CI-LAN «SISNET»* представлен на рис. 2.13.

На рисунке показан только один сетевой разъем *X1*, на самом деле каждый контроллер имеет по два соединенных параллельно разъема. Нарращивание сетевой среды осуществляется последовательным соединением контроллеров.

В описываемой сети используются шесть интерфейсных линий RS232C: линия *TxD* используется по назначению для передачи данных из *PC* к станциям; линия *RxD* используется по назначению, т. е. для приема данных из станций в

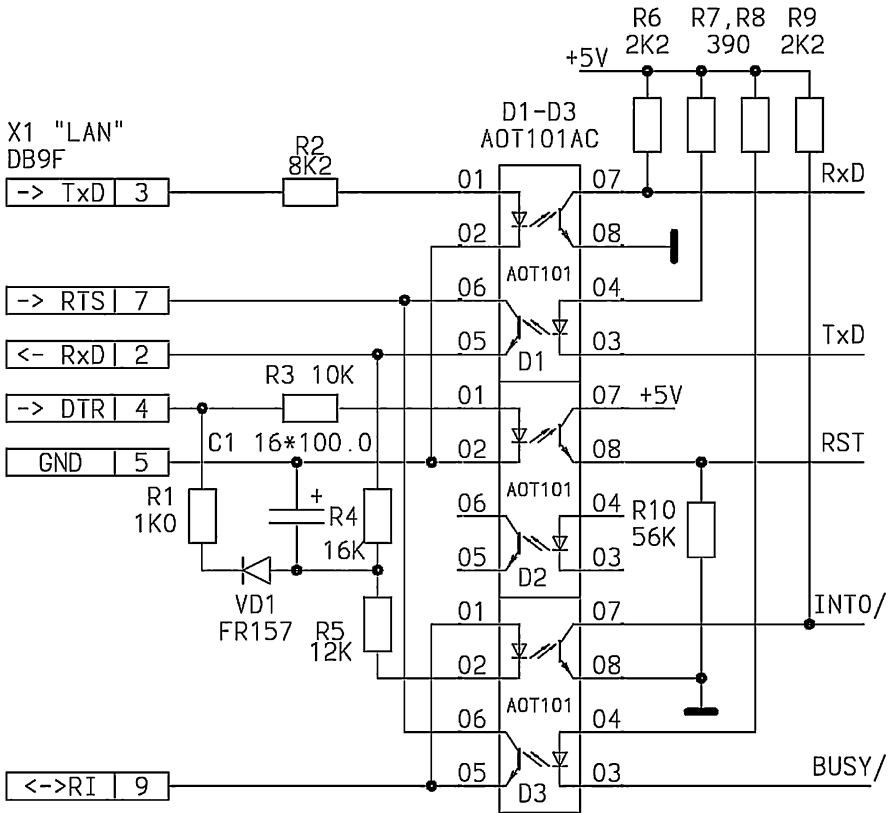


Рис. 2.13. Диспетчер станции CI-LAN «SISNET» (RS232C)

PC; линия *RTS* используется как источник положительного напряжения для питания входной части *MGT*; линия *DTR* используется для генерации сигнала сброса (*RST*) всех контроллеров сети, а также для создания через резистор *R1* и диод *VD1* отрицательного напряжения на конденсаторе *C1*; на линии *RI* генерируется сигнал занятости сети *BUSY/*; линия *GND* — общий провод сетевой среды.

Резистор *R1* необходим для снижения пиковых зарядных токов по линии *DTR*. Номиналы резисторов *R2—R5* подобраны так, что при параллельном подключении до 16 *MGT* к выходу интерфейса *RS232C* персонального компьютера обеспечивается максимально возможная нагрузка на сигнальные линии. Для оптической изоляции применены оптроны *AOT101AC*. Они имеют следующие преимущества: меньший ток срабатывания и время задержки переключения, два оптрона в корпусе. С другой стороны, напряжение изоляции у них более 500 В. Особенностью описываемой схемы является наличие схем формирования «сквозного» (для всех контроллеров) сигнала занятости и общего сброса сети.

Общий сброс всех контроллеров сети происходит при кратковременном переводе линии *DTR* из состояния *L* (нормальный режим) в состояние *H* (сброс). При этом положительное напряжение через резистор *R3* включает верхний светодиод оптопары *D2*, а соответствующий транзистор, открываясь, генерирует положительный импульс сброса микроконтроллера на резисторе *R10*. Сигнал занятости сети генерируется контроллером, который собирается начать передачу. При этом он опрашивает свой вход *INT0/*. Если вход имеет потенциал *H*, то сеть свободна и контроллер переводит свой выход *BUSY/* в состояние *L*. При этом включается нижний светодиод и соответствующий транзистор оптопары *D3*, на линии *RI* формируется потенциал *H*, который уведомляет *PC* о начале цикла передачи одним из контроллеров. Кроме того, включаются верхние оптопары всех контроллеров, при этом на входы *INT0/* всех контроллеров поступает низкий потенциал *L*, свидетельствующий о занятости сети. По окончании цикла передачи передающий контроллер переводит свой выход *BUSY/* в состояние *H*, снимая сигнал занятости линии. Описанный контроллер обеспечивает скорость передачи до 28800.

2.4.2. Диспетчер станции *CI-LAN «MISNET»*

На рис. 2.14 показана схема диспетчера станции *CI-LAN «MISNET»* (*Middle Inform Service NET*) на базе интерфейса *RS485* [40, 41, 52—54].

Приведенная схема имеет стандартную защиту входных цепей (см. описание рис. 2.2). Во входные цепи введена перемычка *JPI*, которая устанавливается на самом последнем контроллере для согласования сети. Отключение выходов передатчиков сети осуществляется переводом вывода *BUSY/* микроконтроллера в состояние *L*, при этом включается нижняя оптопара *D3* и вывод *DE* приемопередатчика *D1* переводится в состояние *L*, что, соответственно, переводит выходы передатчика в высокоимпедансное состояние.

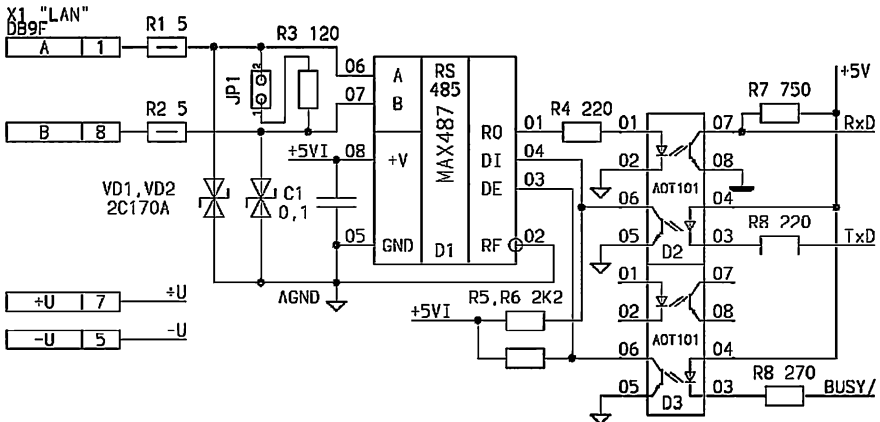


Рис. 2.14. Диспетчер станции *CI-LAN «MISNET»* (*RS485*)

2.4.3. Комбинированный диспетчер станции CI-LAN

В заключение этого раздела рассмотрим вариант принципиальной схемы более сложного комбинированного диспетчера контроллера (см. рис. 2.15). Такой диспетчер может быть применен в CI-LAN на базе интерфейса RS485 для подключения к сети современных измерительных приборов и промышленного оборудования, оснащенных интерфейсом RS232C. Другим вариантом применения приведенного диспетчера может быть создание многоуровневых сетей или сетей с комбинированным интерфейсом.

Сетевая часть диспетчера станции (слева) показана без разъемов и элементов защиты. При необходимости она может быть оснащена защитными или ограничительными диодами и оптической изоляцией от сетевой среды, как это было показано в предыдущих схемах. Для читателя наибольший интерес в данном случае представляет интерфейс с микроконтроллером.

Выходной сигнал RxD от интерфейса RS232C с вывода 12 микросхемы D1 (драйвер RS232C) поступает на первый вход элемента ИЛИ (D3.1), на второй вход которого поступает разрешающий сигнал приема (активный — L) I232_ON/. Выход этого элемента ИЛИ соединен с одним из входов элемента И (D4.1), на второй вход которого поступает RxD сигнал с выхода 01 микросхемы D2 (драйвер RS485). Этот выход может быть открыт при подаче на вход 02 микросхемы D2 разрешающего сигнала приема (активный — L) I485_ON/ или переводиться в высокоимпедансное состояние. При этом через резистор R4 формируется H уровень (разрешающий) на втором входе элемента И. Таким обра-

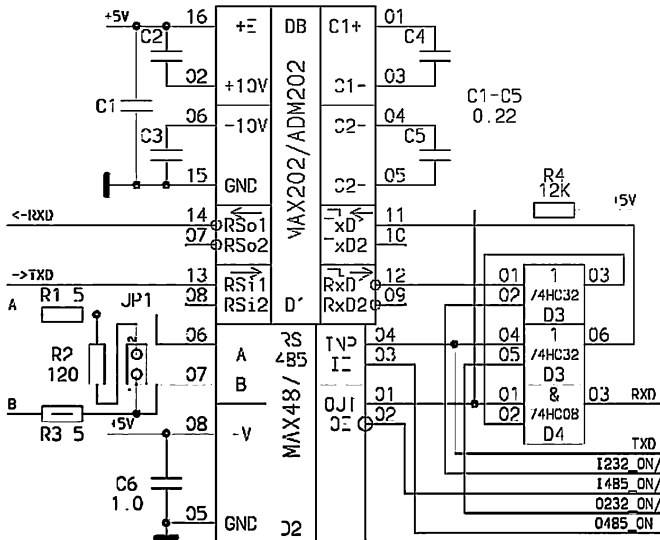


Рис. 2.15. Комбинированный диспетчер станции CI-LAN (RS485 и RS232C)

зом, при различных комбинациях управляющих сигналов микроконтроллера может быть открыт либо один из входных каналов, либо оба сразу.

Выходной сигнал микроконтроллера TxD поступает на вход 04 микросхемы $D2$ и один из входов второго элемента ИЛИ ($D3.2$), на второй вход которого поступает разрешающий сигнал передачи $O232_ON/$ (активный — L), а выход соединен со входом 11 (TxD) микросхемы $D1$. На вход 03 микросхемы $D3$ поступает разрешающий сигнал передачи (активный — L) $O485_ON$. Таким образом, оба передатчика могут открываться или закрываться в любых комбинациях.

Естественно, что отслеживать конфликты при одновременном приеме информации из двух источников, а также управлять диспетчером должно программное обеспечение микроконтроллера.

В заключение этого раздела хочется сказать, что приведены только несколько наиболее интересных схем из всего возможного разнообразия, однако их рассмотрение дает возможность даже начинающему специалисту самостоятельно разработать другие варианты диспетчеров станций.

2.5. Формат фреймов и общий алгоритм работы CI-LAN

Информация в *CI-LAN* передается в виде фреймов (*Frame*) — определенной последовательности байтов команды (от *PC*) и ответа (от *PS*). Существует стандартный формат фреймов, описанный во многих документах для локальных сетей, например в [33], позволяющий работать с фреймами произвольной длины. Однако этот формат в *CI-LAN* используется достаточно редко из-за громоздкости его обработки, избыточности и снижения производительности *CI-LAN* при его применении. Мы рассмотрим наиболее часто применяемый вариант форматов фреймов, а также опишем возможные пути его оптимизации.

Фрейм команды имеет следующий формат (табл. 2.2):

Таблица 2.2. Формат фрейма команды

№ байта	Наименование	Значение
B0	Преамбула	10101010
B1	Адрес станции	Address
B2	Код команды	Command
B3	Младший байт данных	Low Data Byte
B4	Старший байт данных	High Data Byte
B5	Контрольная сумма	$\Sigma(B0...B4)$

Первый байт (B0) — преамбула — байт синхронизации. Второй байт (B1) содержит адрес станции, которой персональный компьютер направляет команду. Код команды — код операции, которую должен выполнить адресуемый контроллер (станция). Далее следуют два байта данных. Если во фрейме конкретной команды данные не нужны, соответствующие байты заменяются нулями.

Все контроллеры станций в любой момент готовы принять фрейм команды. Как только фрейм принят, станции сравнивают адрес фрейма (B1) со своим уникальным адресом, который может быть задан с помощью модификаторов (перемычек или переключателей) или «зашит» в Flash ROM микроконтроллера. Если адрес принятого фрейма не соответствует индивидуальному адресу станции, фрейм игнорируется, никакие действия не производятся и станция ожидает приема следующего фрейма. Если же адрес фрейма совпал с индивидуальным адресом станции, станция рассчитывает контрольную сумму первых пяти байтов и сравнивает с принятым байтом B5. Контрольная сумма обычно считается упрощенно, все байты суммируются, а переносы в старший байт обнуляются. Если контрольная сумма не совпала, фрейм игнорируется и ожидается прием нового фрейма, в противном случае анализируется код команды, выполняются предписанные ей действия, например запись байтов данных во внешние управляющие регистры, формируется фрейм ответа в соответствии с табл. 2.3 и производится попытка передачи фрейма ответа.

Таблица 2.3. Формат фрейма ответа

№ байта	Наименование	Значение
B0	Преамбула	10101010
B1	Адрес станции	Address
B2	Код ошибки	Error Code
B3	Младший байт данных	Low Data Byte
B4	Старший байт данных	High Data Byte
B5	Контрольная сумма	$\Sigma(B0...B4)$

Формат фрейма ответа очень похож на формат фрейма команды за исключением поля B2, в котором может содержаться код ошибки или статуса состояния контроллера.

Как мы уже отмечали, только в сети «SISNET» линия *TxD* находится в монопольном владении *PC* и, следовательно, на ней невозможны конфликты. На линии *RxD* сети «SISNET» и в сетях на базе интерфейса *RS485* конфликты возможны в случае сбоя программного обеспечения систем или при аварийных ситуациях, при этом эти линии являются линиями со случайным доступом и программным анализом конфликтов (*Carrier Sense Multiple Access with Collision Detection network*).

Обычно в *CI-LAN* в случае неполучения фрейма ответа персональный компьютер повторяет команду определенное число раз. Если не получено никакого ответа, *PC* может направить в заданную станцию фрейм команды автотестирования; если после нескольких попыток и на это никакого ответа не получено, то *PC* может сгенерировать общий аппаратный сброс всех станций. Если и это не помогает, *PC* выдает сообщение оператору о фатальной или частичной аварии сети в зависимости от важности функций, выполняемых конкретной станцией.

Возможен и другой ход событий. Вследствие сбоев программной или аппаратной частей станций несколько станций могут попытаться одновременно начать передачу. Для этой ситуации обычно в набор команд станций вводят две специальные команды (*J-Signal* и *OK-Signal*). После приема первой команды в промежуток между выполнением команды и посылкой ответного фрейма вставляется программно-управляемая задержка, равная адресу станции, умноженному на время передачи одного ответного фрейма. При этом осуществляется как бы разнесение по времени ответов станций с различными адресами. Далее подается команда автотестирования, принимаются ответы от всех станций, определяются возможные повреждения, и, если их удается устранить программно, подается фрейм команды *OK-Signal*, которая восстанавливает нормальный рабочий режим сети.

Очевидно, что описан только один из вариантов работы *CI-LAN*. Последовательность операций даже в этом варианте может быть изменена без изменения основных принципов, например, станция сперва может рассчитать и сверить контрольную сумму, а затем проверить совпадение адресов и т. д.

Теперь отметим возможности по оптимизации формата фреймов. Очевидно, что от длины фрейма (т. е. количества байтов) зависит сложность его обработки, длина кода программного обеспечения, возможное количество циклов передачи (фрейм команды + фрейм ответа) в секунду и, в конечном счете, производительность *CI-LAN*.

Рассмотрим возможности по минимизации формата фреймов.

Во-первых, наиболее часто в *CI-LAN* используются 8-разрядные микроконтроллеры, которые наиболее часто работают с данными длиной в один байт. Пожалуй, единственным случаем, когда передача двух байт данных оправдана, является управление цифроаналоговыми преобразователями (*DAC — Digital-to-Analog Converter*) с разрядностью более байта.

Во-вторых, количество команд, выполняемых одним контроллером, редко превышает 16.

В-третьих, в малых сетях количество станций <16, следовательно, адрес занимает полубайт.

Анализируя приведенные доводы, можно сделать заключение, что в зависимости от ориентации, загруженности сети (количества команд и станций) и аппаратной реализации станций можно варьировать длину фреймов, например, объединяя адрес и код команды в один байт или/и удалив из фрейма один байт данных. Вообще, в *CI-LAN* следует, если это возможно, использовать одинаковую длину фреймов команды и ответа, что значительно упрощает программное обеспечение микроконтроллера станции. Но, конечно же, возможен и другой вариант, в сторону увеличения длины фрейма. Все зависит от конкретной ориентации сети...

Описывать алгоритм работы *PC* в *CI-LAN* мы не будем ввиду многообразия его реализации. Отметим лишь, что в простейшем случае алгоритм работы *PC* сводится к подаче в заданные моменты времени определенных команд и реакции на получаемые данные.

В общем случае алгоритм работы станции достаточно прост и представлен на рис. 2.16.

В модуле *INIT PORTS* производится необходимая инициализация переменных и установка регистров или отдельных битов в начальное состояние. Возможно проведение автотестирования и определение индивидуального номера. Модуль *GET COMMANDS* отвечает за получение и анализ командного фрейма. Модуль *MONITOR* анализирует код команды и вызывает соответствующие модули команд *COMMAND 0...F*. Модули команд содержат собственно заданные действия, предварительную отработку данных и формирование ответного фрейма. В случае если адрес и контрольная сумма правильные, но заданной команды не существует, в модуле *COMMAND ERROR* формируется соответствующий фрейм ответа. В некоторых сетях при ошибках контрольной суммы формируется в модуле *CRC ERROR* фрейм соответствующего ответа и производится попытка его передачи, однако такой вариант алгоритма более медленный, чем игнорирование ошибочного фрейма, т. к. при этом резко возрастает возможность конфликтов и затрат времени на их устранение. В модуле *POWER ERROR* формируется фрейм ответа аварийных ситуаций, и не только питания, но и любых других, обнаруживаемых самой станцией. Модуль *SEND COMMAND* осуществляет отправку фрейма.

Как уже было сказано выше, в модулях *COMMAND 0...F* происходит собственно выполнение команд. Различают три типа длины команд (по времени исполнения): короткие — типа записи/чтения байта в/из порт(а); средние и длинные — в качестве которых обычно считают программную реализацию чтения или записи по интерфейсу *SPI*.

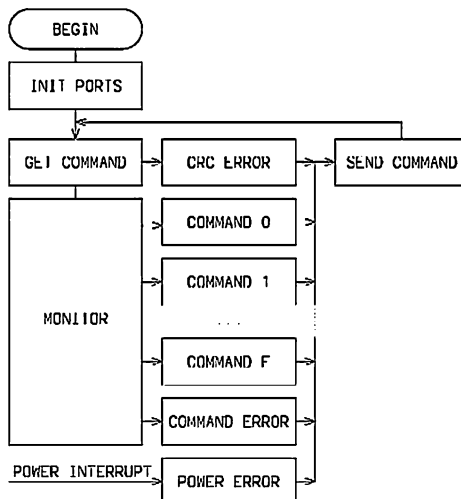


Рис. 2.16. Общий алгоритм работы контроллера станции *CI-LAN*

Существует достаточно простая методика [53], позволяющая проанализировать предельную производительность CI-LAN (т. е. максимально возможное количество циклов передачи-приема фреймов в секунду без учета аварийных ситуаций и сбоев). Последовательность расчета следующая:

1. Выбирается наиболее медленнодействующий контроллер станции (хотя обычно разработчики CI-LAN стремятся использовать однотипные микроконтроллеры). Для примера возьмем микроконтроллер AT89C51.

2. Выбирается тактовая частота микроконтроллера, например 11.059 МГц, при этом период составляет 0,090424 мкс. Определяется длительность одного машинного цикла (в нашем примере равная 12 периодам тактовой частоты) 1,085089 мкс.

3. Разрабатываются наиболее короткая и наиболее длинная команды контроллера.

4. Примерно полагается суммарное время, затрачиваемое PC на формирование фрейма команды и обработку ответного фрейма. Естественно, что это время зависит от многих факторов (типа и частоты процессора, операционной системы, алгоритма задачи и количества фоновых задач).

5. Разрабатывается тестовая программа работы станции, включающая все вышеперечисленные программные блоки.

6. С помощью программного отладчика (например HDC8051) определяются количество циклов и время выполнения каждого из блоков программы при заданной тактовой частоте контроллера. При этом исключаются любые программные задержки, связанные, например, с ожиданием приема или передачи байта.

7. Определяется полный цикл обмена LAN, который состоит из:

$$T_C = T_{PC} + T_R + T_{FRAME_GET} + T_{MONITOR} + T_{COMMAND} + T_{FRAME_SEND} + T_T, \quad (1)$$

где T_{PC} — суммарное время реакции PC; $T_R = F(S)$ — время приема одного фрейма от PC на скорости S ; T_{FRAME_GET} — время выполнения блока приема фрейма команды; $T_{MONITOR}$ — время выполнения блока монитора; $T_{COMMAND} = F(N)$ — время выполнения блока команды как функция от номера команды N и, соответственно, ее длины; T_{FRAME_SEND} — время выполнения блока передачи ответного фрейма; $T_T = F(S)$ — время передачи одного фрейма на скорости S .

8. С учетом того обстоятельства, что фреймы приема и передачи имеют одинаковую длину и передаются на одной скорости ($T_R = T_T$), формула 1 приобретает вид:

$$T_C(S, N) = T_{PC} + 2T_R(S) + T_{FRAME_GET} + T_{MONITOR} + T_{COMMAND}(N) + T_{FRAME_SEND}. \quad (2)$$

9. Рассчитывается производительность LAN в зависимости от скорости передачи данных S и длительности команды N по формуле:

$$P_{MAX}(S, N) = 1 / T_C(S, N). \quad (3)$$

10. Под максимальной (пиковой) производительностью CI-LAN понимается максимально возможное количество полных циклов обмена при выполнении наиболее короткой команды контроллера на максимальной скорости (при отсут-

ствии сбоев, задержек синхронизации, повторов передач фреймов, искажений контрольных сумм, минимальном времени реакции РС и т. д.). Соответственно, под средней производительностью *CI-LAN* понимается количество полных циклов обмена при выполнении команды средней длительности.

Описанная методика носит приближенный оценочный характер, т. к. проводится с учетом ряда допущений. В результате расчета и моделирования тестовой программы, приведенной ниже, получены результаты, показанные на рис. 2.17.

Расчетные значения приведены для двух описанных выше сетей *SISNET (RS232C)* и *MISNET (RS485)*, хотя тестовая программа приводится только для сети *MISNET*. Показаны три группы столбцов. Первая группа столбцов соответствует максимальной производительности сетей при выполнении самых коротких команд, вторая группа — средней производительности при выполнении средних команд и третья группа — минимальной производительности при выполнении длинных команд. Значения производительностей сетей приведены для различных скоростей передачи данных по последовательному каналу.

При расчетах использовался описанный выше формат фрейма и T_{PC} принято равным 5 мкс. Естественно, что возможная минимизация формата фреймов, заключающаяся в уменьшении длины фреймов на один байт, увеличит максимальную производительность (примерно на 18—21 %). Следует отметить, что при расчете максимально возможной производительности использовалась короткая команда (записи байта в порт — $T_{COMMAND} = 6,5$ мкс) При расчете минимально возможной производительности использовалась команда программной записи 16 бит по интерфейсу *SPI* ($T_{COMMAND} = 603$ мкс). Приведенные расчеты позволяют достаточно точно оценить возможности проектируемой *CI-LAN*, что подтверждается многочисленными экспериментами.

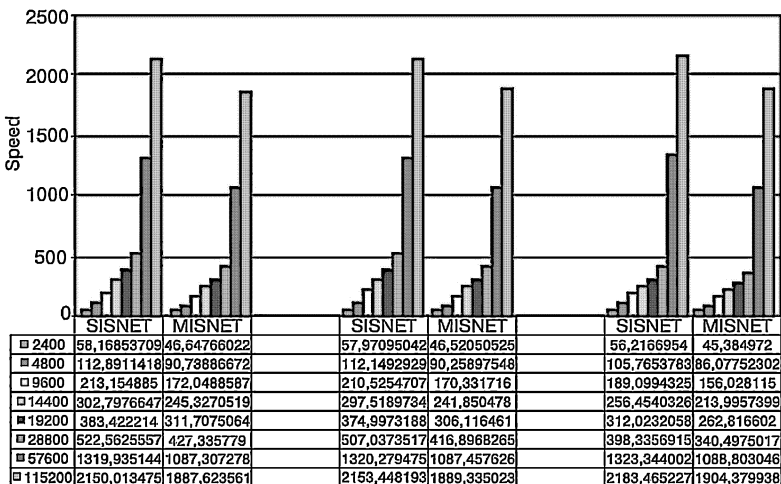


Рис. 2.17. Максимально и минимально возможная производительность *CI-LAN*

2.6. Тестовая программа контроллера станции CI-LAN

В настоящем разделе рассмотрим некоторые фрагменты тестовой программы микроконтроллера станции CI-LAN. Программа написана на языке «C» для микроконтроллера AT89C51 при частоте кварцевого резонатора 11,059 МГц.

В первом фрагменте сведены определения используемых переменных, инициализация данных и все основные подпрограммы общего пользования:

Фрагмент 1:

```

/*****/
/* Test Program for CI LAN "MISNET" */
/*****/
#include "..\include\io51.h"
#define ERROR 0
#define OK 1
#define byte unsigned char
code byte StartSpeed =24;

#pragma memory = data
byte FR[5]; /* Массив фрейма */
byte PTR; /* Счетчик */
byte MyAddress; /* Адрес станции */
byte Ready; /* Готовность приемника */
byte FLAG; /* Флаг приема фрейма */
#pragma memory = default
interrupt [0x23] void SCON_int (void) /* Процедура прерывания порта RS232C */
{
    Ready=1;
    ES=0;
}

void SpeedInit (byte NewSpeed) /* Подпрограмма настройки 1 таймера */
/* для работы RS232C */
{
char ch;
    TR1 = 0; /* Таймер 1 выключить */
    PCON = PCON | 0x80; /* Установка двойной скорости */
    TH1 = TL1 = (NewSpeed*0xFF)+1;
    ch = SBUF; /* Очистить приемный буфер */
    SCON = 0x52;
    TR1 = 1; /* Таймер 1 включить */
}

void SendB (byte SendByte) /* Подпрограмма передачи байта */
{
    while (!TI); /* Ожидать пока завершится предыдущая передача (TI=0) */
    TI=0; /* Очистить флаг передачи */
    SBUF=SendByte; /* Послать байт */
}

byte GetB (void) /* Подпрограмма приема байта */
{
byte InputByte;

```

```

while (!RI);                                /* Ожидать прием байта (RI=0) */
InputByte = SBUF;                            /* Получить байт */
RI = 0;                                       /* Очистить флаг приема */
return InputByte;                             /* Вернуть принятый байт */
}

void Time (char Delay)                       /* Подпрограмма задержки ~ (10+7*Delay) мкс */
{
    while (Delay)
    {
        Delay=Delay-1;
    }
}

void Init_Ports (void)                      /* Начальная инициализация микроконтроллера */
{
    EA=0;                                     /* Запретить все прерывания */
    RI=TI=PTR=0;
    FLAG=0;
    TMOD=0x22;                               /* Таймер в режиме 2: */
                                           /* автоперезагружаемый 8-битный счетчик */
    SpeedInit (StartSpeed);                 /* Установить начальную скорость 2400 */
    MyAddress =P2 & 0x7F;                   /* Прочитать адрес станции через порт P2 */
    Time (100);                             /* Задержка ~ 750 мкс */
    Ready=0;                                 /* Очистить флаг приема */
    TI=0;                                    /* Очистить флаг передачи */
    EA=1;                                    /* Разрешить все прерывания */
}

```

Приведенный фрагмент содержит достаточно полные комментарии, поэтому следует добавить только, что в описываемой программе все контроллеры после аппаратного сброса настраиваются на минимальную скорость 2400. После формирования РС аппаратного сетевого сброса и инициализации микроконтроллеров РС передает каждому контроллеру команду настройки на свою скорость, а затем команду автотестирования для определения состояния станции и устойчивости связи на заданной скорости.

Фрагмент 2 показывает основные подпрограммы приема и передачи фреймов:

Фрагмент 2:

```

/*****
/*** G E T   C O M M A N D   ***/
/*****/
int Frame_Get (void)
{
    byte Bi;
    while (1)
    {
        PTR=0;                               /* Очистить счетчик */
        FLAG=0;                             /* Сбросить флаг приема фрейма */
        while (PTR<6)                       /* Осуществить прием 6 байтов фрейма */
        {

```

```

ES=1;                /* Разрешить прерывания RI & TI */
while (!RI&!Ready); /* Если поступил байт, принять его */
FLAG=1;             /* Установить флаг приема фрейма */
Ready=0;            /* Очистить флаг готовности приемника */
Bi=GetB();          /* Полученный байт записать */
if (!PTR)           /* Если байт первый (PTR==0) и не равен коду
                    /* преамбулы (Bi!=0xAA), ждать следующий байт */
{
    if (Bi!=0xAA)   /* Синхронизация */
        continue;
}
FR[PTR++]=Bi;      /* Записать принятый байт во массив фрейма FR[PTR]
                    и увеличить счетчик */
}
FLAG = 0;          /* Фрейм принят, сбросить флаг приема */
Bi =(FR[0]+FR[1]+FR[2]+FR[3]+FR[4])&0x00FF;
                    /* Расчет контрольной суммы */
if (FR[5]!=Bi)    /* Если контрольная сумма фрейма не равна расчетной */
{
    PTR=0;         /* Сбросить счетчик */
    continue;     /* Ожидать следующий фрейм */
}
if ((FR[1]&0x7F)==MyAddress)
    break;        /* Если адрес фрейма равен адресу станции
                    - завершить, иначе ожидать следующий фрейм */
}
ES=1;             /* Разрешить прерывания RI & TI */
return OK;
}
/*****/
/**** SEND FRAME ****/
/*****/
void Frame_Send (void)
{
    while (FLAG); /* Ожидать завершения приема фрейма */
    PTR=0;        /* Обнулить счетчик */
    TI =1;        /* Разрешить передачу */
    while (PTR<6)
        SendB (FR[PTR++]); /* Передать 6 байт фрейма */
}

```

Фрагмент 3 показывает подпрограммы монитор, примеры написания нескольких команд и подпрограммы формирования фрейма ответа:

Фрагмент 3:

```

/*****/
/**** OUTPUT FRAME ****/
/*****/
void Output_Frame (byte B2, byte B3, byte Stat) /* Подпрограмма формирования фрейма
                                                /* ответа */
{
    FR[0]=0xAA; /* Преамбула */
    FR[1]=MyAddress; /* Адрес станции */
}

```

```

FR[2]=Stat;          /* Статус или ошибка */
FR[3]=B2;           /* Данные или детализация состояния */
FR[4]=B3;           /* Данные или детализация состояния */
FR[5]=(FR[0]+FR[1]+FR[2]+FR[3]+FR[4])&0x00FF;
/* Контрольная сумма */
}

void Form_Comm_Error (void)          /* Подпрограмма формирования фрейма ответа
при получении */
{
    Output_Frame (0x02, 0x00, 0x01); /* несуществующей команды. Код 0x01 */
}

/*****
***      Command Subroutines      ***
*****/
int Command_0 (void)                /* Подпрограмма команды установки
скорости 2400 */
{
    SpeedInit (StartSpeed);         /* Установить скорость 2400 */
    Output_Frame (2400,0,0);         /* Сформировать ответный фрейм */
    Frame_Send ();                  /* Отправить фрейм */
}

/* ..... Ряд команд пропущено */
int Command_F (void)                /* Подпрограмма пустой команды */
{
    /* Тело подпрограммы */

return OK;
}

/*****
***      M O N I T O R      ***
*****/
int Monitor (void)                  /* Подпрограмма монитора */
{
    switch (FR[2])                  /* Дешифровать код команды
{
    case 0x00: return Command_0 (); /* Вызвать команду 0 */
    case 0x01: return Command_1 (); /* Вызвать команду 1 */

/*..... Ряд кодов команд пропущено */

    case 0x0f: return Command_F (); /* Вызвать команду F */

/*..... Ряд кодов команд пропущено */

    default: return ERROR;         /* При поступлении кода несуществующей команды
вернуть код ошибки */
}
}

```

В приведенном фрагменте приведен текст подпрограммы только одной команды, остальные команды не приводятся из-за ограниченного объема. То же касается и подпрограммы монитора. Напомним, что в приведенном формате фрей-

ма для кода команды отведен один байт, т. е. всего может быть до 255 команд. Получение фрейма с ошибочной командой вызовет формирование фрейма ошибки (в нашем примере 0x01). Аналогично могут быть написаны и другие подпрограммы формирования ошибочных фреймов. Их формирование может быть вызвано прерываниями или опросом каких-либо портов.

В заключение приведем фрагмент 4, содержащий главный модуль программы.

Фрагмент 4:

```

/*****
/**   MAIN BEGINING   **/
/*****
void main (void)                /* Главный модуль программы */
{
    Init_Ports ();              /* Начальная инициализация микроконтроллера */
    while (1)
    {
        Frame_Get();           /* Получить фрейм команды */
        if (Monitor () == ERROR) /* Выполнить подпрограмму монитора */
        {                       /* Если монитор вернул ошибку */
            Form_Comm_Error (); /* Сформировать фрейм ошибки */
            Frame_Send ();      /* Отправить фрейм (ошибки) */
        }
    }
}
/*****   END   *****/

```

Приведенная программа может послужить основой для написания читателем собственной рабочей программы.

3. Схемотехника универсальных технологических контроллеров

3.1. Предпосылки создания универсальных технологических контроллеров

Развитие современных технологий, производства, измерительной техники, техники научных исследований и бытовой техники немислимо без использования средств вычислительной техники. Особое значение в этом процессе имеет микропроцессорная, а точнее было бы сказать, микроконтроллерная техника. Микроконтроллеры (или системы на их основе) можно сегодня обнаружить практически в любом современном изделии, начиная от наручных электронных часов и заканчивая сложными вычислительными системами. Наиболее сложные микроконтроллерные системы используются в различных технологических системах и технике научных исследований. Микроконтроллеры или системы микроконтроллеров, предназначенные для автоматизации производственных процессов или научных исследований, принято называть *технологическими микроконтроллерами*.

Рассмотрим, какие факторы оказывают наибольшее влияние на развитие современных *технологических микроконтроллерных систем (TMCS)*:

1. Важнейшими факторами развития *TMCS* является научная проработка архитектуры и схемотехники распределенных микроконтроллерных систем, разработка и промышленный выпуск многих периферийных узлов в интегральном исполнении, развитие средств и приемов программирования и отладки.

2. В настоящее время отмечается быстрое снижение цен на персональные компьютеры (*PC*) при значительном повышении их вычислительной мощности и надежности. Это позволяет включать *PC* в состав различных *TMCS* в качестве мощных систем хранения, документирования и визуализации информации на самом верхнем уровне управления.

3. Значительно расширилась номенклатура выпускаемых микроконтроллеров, а их стоимость постоянно уменьшается и приближается к стоимости обычных цифровых микросхем. При этом многие из микроконтроллеров характеризуются достаточно высокой производительностью, соизмеримой с производительностью *PC* на базе микропроцессоров *Pentium-II*, малыми потребляемыми токами при достаточно больших токах нагрузки. Многие микроконтроллеры имеют встроенную электрически программируемую *Flash*-память программ и энергонезависимую память данных, последовательные высокоскоростные каналы передачи данных, несколько таймеров и развитую систему прерываний. В некоторые микроконтроллеры встроены 10—16-разрядные аналого-цифровые и цифроаналоговые преобразователи, аналоговые компараторы, развитая система защиты от зависаний *WDT* и другие полезные возможности.

4. Во всем мире наблюдается глобальная тенденция к реструктуризации крупных производственных и научных учреждений и созданию на их основе малых производственных фирм и научно-исследовательских центров. В то же время отмечается снижение или полное прекращение объемов централизованного финансирования автоматизации производства и научных исследований. Все это приводит к еще большему ускорению процессов автоматизации, т. к. автоматизированные производства снижают конечную стоимость продукции (или экспериментов). В то же время стремление к снижению затрат на автоматизацию побуждает к использованию универсальных ТМСС, которые могут быть перенастроены на новые задачи быстро и с минимальными затратами.

Перечисленные выше основные факторы привели к ускоренному развитию одного из направлений современной микропроцессорной техники — МСС. Рассмотрим более подробно основные требования к современным МСС.

3.2. Основные понятия и тенденции развития ТМСС

Практически во всех работах, посвященных архитектуре современных ТМСС для малых систем автоматизации производства и научных исследований, отмечаются следующие основные понятия и истекающие из них требования:

1. Современное понятие *децентрализации* системы предполагает и территориальное рассредоточение вычислительной системы, и разделение функций системы (измерений, первичной обработки и управления) на ряд слабосвязанных функций, и распределение этих функций между несколькими децентрализованными вычислителями [55, 56]. Таким образом, современные *децентрализованные* системы являются системами с множеством потоков команд и данных (ММД). Территориально рассредоточенные *децентрализованные* системы принято называть локальными вычислительными сетями LAN, а отдельные автономные элементы сети — станциями (Station). С другой стороны, каждая из станций также может выполнять несколько слабосвязанных функций одновременно и состоять из нескольких слабосвязанных вычислителей, т. е. быть *мультимикропроцессорной* (ММПС). На практике это означает, что сложные технологические микроконтроллерные системы могут состоять из ряда автономно работающих станций (микроконтроллеров), которые работают по различным программам и расположены в различных местах управляемого объекта. Связь между станциями осуществляется через LAN. В некоторых случаях, как правило, когда производительности одного микроконтроллера не хватает для решения определенного набора задач или когда задачи кардинально различаются, в одной станции может находиться не один, а несколько микроконтроллеров, решающих свои задачи. Из этого вытекают следующие основные требования:

- каждая станция должна иметь как минимум один, а лучше несколько современных сетевых интерфейсов;
- сложные станции должны обеспечивать возможность совместной и независимой работы нескольких микроконтроллеров на общие ресурсы (интерфейсы связи с LAN, память, периферию и т. д.).

2. Суть требования *минимальной стоимости*, вообще говоря, очевидна, однако само это требование неразрывно связано с такими понятиями, как *максимальная эффективность* и *минимальная избыточность*. Под понятием *максимальной эффективности*, как правило, понимается соотношение производительность — цена, однако в последнее время это понятие часто объединяется с понятием *минимальной избыточности* системы, при этом под избыточностью понимают как аппаратную избыточность (неиспользуемые узлы аппаратуры или излишние сигналы управления), так и временную (например передача излишней информации по сетям). Из изложенных требований следует, что разработчики *TMCS* должны решить две, казалось бы, противоречивые задачи, с одной стороны, обеспечение *минимальной избыточности* (что подразумевает наличие в станции только тех функциональных узлов, которые необходимы для решения конкретной задачи), с другой — обеспечение *максимальной эффективности* (т. е. получения при минимальной цене контроллера максимальных функциональных возможностей, другими словами, возможности, при необходимости, наращивания системы). Поставленные противоречивые задачи наиболее просто решаются при выполнении двух условий:

- установка необязательных функциональных узлов (выполненных, как правило, в виде отдельных специализированных микросхем) должна осуществляться на *сокет*ы (*панельки*) или слоты;
- станции *TMCS* должны строиться по принципу *модульности*.

3. *Модульность* — свойство территориально сосредоточенных элементов системы — станций, означающее конструктивное выполнение всех узлов в виде набора функционально законченных автономных модулей (плат), имеющих одинаковые конструктивный (разъемы) и электрический (набор сигналов) интерфейс с *магистралью* (*шиной*) [6, 57].

В *модульной TMCS* все модули (узлы) должны иметь одинаковые разъемы.

Все контакты разъемов (за исключением особых случаев) в *модульной* системе должны быть соединены параллельно в так называемую *магистраль* или *шину*.

Все сеансы обмена по магистрали должны подчиняться определенным правилам (иметь определенные временные диаграммы обмена, определенную номенклатуру сигналов, определенные электрические уровни сигнала и т. д.). Эти правила обычно называются стандартами магистрали. В настоящее время в *TMCS* часто используются модифицированные стандарты (с усеченным набором линий).

4. Кроме отмеченных выше преимуществ *модульных* систем имеется еще целый ряд полезных свойств:

- *модульность* позволяет легко заменять вышедшие из строя модули и заменять их работающими, т. е. повышается *ремонтпригодность* системы;
- понятие *модульности* связано с понятием *гибкости*. *Гибкость* — свойство *модульных* систем легко изменять структуру изделий за счет различного сочетания модулей. Кроме того, под *гибкостью* системы понимается возможность осуществления *модернизации* за счет независимого обновления (модернизации) каждого из модулей. Облегчается разработка новых систем и увеличивается срок морального старения за счет применения

комбинации новых и уже существующих модулей, повышаются регулярность структуры и, как следствие, контрольно-диагностические качества, обеспечивается простота переориентации системы на другой объект регулирования или исследования;

- *модульные* системы обладают важным свойством *наращиваемости*. Это свойство означает, что первоначально для модульной системы разрабатывается *ядро* (или центральный микроконтроллерный модуль с важнейшими функциональными узлами), а дальнейшее увеличение функциональных возможностей достигается за счет увеличения периферийных модулей с дополнительными функциональными возможностями;
- кроме того, понятие *модульности* связано с понятием *многофункциональность* — свойство *модульных ТМCS*, означающее, что система не имеет жесткой специализации и привязки к объекту регулирования или исследования и, соответственно, может выполнять разные функции.

5. Понятие *модульности* подразумевает и свойство *конструктивного единства* — понятие, означающее конструктивное единообразное исполнение всех модулей станции. Доминирующим стандартом в настоящее время является *EUROCARD (IEC48D, 1980)*, позволяющий создавать ряд вариантов плат как комбинацию размеров ширины (160 и 220 мм) и высоты (100 и 233,53 мм). Наиболее часто используются платы с размерами 220 × 233,53 мм² (*MULTIBUS-II*), 100 × 220 мм² (*EUROBUS, ESONE*) и 100 × 160 мм². Это позволяет использовать разнообразные стандартные корпуса, выпускаемые различными производителями, и *крейты* (многоплатные конструктивы для *магистрально-модульных систем*).

6. Развитие *магистрально-модульных систем* предполагает обеспечение принципа *максимальной универсализации*, состоящего в том, что каждый вводимый в систему модуль должен обеспечивать построение системы, удовлетворяющей требованиям максимального числа применений.

7. Магистрально-модульные *ТМCS* должны удовлетворять требованиям *простоты установки и эксплуатации*, а также обеспечивать достаточно простую модернизацию.

8. Современные *ТМCS* имеют свойство *интеллектуальности* — свойство большинства модулей, связанное с наличием в них микроконтроллеров или микропроцессоров, работающих по своей индивидуальной программе, уровень *интеллектуальности* можно легко увеличивать в уже работающей системе путем модификации программного обеспечения модулей.

9. Многие сложные *ТМCS* обладают свойством *мультимагистральности* — оно означает использование в одной станции одновременно несколько различных магистралей. Такое техническое решение широко используется и в современных *РС*, и во многих магистрально-модульных системах. Наиболее широкое распространение получили двухшинные системы. На каждой из магистралей расположены слабосвязанные микропроцессоры и микроконтроллеры различной мощности и разрядности, причем магистрали связаны через общий двухвходовый ресурс, чаще всего — оперативную память [6, 37, 46, 58]. В более простых и сетевых *ТМCS* понятие *мультимагистральности* часто трансформи-

руется в понятие *мультиинтерфейсности*, которое означает, что ядро (главный модуль *TMCS*) оснащается максимально возможным количеством интерфейсов (*RS232C*, *RS485*, *SPI*, *CAN*, магистральный интерфейс, параллельные оптически развязанные регистры, интерфейсы интеллектуальных жидкокристаллических индикаторов — *LCD* и т. п.).

10. Безусловным принципом при построении проблемно-ориентированных систем является принцип *достаточности производительности*. Достижение необходимой производительности обеспечивается, как правило, созданием *многоконтроллерных* систем в одной станции, причем кроме обеспечения достаточной производительности за счет параллельного выполнения независимых задач или частей одной задачи достигается повышение надежности путем обеспечения свойства *постепенной деградации* системы при отказах отдельных контроллеров за счет перераспределения задач, возрастает также *гибкость* системы.

11. Важным и очевидным принципом построения *магистральных* систем является принцип *асинхронности* и *минимальности количества сигналов управления*, соблюдение этого принципа упрощает протокол обмена между модулями, повышает надежность связи и быстродействие обмена.

12. Важнейшим понятием, используемым при разработке современных *TMCS*, является понятие *совместимости или преемственности*. При разработке *TMCS* рассматриваются понятия *архитектурной, программной и выводной совместимости*. При анализе *архитектурной совместимости* рассматривается архитектура различных семейств микросхем микроконтроллеров, которые используются (или могут быть использованы) в создаваемой *TMCS* (или семействе *TMCS*). Единство архитектуры используемых микросхем микроконтроллеров или незначительные отличия в архитектуре существенно повышают эксплуатационные и потребительские качества создаваемой системы. Под *программной совместимостью* подразумевается, что различные семейства микросхем микроконтроллеров, которые используются (или могут быть использованы) в создаваемой *TMCS* (или семействе *TMCS*), имеют очень близкую систему команд и по идеологии, и по мнемонике. При этом учитываются также *привычность* и *известность* архитектуры и системы команд семейств микроконтроллеров, психологический фактор *«привычки»* потенциального пользователя к определенным семействам микроконтроллеров [59], наличие, известность и доступность компиляторов и симуляторов языков высокого уровня (как правило, языка «С»). Зачастую при разработке *TMCS* разрабатывается только базовое программное обеспечение (*SW*) или прототип *SW*, а собственно рабочие программы для конкретных приложений дорабатывают пользователи системы. При этом фактор *«привычки»* для пользователя выходит на первое место по значимости при выборе той или иной *TMCS*. Под выводной совместимостью понимается, что у различных семейств микросхем микроконтроллеров, которые используются (или могут быть использованы) в создаваемой системе, функциональное назначение выводов микроконтроллеров совпадает или отличается очень незначительно.

На конец прошлого столетия в мире было создано несколько десятков тысяч различных микроконтроллеров. Около 70 % из них — узкоспециальные микро-

контроллеры, ориентированные на решение специальных задач. Около 10 % — морально устаревшие микроконтроллеры. Около 15 % — новые малоизвестные или избыточно мощные микроконтроллеры, для которых разработаны принципиально новые языки программирования, малодоступны документация, системы разработки и отладки ПО, программаторы и т. д. И только около 5 % микроконтроллеров и микропроцессоров являются широко распространенными и известными. Из них безусловно выделяются и по сей день успешно работают в большинстве ТМС семейства микропроцессоров и микроконтроллеров фирмы *Intel* (MCS-51, MCS-96, MCS-80/85, MCS-86/88 и т. д.) и многих других фирм, использующих архитектуру этих семейств для своих микросхем. В последние годы наибольшей популярностью, пользуются усовершенствованные новые микроконтроллеры фирм *Atmel* [60—62], *MAXIM* [63] и *Cygnal* [64—66].

Из всего сказанного в п. 12 следует ряд важнейших принципов проектирования ТМС:

- наибольшей известностью, популярностью, оснащенностью и доступностью средств разработки и отладки *SW* обладают семейства микроконтроллеров *x51*;
- семейства микроконтроллеров *x51* выпускаются в двух наиболее популярных и удобных для установки на *сокеты* корпусах *DIP20* (*AT89C1051/2051/4051*) и *DIP40* (*AT89C51/52/53/55/8252*) [60, 61]. Кроме того, при соответствующей разводке печатной платы возможна установка на одной плате обоих типов микроконтроллеров на выбор (*сокет DIP20* устанавливается внутри *сокеты DIP40*) [67]. Существует выводная совместимость этих микроконтроллеров с микроконтроллерами *AVR* — *DIP20* (*AT90S2313*) и *DIP40* (*AT90S4414/8515*). Таким образом, существует ряд микроконтроллеров двух типов (*x51* и *AVR*) в двух вариантах корпусов, устанавливаемых практически на одном и том же месте на печатной плате (естественно, с соответствующими параметрами и себестоимостью);
- между указанными выше типами существует также достаточно большая программная совместимость, особенно на языке высокого уровня «С»;
- имеющиеся архитектурные отличия семейств *x51* и *AVR* сосредоточены на уровне ядра микроконтроллера и для разработчика ТМС практически отсутствуют;
- все вышесказанное выше в настоящем пункте позволяет сделать вывод о том, что семейства микроконтроллеров *x51* и *AVR* идеально подходят для ТМС.

13. Как уже отмечалось выше, современные ТМС используются в *распределенных* (в пространстве) системах и наиболее часто используются в составе *командно-информационных сетей CI-LAN*. В них единственный *PC*, включенный в *LAN*, играет роль генератора заданий и получателя данных, осуществляет вторичную обработку, систематизацию, сохранение и визуализацию данных. Периферийные станции осуществляют съем аналоговой и дискретной информации с различных датчиков, ее предварительную обработку, управляют различными внешними объектами, производят тестирование своих подсистем и передают полученную информацию в *PC*.

14. Функционирование *TMCS* в составе *распределенных* технологических систем и локальных сетей означает территориальное разнесение станций. При этом при гальванической связи между контроллерами возможно возникновение паразитных токов по цепям питания и возникновение различного рода помех. Для исключения этих явлений необходимо обеспечить гальваническую (чаще всего оптическую) развязку сетевых интерфейсов.

15. Очевидно, что технологические контроллеры работают в условиях сильных *электромагнитных* и *электростатических* полей, генерирующих различного рода помехи. Они могут привести к сбоям программы и зависаниям *TMCS*. Кроме того, работа в составе *LAN* накладывает определенные требования на включение/выключение отдельных станций сети. При включении должен обеспечиваться гарантированный сброс и инициализация станции, а при выключении станция не должна нарушать работоспособность остальных станций и сети в целом. Из сказанного следует:

- станции *TMCS* должны иметь специальные цепи сброса питания — *супервизоры* (*SV — Power Supervisory*) и *схемы защиты от зависаний — охраняемые таймеры* (*WDT*);
- для управления сильноточным оборудованием станции должны быть оснащены гальванически (чаще всего оптически) программно-управляемые развязанные входы и выходы;

Очевидно, что технологические системы работают *в реальном времени*. Это обстоятельство предъявляет ряд требований к схемотехнике и элементной базе *TMCS*:

- желательно, чтобы станции *TMCS* были оснащены *таймерами реального времени* (*RTC*), работающими независимо от наличия питания, для временной привязки к реальным событиям и процессам;
- элементная база *TMCS* должна иметь максимально возможное быстродействие (минимальное время переключения);
- функциональные узлы (например оперативная память), наиболее интенсивно используемые в вычислениях, должны иметь высокое быстродействие и параллельный интерфейс;
- объем оперативной памяти должен быть достаточным для проведения первичной обработки и буферизации данных для снижения потерь времени на передачу данных за счет снижения интенсивности и объема передач.

16. Кроме дискретного управления технологическими объектами *TMCS* часто производят измерения аналоговых величин и выдают аналоговые управляющие воздействия. Как следствие:

- в состав ядра *TMCS* должен быть включен *аналого-цифровой преобразователь* (*ADC*) со входным коммутатором среднего быстродействия с максимально доступной разрядностью при разумно низкой стоимости. Аналого-цифровые преобразователи высокого быстродействия обычно включаются в состав плат расширения.
- в состав ядра *TMCS* должны быть включены несколько *цифроаналоговых преобразователей* (*DAC*) для аналогового управления различными устройствами.

17. К энергопотреблению элементной базы *TMCS* также предъявляются высокие требования. Использование элементной базы с низким потреблением снижает уровень помех, улучшает массогабаритные характеристики источников питания, снижает общую температуру элементов, повышает надежность функционирования системы.

18. Станции современных распределенных *TMCS* могут быть разнесены в пространстве на значительные расстояния. При этом наличие в станциях *локальных устройств визуализации* значительно облегчает настройку и диагностику системы, снижает информационные потоки по сети, увеличивает эксплуатационные качества. В качестве локальных устройств визуализации обычно используются либо интеллектуальные жидкокристаллические (*LCD*) или светодиодные (*LED*) индикаторы, либо устройства отображения информации на телевизионных (*TV*) индикаторах. *LCD*-индикаторы более предпочтительны, поскольку они более экономичны в плане энергопотребления. *LED*-индикаторы обычно используют, если *TMCS* ориентирована на работу в условиях ухудшенной видимости. *TV*-индикаторы используют в случае необходимости локального отображения больших динамически изменяющихся массивов информации. Из этого следует:

- в состав ядра *TMCS* должен быть включен *LCD (LED)* интерфейс;
- узел связи ядра микроконтроллера с магистралью должен быть оснащен средствами расширения адресного пространства для работы с памятью внешнего *TV*-индикатора. Такие узлы расширения называют диспетчерами памяти.

3.3. Технические требования к главному микроконтроллерному модулю *TMCS*

Анализ приведенных выше понятий и тенденций развития позволяет сформулировать требования к составу функциональных модулей и конструктивному исполнению центрального микроконтроллерного модуля *TMCS*.

1. В соответствии с причинами, приведенными в п. 12, центральный микроконтроллерный модуль (далее просто ядро) *TMCS* должен иметь возможность работать под управлением одного из следующих микроконтроллеров: *AT89C51/52/53/55/8252* и *AT90S4414/8515* в корпусе *DIP40* или *AT89Cх051* и *AT90S2313* в корпусе *DIP20*. В последнем случае допускается усечение функциональных возможностей модуля. Кроме того, допускается использование микроконтроллеров и в более современных корпусах при конструктивном исполнении их в виде вставных слотов с унификацией магистрали слота.

2. В соответствии с п. 15, ядро должно иметь супервизор питания и сторожевой таймер (*WDT*) для защиты от зависаний.

3. В соответствии с требованиями п. 16, ядро должно содержать быстродействующую оперативную память (*RAM*) с параллельным доступом достаточно большого объема.

4. В соответствии с требованиями п. 16, ядро должно содержать таймер реального времени (*RTC*), желательно с параллельным доступом и встроенной энергонезависимой памятью.

5. В состав ядра необходимо включить аналого-цифровой преобразователь (*ADC*) среднего быстродействия с разрядностью 12—16 двоичных разрядов и несколько цифроаналоговых преобразователей (*DAC*) в соответствии с п. 17. Желательно, чтобы они были подключены через отдельный дополнительный интерфейс (в качестве основного считается интерфейс с магистралью, *RAM* и *RTC*), например, последовательный приборный интерфейс (*SPI*), причем и основной, и дополнительный интерфейсы должны быть выведены на магистральный разъем (в соответствии с пп. 1, 3, 4, 9) или дополнительный штыревой разъем.

6. Очевидно, что центральный модуль должен иметь максимально возможное количество регистров ввода и вывода информации, причем часть из них должна иметь гальваническую развязку (по пп. 9, 15).

7. Ядро *TMCS* должно иметь максимально возможное количество последовательных интерфейсов (*RS232C*, *RS485*, *CAN*, *SPI* и т. д.) для работы в составе *CI-LAN* (по п. 1, 13), причем по крайней мере часть из них должна иметь оптическую изоляцию (по пп. 9, 14, 15).

8. Поскольку ядро ориентировано на работу в составе *CI-LAN* (по п. 1, 13), в его состав должен быть включен модификатор (перемычки или переключатели) сетевого адреса.

9. Желательно, чтобы ядро *TMCS* было оснащено интерфейсом для подключения интеллектуальных *LCD* или *LED* локальных индикаторов (по п. 19).

10. В качестве основного магистрального интерфейса ядра (с учетом выбранных типов микроконтроллеров) наиболее подходит усеченный модифицированный интерфейс *I41 (MULTIBAS-I)* [48]. Он должен иметь достаточное количество адресных линий (адресное пространство) для подключения дополнительных модулей *TMCS* (по пп. 4, 19), высокое быстродействие и минимальное количество сигналов управления (по пп. 10, 11).

11. Элементная база ядра должна иметь достаточно высокое быстродействие и низкое энергопотребление (в соответствии с пп. 16, 18).

12. Вся *TMCS* должна быть выполнена в соответствии с магистрально-модульными принципами (п. 3—8), т. е. все модули, включая ядро, должны иметь одинаковые размеры (100 × 220 мм² или 100 × 160 мм²) и магистральный разъем (*CNP59/96* или *C96*).

3.4. Обобщенная функциональная схема центрального микроконтроллерного модуля

Вышеперечисленные основные требования позволяют составить обобщенную функциональную схему центрального микроконтроллерного модуля *TMCS*, представленную на рис. 3.1.

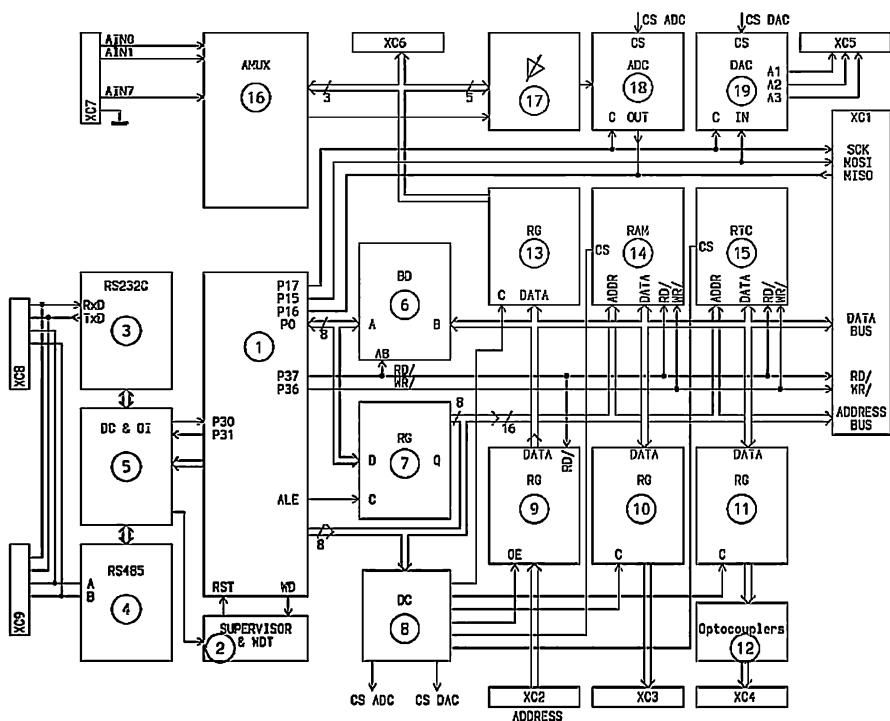


Рис. 3.1. Обобщенная функциональная схема центрального микроконтроллерного модуля *TMC5*

Функциональная схема содержит следующие основные узлы:

1. Сокеты для установки одного из следующих микроконтроллеров: *AT89C51/52/53/55/8252* и *AT90S4414/8515* в корпусе *DIP40* или *AT89C1051/2051/4051* и *AT90S2313* в корпусе *DIP20*.

2. Супервизор питания со схемой защиты от зависаний *WDT*. Выход супервизора соединен со входом *RST* микроконтроллера. Следует отметить, что полярность сигналов *RST* различна для микроконтроллеров серии *x51* и *AVR*, следовательно, супервизор должен обеспечивать выдачу сигналов сброса обеих полярностей и возможность выбора одной из них. Один из входов супервизора соединен с интерфейсом *RS232C* с целью обеспечения возможности сброса ядра по этому интерфейсу. На вход *WDT* должны периодически подаваться импульсы запуска от микроконтроллера.

3. Интерфейс *RS232C*.

4. Интерфейс *RS485*.

5. Логика управления интерфейсами с оптической изоляцией одного или обоих из них. Она связана с интерфейсами 3 и 4 и микроконтроллером 1. Количество связей с микроконтроллером должно быть минимальным, при этом функ-

циональные возможности управления интерфейсами желательно иметь максимальными.

6. Шинный формирователь данных магистрали.

7. Регистр защелки младшего байта адреса.

8. Дешифратор адреса.

9. Регистр ввода значений модификатора *XC2*. Модификатор *XC2* (и все остальные разъемы, кроме *XC1*) должен быть выполнен в виде штыревого разъема типа *PLD*. Это позволяет устанавливать на него перемычки для установки адреса, а если модификатор адреса не используется или используется не полностью, на разъем может устанавливаться плоский кабель и входы разъема могут использоваться для других целей.

10. Регистр вывода гальванически связанных данных через штыревой разъем *XC3*.

11. Регистр вывода гальванически связанных данных через штыревой разъем *XC4* или оптически изолированных данных через узел оптронов 12 и штыревой разъем *XC4*. На четные контакты штыревого разъема *XC4* данные поступают непосредственно с выходов регистра 11, на нечетную — через узел оптронов 12. Такое решение позволяет по необходимости выбирать произвольную комбинацию изолированных и неизолированных сигналов.

12. Узел оптронов.

13. Регистр управления функциями входного аналогового коммутатора 16 и конфигурируемого операционного усилителя 17.

14. Быстродействующая оперативная память с параллельным доступом.

15. Таймер реального времени с параллельным доступом и встроенной энергонезависимой оперативной памятью.

16. Входной аналоговый мультиплексор для коммутации входных сигналов аналого-цифрового преобразователя *ADC* (18).

17. Конфигурируемый операционный усилитель с возможностью выбора polarity сигнала и коэффициента усиления.

18. Среднескоростной аналого-цифровой преобразователь *ADC* с разрядностью 12—16 двоичных разрядов и связью по последовательному каналу *SPI*.

19. Сдвоенный или встроенный цифроаналоговый преобразователь *DAC* с разрядностью 10—8 двоичных разрядов и связью по последовательному каналу *SPI*.

Кроме перечисленных функциональных узлов на функциональной схеме показаны разъемы: *XC1* — разъем магистрали (может быть в метрическом *СНП59/96* или дюймовом *С96* исполнении); *XC2* — штыревой разъем *PLD-16*, используемый как модификатор адреса или для подачи внешних входных сигналов; *XC3* — штыревой разъем *PLD-16*, используемый для выдачи выходных оптически не развязанных сигналов; *XC4* — штыревой разъем *PLD-20*, используемый для выдачи выходных оптически не развязанных или развязанных сигналов; *XC5* — штыревой разъем *PLD-6*, используемый для выдачи выходных аналоговых сигналов с *DAC* (19); *XC6* — штыревой разъем *PLD-16*, используемый для выдачи выходных сигналов регистра 13 в случае, если мультиплексоры 16 и конфигурируемый операционный усилитель 17 не используются; *XC7* —

штыревой разъем *PLD-16*, используемый для подачи входных аналоговых сигналов на *ADC* (18) через мультиплексор 16 и конфигурируемый операционный усилитель 17; *XC8* и *XC9* — два параллельно соединенных штыревых разъема *PLD-10*, используемых для подключения к интерфейсам *RS232C* и *RS485*.

Следует отметить, что ряд сигналов управления на обобщенной схеме не указан, поскольку их окончательное назначение и источник их выдачи будут определены после выбора типов и схем включения всех узлов функциональной схемы (например, не указаны оптически развязанные входы, входы прерываний, сигналы управления интерфейсами и т. д.). Таким образом обобщенная функциональная схема в некотором смысле является информацией к размышлению.

В следующих разделах мы остановимся на схемотехнике каждого из узлов, выработаем критерии выбора, выберем оптимальные варианты, приведем принципиальные схемы и в заключение цикла получим полную принципиальную схему центрального микроконтроллерного узла.

3.5. Супервизоры питания и охранные таймеры

3.5.1. Основные понятия о супервизорах питания и охранных таймерах

Прежде чем начать разговор о предмете этого раздела, попытаемся дать определение рассматриваемого предмета.

Супервизором питания (*Power Supervisory Circuits*) или *Монитором* (*Monitor*) называется электронная схема (далее — *SV*), предназначенная для выполнения функции начального сброса (или инициализации) микроконтроллера (микропроцессора) при включении питания или снижении напряжения питания ниже определенного значения в процессе работы. При этом длительность сигнала сброса должна быть больше определенной оговоренной величины. Полярность сигнала сброса должна соответствовать полярности входа сброса микроконтроллера (микропроцессора). Супервизор должен иметь дополнительный вход для сброса системы от дополнительной кнопки сброса. В некоторых случаях супервизоры имеют дополнительные входы, анализирующие другие напряжения питания. Кроме того, супервизоры могут формировать сигналы прерывания при различных ситуациях.

Охранным (сторожевым) таймером (*WDT — WatchDog Timer*) называется специализированная схема защиты микроконтроллерной (микропроцессорной) системы от программных зависаний. Идея защиты достаточно проста. Микроконтроллер периодически, не реже определенного фиксированного временного интервала подает импульс перезапуска таймера (далее импульс *WD*) на вход схемы защиты. Импульс подается через дифференцирующую цепь для исключения потенциального влияния на схему. Схема представляет собой ждущий одновибратор с фиксированной задержкой выработки выходного импульса сброса (далее *RST*) и фиксированной длительностью этого импульса. Если период поступления импульсов *WD* не превышает величины фиксированной задержки,

выходные импульсы *RST* не вырабатываются, т. к. происходит постоянный перезапуск сторожевого таймера.

Микросхемы супервизоров питания выпускаются многими известными фирмами мира: *Analog Devices* [69], *Dallas Semiconductors* [70], *Electronic Technology* [71], *IMP* [72], *Linear Technology* [73], *MAXIM* [74], *Microchip* [75], *Sipex* [76], *Telcom* [77], *Texas Instruments* [78]. Кроме основной функции — собственно супервизора питания, многие микросхемы оснащаются охранным таймером *WDT*, а также рядом дополнительных функций. В зависимости от количества встроенных функциональных узлов, микросхемы выпускаются в самых разнообразных корпусах.

3.5.2. Критерии выбора супервизора питания для универсальных технологических контроллеров

Перечислим основные и возможные функции (свойства) микросхем супервизоров питания:

- 1) выработка сигнала *RST* при включении питания;
- 2) выработка сигнала *RST* при снижении питания ниже определенного значения (допуска) в процессе работы для исключения возможности неправильного функционирования;
- 3) наличие выходов сигнала *RST* одной или обеих полярностей;
- 4) наличие входа для внешней кнопки сброса;
- 5) наличие компаратора раннего предупреждения (о снижении питания);
- 6) наличие встроенного сторожевого таймера *WDT*;
- 7) возможность программирования (подбором резисторного делителя) порогового напряжения (допуска) выработки сигнала *RST*;
- 8) возможность программирования (подбором конденсатора) длительности сигнала *RST*;
- 9) возможность программирования (подбором конденсатора) периода срабатывания *WDT*;
- 10) наличие схемы подключения резервного питания (аккумулятора);
- 11) наличие схемы контроля заряженности источника резервного питания;
- 12) наличие схемы подзарядки источника резервного питания;
- 13) наличие одного или нескольких входов и выходов выборки (*CS*) памяти, блокирующих работу памяти при снижении напряжения питания ниже определенной величины (допуска);
- 14) наличие встроенной системы звуковой индикации снижения питания;
- 15) наличие встроенной светодиодной системы индикации снижения питания;
- 16) наличие входа или входов для анализа дополнительных напряжений питания (например +12 В, -12 В, и т. д.);
- 17) наличие выхода или выходов сигналов прерывания при снижении питания;
- 18) наличие отдельного выхода от *WDT* или объединение этого выхода с выходами сигнала *RST*;

19) исполнение микросхемы *SV* в различных типах корпусов (*TO92*, *SOT223*, *SOT23*, *DIP8*, *SO8*, *В μ SOP8*, *DIP14*, *SO14*, *DIP16*, *SO16*, *DIP20*, *DIP28*, *PLCC28* и т. д.).

Анализируя приведенные выше свойства микросхем *SV*, становится ясно, что многие из них являются избыточными при решении нашей задачи (разработке универсальных технологических контроллеров). Определим критерии — главные (обязательные) и желательные качества, по которым мы будем выбирать микросхемы *SV*:

1. Микросхема *SV* должна вырабатывать сигнал *RST* определенной длительности для инициализации микроконтроллера (микропроцессора) при включении питания. Желательна возможность программирования длительности сигнала *RST* и наличие нескольких вариантов допусков напряжения срабатывания.

2. Микросхема *SV* должна вырабатывать *RST* при снижении напряжения питания ниже определенного значения (допуска) в процессе работы.

3. При разработке универсальных технологических контроллеров, в которых возможно применение микроконтроллеров различных типов (с различной полярностью сигнала *RST*), желательно, чтобы микросхема *SV* обеспечивала генерацию сигнала *RST* обеих полярностей.

4. С целью снижения площади печатной платы, снижения стоимости и обеспечения требований высокой ремонтпригодности необходимо, чтобы микросхема *SV* была выполнена в корпусе *DIP8*.

5. Микросхема *SV* должна иметь дополнительный вход для сброса системы от дополнительной кнопки сброса.

6. Желательно, чтобы микросхема *SV* имела встроенный *WDT*. Желательно, а не обязательно потому, что наличие встроенного *WDT* повышает стоимость микросхемы. В то же время многие современные микроконтроллеры имеют встроенный *WDT*, при этом *WDT* в микросхеме *SV* может оказаться избыточным. К тому же *WDT* можно достаточно просто организовать с помощью нескольких внешних (по отношению к *SV*) резисторов и конденсаторов при наличии у *SV* входа для сброса системы от дополнительной кнопки сброса.

Перечисленные критерии позволяют осуществить выбор микросхем *SV* из всего многообразия выпускаемых микросхем. В табл. 3.1 приведены данные простых микросхем *SV*, выпускаемых в корпусах до *DIP16* и имеющих небольшой набор дополнительных функций. С полными перечнями микросхем *SV* и их документацией можно ознакомиться на сайтах производителей, приведенных в литературе.

Таблица 3.1

Тип микросхемы	Напр.пит., В	RESIN	Вых. сигнала RST	Тип корпуса	WDT
ADM1232, DS1232, ETC1232, IMP1232, LTC1232, MAX1232, TC1232	5	+	Low, High	DIP8, SO16, SO8, В μ SOP8	+
ADM1812, DS1812, MPC101-xxx	5	—	High	TO92, SOT23	—
ADM1813, DS1813	5	+	Low	TO92, SOT23	—
ADM1815, DS1815, MPC100-xxx, TC54VCxx	3,3	—	Low	TO92, SOT23	—

Продолжение таблицы 3.1

Тип микросхемы	Напр.пит., В	RESIN	Вых. сигнала RST	Тип корпуса	WDT
ADM1816, DS1816, MPC130-xxx	3,3	–	Low	TO92, SOT23	–
ADM1817, DS1817, MPC101-xxx	3,3		High	TO92, SOT23	–
ADM1818, DS1818	3,3	+	Low	TO92, SOT23	–
ADM705, DS1705, ETC705x, IMP705x, MAX705x, SP705x	5	+	Low	DIP8, SO8, ВμSOP8	+
ADM706, DS1706, ETC706, IMP706x, MAX706x, SP706x	5–3,3	+	Low / High	DIP8, SO8, ВμSOP8	+
ADM707, DS1707, ETC707, MAX707x, SP707x	5	+	Low, High	DIP8, SO8, ВμSOP8	–
AMD1810, DS1810, MPC100-xxx, TC54VCxx	5	–	Low	TO92, SOT23	–
AMD1811, DS1811, MPC130-xxx	5	–	Low	TO92, SOT23	–
AMD708x, DS1708, ETC708, MAX708x, SP708x	5–3,3	+	Low, High	DIP8, SO8, ВμSOP8	–
DS1231	5	–	Low, High	DIP8, SO16	–
DS1233, IMP1233, MPC130	5–3,3	+ / –	Low	TO92, SOT223	–
DS1236	5	+	Low, High	DIP16, SO16	+
DS1238	5	+	Low, High	DIP16, SO16	+
DS1239	5	+	Low	DIP16, SO16	–
DS1632	5	+	Low, High	DIP16, SO16	–
DS1814, MAX823, MAX824, MAX825, TPS3823, TPS3824, TPS3825	5	+ / –	Low / High	SOT23-5	+ / –
DS1819, MAX823, MAX824, MAX825, TPS3823, TPS3824, TPS3825	3,3	+ / –	Low, High / Low	SOT23-5	+ / –
DS1832	3,3	+	Low, High	DIP8, SO8	+
DS1833	5	–	High	TO92, SOT223	–
DS1834	5–3,3	+	Low/ High	DIP8, SO8, ВμSOP8	–
DS1836	5–3,3	–	Low/ High	DIP8, SO8	–
MAX801	5	–	Low, High	DIP8	+
MAX814	5–3,3	+	Low, High	DIP8	–
MAX816	5	+	Low, High	DIP8	–
MAX700	5	+	Low, High	DIP8	–
MAX701	5	+	Low, High	DIP8	–
UCC1946, UCC2946, UCC3946	<5	+	Low	DIP8	+
TL7705	5	+	Low, High	DIP8	–
TLC7701, TLC7725, TLC7703, TLC7733, TLC7705	2–6 V	+	Low, High	DIP8	–
TL7759	1–7 V	–	Low, High	DIP8	–

3.5.3. Предварительный анализ и выбор микросхем SV

Из приведенных в таблице типов микросхем SV можно отобрать всего несколько типов, удовлетворяющих приведенным выше критериям (см. табл. 3.2).

Таблица 3.2

Тип микросхемы	Напр.пит., В	RESIN	Вых. сигнала RST	Тип корпуса	WDT
ADM1232, DS1232, ETC1232, IMP1232, LTC1232, MAX1232, TC1232	5 В	+	Low, High	DIP8, SO16, SO8, ВμSOP8	+
ADM707, DS1707, ETC707, MAX707x, SP707x	5 В	+	Low, High	DIP8, SO8, ВμSOP8	–
AMD708x, DS1708, ETC708, MAX708x, SP708x	5–3,3 В	+	Low, High	DIP8, SO8, ВμSOP8	–
DS1819, MAX823, MAX824, MAX825, TPS3823, TPS3824, TPS3825	3,3 В	+ / –	Low, High / Low	SOT23-5	+ / –
MAX814	5–3,3 В	+	Low, High	DIP8	–
MAX816	5 В	+	Low, High	DIP8	–
MAX700	5 В	+	Low, High	DIP8	–
MAX701	5 В	+	Low, High	DIP8	–
TL7705, TLC7705, TLC7733, TLC7703, TLC7725	5 В	+	Low, High	DIP8	–

Как видно из табл. 3.2, все отобранные типы микросхем выпускаются фирмами *MAXIM* и *Texas Instruments*, поэтому далее мы будем ссылаться на техническую документацию этих фирм. В табл. 3.3 приведены назначения выводов микросхем отобранных типов.

Таблица 3.3

Тип ИМС SV	Функциональное назначение выводов								Ссылка, PDF
	1	2	3	4	5	6	7	8	
MAX824	RST/	GND	RST	WDI	VCC	–	–	–	1249[74]
MAX825	RST/	GND	RST	RSTIN/	VCC	–	–	–	1249[74]
MAX707, MAX708	RSTIN/	VCC	GND	PFI	PFO/	–	RST/	RST	1151[74]
MAX814	RSTIN/	VCC	GND	PFI	PFO/	LOWL/	RST/	RST	1196[74]
MAX816	RSTIN/	VCC	GND	PFI	PFO/	RSTIN	RST/	RST	1196[74]
MAX700	RSTIN/	SENSE	HYST	GND	RST/	RST	CTL	VCC	1383[74]
MAX701	RSTIN/	–	–	GND	RST	RST/	–	VCC	1383[74]
MAX1232	RSTIN/	TD	TOL	GND	RST	RST/	WDI/	VCC	1286[74]
TL7705	Cref	RSTIN/	Ct	GND	RST/	RST	SENSE	VCC	Slvae04[78]
TLC7705, 7733, 7703, 7725	Cref	RSTIN/	Ct	GND	RST/	RST	SENSE	VCC	Slvs087k[78]

В табл. 3.3 приняты следующие обозначения:

RESIN/ — инверсный вход кнопки сброса;

TD — вход программирования периода WDT (150 мс / 600 мс) (MAX1232);

TOL — вход программирования допуска SV (4,5 В / 4,75 В) (MAX1232);

GND — общий вывод питания;

RST — прямой выход сигнала сброса;

RST/ — инверсный выход сигнала сброса;

WDI/ — вход перезапуска WDT;

VCC — вход напряжения питания

CTL — вход управления контроля питания (MAX700). Если «GND» — контроль напряжения питания, если «VCC» — контроль входа SENSE;

SENSE — дополнительный вход контроля питания;

HYST — при контроле входа SENSE (MAX700) можно ввести дополнительную задержку перед выработкой сигналов сброса длительностью 5 мкс путем включения резистора номиналом 1 кОм между выводами HYST и GND;

PFI — вход для анализа дополнительного источника питания (MAX707, MAX708) с помощью внешнего резистивного делителя (сравнение с внутренним опорным напряжением 1,25 В);

PFO/ — выход прерывания, свидетельствующего о снижении анализируемого по входу PFI напряжения (MAX707, MAX708);

RSTIN — дополнительный вход анализа питания (MAX816);

LOWL/ — выход прерывания (MAX816), свидетельствующий о снижении напряжения питания.

При внимательном рассмотрении данных, приведенных в табл. 3.3, можно разделить приведенные микросхемы SV на 3 группы по признаку максимального совпадения функциональных выходов. Эти группы показаны на рис. 3.2:

1) MAX824, MAX825 и их аналоги;

2) MAX707, MAX708, MAX814, MAX816 и их аналоги;

3) MAX1232, MAX700, MAX701, TL(C)7705 и их аналоги.

Анализ приведенных трех групп микросхем SV и содержимого таблицы 3.2 позволяет сделать заключение о количестве микросхем различных производителей, входящих в каждую группу:

1) в первую группу входят 5 микросхем (DS1819, MAX824, MAX825, TPS3824, TPS3825);

2) во вторую группу входят 12 микросхем (ADM707, DS1707, ETC707, MAX707, SP707, AMD708, DS1708, ETC708, MAX708, SP708, MAX814, MAX816);

3) в третью группу входят 14 микросхем (ADM1232, DS1232, ETC1232, IMP1232, LTC1232, MAX1232, TC1232, MAX700, MAX701, TL7705, TLC7705, TLC7733, TLC7703, TLC7725).

В первую и третью группы входят по одному типу микросхем SV со встроенным WDT (MAX824 и MAX1232), что является безусловным минусом для второй группы, в которой таких микросхем нет.

Следует остановиться также на стоимости микросхем различных групп. Стоимости микросхем приведены по данным производителя MAXIM [74] при по-

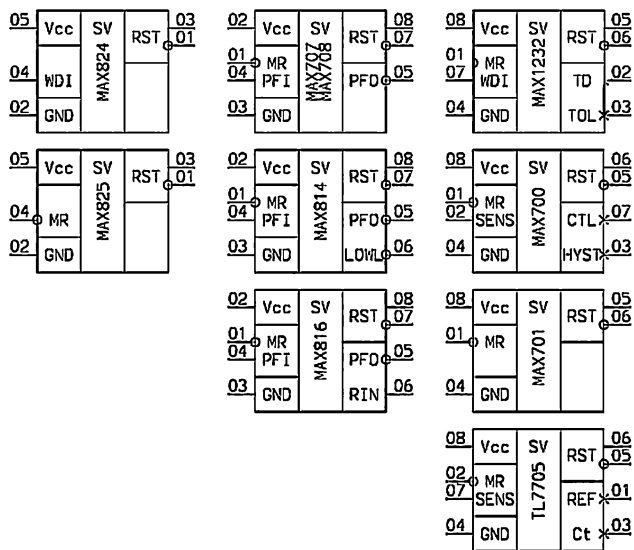


Рис. 3.2. Функциональные изображения выбранных микросхем супервизоров

ставках партиями (см. табл. 3.4), т. е. розничные цены на микросхемы будут несколько выше.

Таблица 3.4

Тип	MAX824/825	MAX707/708	MAX814/816	MAX1232	MAX700/701	TL(C)7705
Цена, \$	1,20	0,88	4,39	1,71	2,17	0,55

3.5.4. Схемы включения микросхем SV

Рассмотрим принципиальные схемы включения для каждой из групп микросхем SV.

Первая группа, содержащая два типа микросхем, отличающихся функциональным назначением только одного вывода, показана на рис. 3.3.

На рис. 3.3 показаны два варианта включения SV MAX824/825. Первый вариант схемы подразумевает, что SV всегда должен иметь функции WDT. Элементы C1, C2, R1, R2, VD1 реализуют функции WDT. Конденсатор C1 (10н—33н) разделительный и необходим для исключения влияния какого-либо из логических уровней на выходе WDI микроконтроллера. Резистор R1 (100—220 кОм) необходим для разряда конденсатора C1. Диод VD1 обеспечивает прохождение положительных импульсов на времязадающий конденсатор C2 (1,0—2,2 мкФ). Резистор R2 обеспечивает разряд конденсатора C2 в интервалах между импульсами WDI. Переключки JP2 необходимы для выбора необходимой полярности импульсов сброса RST. Символ «*» после наименования сигнала RST обозначает, что полярность сигнала может быть различной. При использовании микро-

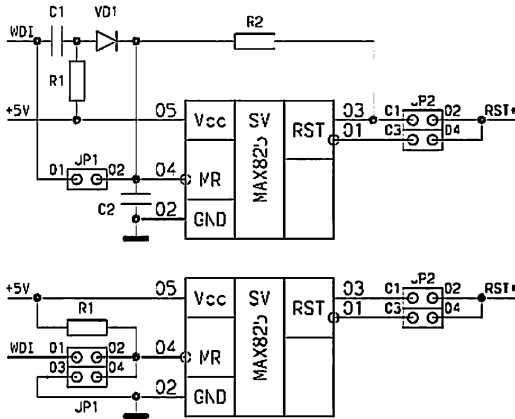


Рис. 3.3. Схемы включения микросхем SV первой группы

схемы MAX825 все элементы, показанные на первой схеме рис. 3.3 должны быть установлены, перемычка *JP1* не устанавливается, при использовании микросхемы MAX824 (имеющей встроенный WDT) элементы *C1*, *C2*, *R1*, *R2*, *VD1* не устанавливаются, а перемычка *JP1*, наоборот, устанавливается и обеспечивает поступление сигнала *WDI* на соответствующий вход MAX824.

Второй вариант схемы на рис. 3.3 обеспечивает установку либо MAX824 с функциями WDT (установлена перемычка *JP1*{1—2}), либо MAX825 без WDT, но с дополнительной внешней кнопкой сброса, подключаемой на перемычки *JP1*{3—4} при этом перемычка *JP1*{1—2} не устанавливается. Очевидно, что установкой соответствующей перемычки *JP2* выбирается необходимая полярность сигнала *RST*.

Вторая группа микросхем SV содержит четыре типа микросхем, которые не имеют встроенного WDT. У этих микросхем все основные выводы совпадают. Поэтому, если не принимать во внимание возможное использование функциональных дополнительных возможностей (соответственно и дополнительных входов/выходов), схемы включения этой группы микросхем одинаковые и представлены на рис. 3.4.

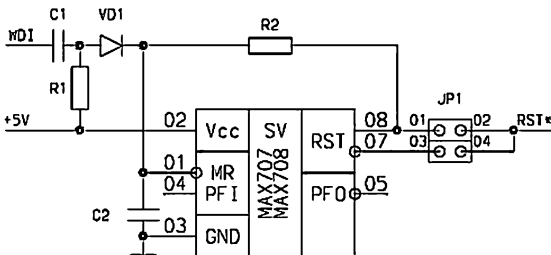


Рис. 3.4. Схема включения микросхем SV второй группы

Функциональное назначение элементов на рис. 3.4 совпадает с назначением элементов на первом рис. 3.3.

Третья группа микросхем *SV* содержит четыре типа микросхем, которые отличаются достаточно большим количеством выводов. Практически у этой группы полностью совпадают только выводы питания (*Vcc* и *GND*). Выводы выходов *RST* хотя и выведены на два вывода (5, 6), но имеют различное расположение для различных микросхем. К тому же вход компаратора для подключения внешней кнопки, который используется для построения внешнего *WDT*, выведен на различные выводы (1 и 2). Конечно, и при этих обстоятельствах можно обеспечить возможность установки на плату всех этих типов микросхем *SV*, но реализуется это за счет достаточно большого количества навесных элементов, которые хотя и имеют малую стоимость, но занимают достаточно большую площадь на печатной плате. Такой вариант схемы включения показан на рис. 3.5.

Функциональное назначение элементов *C1*, *C2*, *R1*, *R2*, *VD1*, *JP1* описано выше. Поскольку выводы *RST* отдельных микросхем выполнены с открытым коллектором или истоком, для работы *SV* необходимо подключение соответствующих резисторов на питание или общий провод. Для этих целей служат резисторы *R4*, *R5* (номиналы должны соответствовать минимальному сопротивлению на входе *RST* микроконтроллера, например для *AT89C5x* — 56 К). Соответствующее подключение обеспечивается установкой переключателей *JP2*, *JP3*. Одной из переключателей *JP4* резистор *R2* подключается к выводу сигнала *RST* положительной полярности. Остальные элементы (*JP5*, *C3*, *C4*, *R3*) необходимы для обеспечения всех возможных вариантов включения всех *SV*.

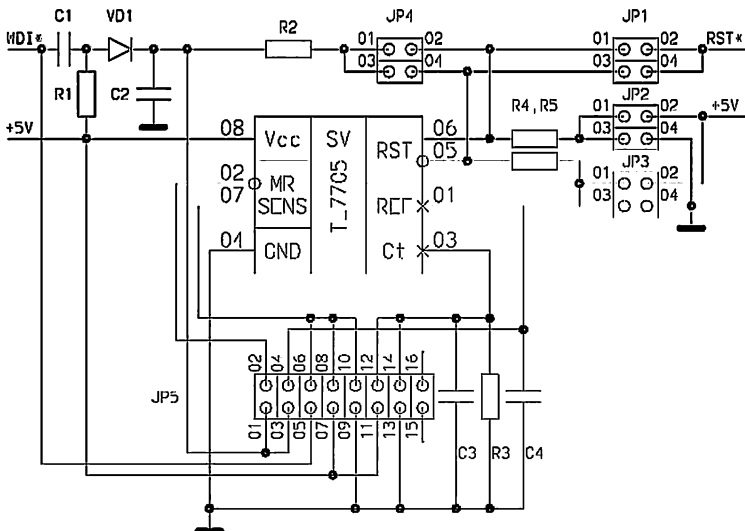


Рис. 3.5. Схема включения всех микросхем *SV* третьей группы

Очевидно, что возможно разделение этой группы на более мелкие подгруппы по совпадению полярности выходных сигналов *RST*. В первую такую подгруппу можно отнести микросхемы *MAX1232* и *MAX701*, а во вторую — *MAX700* и *TL7705* (с соответствующими аналогами других производителей). Такое разделение позволяет существенно упростить схемы включения (они не приводятся, т. к. достаточно очевидны). Покажем лишь наиболее простой и низкостойимый вариант включения *SV* с функциями *WDT* на микросхеме *TL7705* (*TLC7705* и аналоги) — см. рис. 3.6.

Функциональное назначение элементов *C1*, *C2*, *R1*, *R2*, *VD1*, *JP1* соответствует вышеописанному. Резисторы *R3*, *R4* обеспечивают нагрузку на открытые коллекторы выходных транзисторов. Конденсатор *C3* определяет длительность сигнала сброса. Конденсатор *C4* предназначен для работы внутреннего источника опорного напряжения. Следует заметить, что приведенная схема при низкой стоимости микросхемы позволяет реализовать все необходимые режимы и настраивать все параметры подбором величин *C2*, *C3* и *R2*.

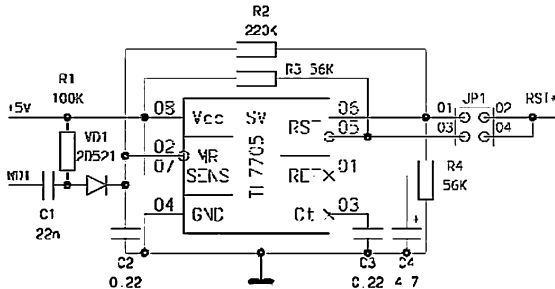


Рис. 3.6. Схема включения микросхемы *SV TL7705* с реализацией функции *WDT*

3.5.5. Заключительный этап выбора микросхем *SV*

До сих пор мы занимались рассмотрением технических аспектов выбора микросхем *SV* для универсальных технологических контроллеров. На заключительном этапе вступают в силу уже иные дополнительные критерии выбора (экономические, конструкционные, организационные), такие как:

- 1) стоимость реализации узла *SV* с возможностью организации *WDT*;
- 2) количество дополнительных элементов;
- 3) количество связей и модификаторов (перемычек);
- 4) площадь печатной платы, занимаемая узлом *SV*;
- 5) доступность элементов и сроки их поставки.

Из соображений минимальной стоимости наиболее привлекательны варианты реализации узлов *SV* на микросхемах *TL7705* (~\$1,0), *MAX824/825* (~\$1,3), *MAX707/708* (~\$1,6).

Количество дополнительных элементов минимально при реализации *SV* на микросхемах *MAX824/825*. В этом же случае получается минимальным количество связей и площадь печатной платы.

Наиболее доступными являются микросхемы TL7705, они практически всегда есть в наличии во всех специализированных центрах торговли. К сожалению, этого нельзя сказать о микросхемах фирмы MAXIM, сроки поставки многих микросхем достигают нескольких недель...

Таким образом, учитывая все приведенные доводы, следует сделать вывод о том, что наиболее оптимальным узлом SV с реализацией функций WDT является узел на микросхеме TL7705 (см. рис. 3.6), поскольку он полностью отвечает как основным — техническим, так и дополнительным — организационным, конструкционным и экономическим критериям. Вторым вариантом реализации узла SV с реализацией функций WDT является узел на микросхемах MAX824/825, т. к. имеются определенные организационные сложности с приобретением микросхем и разработчику необходимо иметь в наличии оба типа микросхем для реализации SV с и без WDT.

При выборе микросхемы SV показана современная методика проектирования различных узлов современной вычислительной техники.

Методика включает:

- определение предмета разработки;
- перечисление основных и возможных функций разрабатываемого узла;
- выработку основных критериев выбора элементной базы для разрабатываемого узла;
- перечисление основных производителей элементной базы;
- составление таблицы основной элементной базы;
- предварительный анализ и выбор элементной базы по основным критериям;
- составление таблицы функциональных выводов;
- сопоставление графических примитивов (изображений элементов);
- классификация и анализ отобранных элементов (разбиение на группы);
- рассмотрение схем включения элементов различных групп;
- сопоставление отобранных элементов по дополнительным критериям — организационным, конструкционным и экономическим;
- окончательное заключение (выбор основного и запасного вариантов).

Описанная методика позволит читателю грамотно решать другие аналогичные задачи.

3.6. Таймеры реального времени

3.6.1. Функциональные характеристики таймеров реального времени

Таймером реального времени (*RTC — Real Time Clock*) называется схема, предназначенная для независимого (от процессора или контроллера, а также от наличия питания) подсчета текущего времени и ведения функций календаря. Таймеры реального времени (далее просто таймеры или *RTC*) выпускаются рядом фирм: *Dallas Semiconductors* [70], *Thomson* [79], *Philips* [80], *Epson* [81], *Motorola* [82], *Unitrode (BenchMarq)* [83] и некоторые другие. Несомненным

лидером в этой области является фирма *Dallas Semiconductors*, которая выпускает наиболее разнообразную номенклатуру микросхем *RTC*.

Существует по крайней мере по два стандартных формата записи данных времени и календаря в *RTC*.

Первый, основной, формат записи данных времени обозначается как *STD* (*Standard Timer Data*) и содержит шесть байт (в некоторых 4-битных *RTC* — шесть младших полубайт), в которых записываются часы (*H*), минуты (*M*) и секунды (*S*) в виде *HH:MM:SS*.

Второй, расширенный, формат записи времени обозначается как *STD + hh*, где под байтами *hh* подразумевается значение десятков и единиц сотых долей секунды. Такой формат содержит 8 байт (полубайт) — *HH:MM:SS:hh*.

Первый, стандартный, формат записи данных календаря также обозначается как *STD* и содержит значения двух последних цифр года (*YY*), номер месяца (*MM*), номер дня (*DD*). Этот формат содержит шесть байт — *YY-MM-DD*.

Второй, расширенный, формат записи данных календаря обозначается *Y2K* и содержит в отличие от предыдущего случая значения всех четырех цифр года. Формат содержит 8 байт данных календаря — *YYYY-MM-DD*.

Перечислим основные и возможные функции (свойства) микросхем *RTC*.

1. Основная функция — генерация кодов времени и календаря, для чего микросхема имеет встроенный высокостабильный кварцевый генератор, обеспечивающий, как правило, точность не хуже ± 1 минута за месяц, набор счетчиков и программно доступные регистры, в которых помещаются данные текущего времени и календаря.

2. Многие микросхемы *RTC* имеют функцию генерации прерывания в запрограммированное время (*Time of Day Alarm*). Это прерывание генерируется раз в сутки. Обозначается эта функция буквой «А» (здесь и далее мы будем использовать обозначения, применяемые в документации фирмы *Dallas Semiconductors*).

3. Некоторые *RTC* имеют встроенный супервизор питания, который генерирует сигнал сброса (*Microprocessor Reset*) для микропроцессора (микроконтроллера) и удерживает его в состоянии сброса до тех пор, пока напряжение питания не достигло нормы. Функциональная возможность обозначается как *UR*. Это сделано для исключения возможности нарушения данных *RTC* во время переходных процессов питания и для блокировки возможности обращения процессора во время, когда питание не в норме.

4. В некоторых *RTC* имеется вход для внешнего сигнала (*Kickstart — KS*), который может вызвать генерацию выходного сигнала *RTC* для включения питания микропроцессорной системы.

5. Во многих микросхемах имеется выход (*Periodic — P*), который можно запрограммировать на генерацию импульсов в диапазоне периодов от 122 мкс до 500 мс.

6. Некоторые микросхемы имеют встроенный сторожевой таймер (*WDT*).

7. Многие микросхемы *RTC* имеют встроенную энергонезависимую оперативную память (*Random Access Memory — RAM*), причем размер этой памяти может изменяться от нескольких байт до 512 Кбайт. Это очень полезное свойст-

во микросхем *RTC* позволяет при достаточном объеме такой памяти использовать ее в качестве *Flash*-памяти для хранения различного рода настроек.

8. Еще одно полезное свойство некоторых микросхем *RTC* заключается в возможности очистки всей оперативной памяти (*RAM Clear — RC*), для чего в этих микросхемах имеется специальный вход.

9. Встречаются *RTC* с выходом, индицирующем занятость микросхемы внутренними операциями (*Update in Progress — U*).

10. Встречаются таймеры с дополнительным прерыванием в запрограммированное время (*Wake-up — WU*).

11. Для некоторых применений может быть полезна еще одна особенность — возможность записи времени между событиями (*Event Recorder — ER*).

12. Очень полезное свойство (*Power Cycle Counter — PC*) — счетчик, который считает количество циклов, прошедших с момента включения питания, т. е. с момента начала работы системы.

13. Встречается также свойство (*Vcc Active Counter — VC*) — счетчик, который считает только, когда питание активно.

14. Выпускаются микросхемы *RTC* с напряжением питания 5 В или 3,3 В.

15. Микросхемы могут быть выполнены в различных корпусах (*DIP, SO*), модулях (*M*) и в виде слотов (*S*).

16. Выпускаются таймеры *RTC* с внешним резервным питанием (батарежкой), с встроенной литиевой батарейкой (как правило, обеспечивающей до 10 лет непрерывной работы) и со встроенными гнездами для сменных батарей.

17. Доступ к данным микросхемы может быть параллельными (*Parallel*), смешанным (*Mixed*), побайтным (*Bytewise*), а также с использованием последовательных интерфейсов *2-Wire, 3-Wire* и *SPI*.

18. Многие микросхемы *RTC* с параллельным интерфейсом поддерживают несколько спецификаций шин, например *Intel* и *Motorola*.

19. Выпускаются таймеры с байтовым и полубайтным доступом.

20. Естественно, что выпускаются микросхемы с нормальным и расширенным температурным диапазоном работы.

21. Некоторые микросхемы *RTC*, особенно содержащие большой объем энергонезависимой памяти, совместимы по корпусу и функциональному назначению выводов с некоторыми микросхемами *ROM (Read Only Memory)*.

Кроме перечисленных основных и часто встречающихся дополнительных функций некоторые микросхемы имеют дополнительные специфические функции, например такие, как температурный контроль (*Temperature Control — TC*), температурный будильник (*Temperature Alarm — TA*), серийный номер (*Serial Number — SN*), встроенный аналого-цифровой преобразователь (*ADC*) и т. п.

3.6.2. Критерии выбора *RTC* для универсальных технологических контроллеров

Анализируя приведенные выше свойства микросхем *RTC*, становится ясно, что многие из них являются избыточными при решении нашей задачи (разработке универсальных технологических контроллеров). Определим критерии —

главные (обязательные) и желательные качества, по которым мы будем выбирать микросхемы *RTC*:

1. Безусловно, что микросхема *RTC* должна иметь функции счета времени и календаря. Формат данных времени и календаря принципиального значения не имеет.

2. Поскольку технологические контроллеры ориентированы на работу в реальном масштабе времени, обмен между собственно микроконтроллером и его периферийными узлами должен осуществляться достаточно быстро. В связи с этим микросхема *RTC* должна обеспечивать параллельный байтовый доступ к данным.

3. Желательно, чтобы микросхема *RTC* имела *Intel*-совместимую шину.

4. Обязательным является наличие встроенной энергонезависимой памяти. Объем памяти зависит от назначения разрабатываемых контроллеров. Для обычных технологических задач, как правило, хватает памяти объемом около 50 байтов. В достаточно редких случаях, когда контроллер должен запоминать и хранить при отключенном питании большой объем информации, необходимо выбирать *RTC* с *большим* объемом встроенной памяти. С учетом того, что адресное пространство большинства микроконтроллеров составляет 64 К и часть этого пространства занимает быстродействующая *RAM* и адреса периферийных устройств, объем встроенной (в *RTC*) оперативной памяти обычно не превышает 32 К.

5. Желательно, чтобы корпус и функциональное назначение выводов были совместимы с некоторыми микросхемами *Flash*-памяти.

6. Обязательно, чтобы микросхема *RTC* либо имела встроенную литиевую батарейку, либо имела гнезда для установки сменных батареек, т. е. чтобы микросхема была в модульном исполнении. Это значительно экономит площадь печатной платы.

7. Обязательно, чтобы микросхема *RTC* могла устанавливаться в стандартизованные *DIP*-панельки для обеспечения максимальной ремонтпригодности и модифицируемости контроллера.

Перечисленные критерии позволяют осуществить выбор микросхем *RTC* из всего многообразия выпускаемых микросхем. В таблице 3.5 приведены данные наиболее распространенных микросхем *RTC*.

Таблица 3.5

Тип микросхемы	Питание, В	Формат времени	Формат календаря	RAM, байт	Возможности	Тип шины	Корпус
DS1216B	5	STD+hh	STD	2К, 8К		P-Phantom	S
DS1216C	5	STD+hh	STD	8К, 32К		P-Phantom	S
DS1216D	3,3/5	STD+hh	STD	8К, 32К, 128К		P-Phantom	S
DS1216E	3/5	STD+hh	STD	8К, 32К		P-Phantom	S
DS1216F	3/5	STD+hh	STD	8К, 32К, 128К		P-Phantom	S

Продолжение табл. 3.5

Тип микросхемы	Питание, В	Формат времени	Формат календаря	RAM, байт	Возможности	Тип шины	Корпус
DS1216H	3,3/5	STD+hh	STD	128K, 512K		P-Phantom	S
DS1243Y	5	STD+hh	STD	8K		P-Phantom	M28
DS1244Y	3,3/5	STD+hh	STD	32K		P-Phantom	M28, P34
DS1248Y	3,3/5	STD+hh	STD	128K		P-Phantom	M32, P34
DS1251Y	3,3/5	STD+hh	STD	512K		P-Phantom	M32, P34
DS1254	3,3/5	STD+hh	STD	2M		P-Phantom	B168
DS1284, BQ4845	5	STD+hh	STD	50	A, WD	P-Bytewide	D28, L28
DS1286, BQ4847	5	STD+hh	STD	50	A, WD	P-Bytewide	M28
DS12885, RTC6593, MC146818A, BQ3285	5	STD	STD	114	A, P, U	P-Muxed	D24, L28, S24
DS12887, M48T86, BQ3287	5	STD	STD	114	A, P, U	P-Muxed	M24
DS12887A, BQ3287A	5	STD	STD	114	A, P, U, RC	P-Muxed	M24
DS12C887, BQ3287, M48T86	5	STD	Y2K	113	A, CR	P-Muxed	M24
DS12C887A, BQ3287A	5	STD	Y2K	113	A, P, U, RC	P-Muxed	M24
DS1302	2-5	STD	STD	31	TC	3-Wire	C8, D8, S8, S16
DS1305/1306, MC68HC68	2-5	STD	STD	96	2A, TC	3-Wire or SPI	C16, D16, T20
DS1307, МК41Т56, PCF8583/93, RTC8583/92	5	STD	STD	56		2-Wire	C8, D8, S8
DS1315	3,3/5	STD+hh	STD			P-Phantom	D16, S16, T20
DS1337	1,8-3,3	STD	Y2K		2A	2-Wire	C8, D8, S8
DS1384	5	STD+hh	STD	50	A, WD	P-Bytewide	Q44
DS1386, M48T59, BQ4832Y	5	STD+hh	STD	8K, 32K	A, WD	P-Bytewide	M32, P34
DS14285, BQ4285	5	STD	STD	114	A, P, U	P-Muxed	D24, L28, S24
DS14287, BQ4287	5	STD	STD	114	A, P, U	P-Muxed	M24
DS1486, BQ4842	5	STD+hh	STD	128K	A, WD	P-Bytewide	M32, P34
DS1500	3,3/5	STD	Y2K	256	A, KS, WU, WD, UR	P-Bytewide	D32, S32
DS1501, BQ4845	3,3/5	STD	Y2K	256	A, KS, WU, WD, UR	P-Bytewide	D28, S28, T28

Продолжение табл. 3.5

Тип микросхемы	Питание, В	Формат времени	Формат календаря	RAM, байт	Возможности	Тип шины	Корпус
DS1510	3,3/5	STD	Y2K	256	A, KS, WU, WD, UR	P-Bytewide	M32
DS1511, BQ4847	3,3/5	STD	Y2K	256	A, KS, WU, WD, UR	P-Bytewide	M28
DS1543	3,3/5	STD	STD	8K	A, WD, UR	P-Bytewide	M28
DS1553, M48T59A	3,3/5	STD	Y2K	8K	A, WD, UR	P-Bytewide	M28, P34
DS1554, BQ4832Y	3,3/5	STD	Y2K	32K	A, WD, UR	P-Bytewide	M32, P34
DS1556, BQ4842	3,3/5	STD	Y2K	128K	A, WD, UR	P-Bytewide	M32, P34
DS1557	3,3/5	STD	Y2K	512K	A, WD, UR	P-Bytewide	P34
DS1602	5	BIN	BIN			3-Wire	D8, S8
DS1603	5	BIN	BIN			3-Wire	SIP7
DS1615	5	STD	STD	32	A, TA, TS	3-Wire	C16, D16, S16
DS1616	5	STD	Y2K	32	A, TA, CB	3-Wire	D24, S24
DS1629	2,5-5	STD	STD		A, TA	3-Wire	S8
DS1642, M(K)48Tx2	5	STD	STD	2K	NV	P-Bytewide	M24
DS1643, M(K)48Tx8	5	STD	STD	8K	NV	P-Bytewide	M28, P34
DS1644, M48T35, BQ4830Y	5	STD	STD	32K	NV	P-Bytewide	M28, P34
DS1646	5	STD	STD	128K	NV	P-Bytewide	M32, P34
DS1647, BQ4850Y	5	STD	STD	512K	NV	P-Bytewide	M32, P34
DS1670	3,3	STD	STD		A, ADx3, WD, UR	3-Wire	S20, T20
DS1672	2/3/3,3	BIN	BIN		UR, TC	2-Wire,	
DS1673	3/5	STD	STD		A, ADx3, WD, UR	3-Wire	S20, T20
DS1677	5	STD	STD		A, ADx3, WD, UR	3-Wire	T20
DS1678	5	STD	Y2K	32	A, CR, ER	2-Wire	D8, S8
DS1680	3,3/5	STD	STD		A		Q44
DS1682	3-5	BIN		10-EPROM		2-Wire,	
DS1685, BQ3285L	3/5	STD	Y2K	114+128	A, P, U, KS, WU, RC	P-Muxed	C24, D24, L28, S24
DS1687, BQ3287E	3/5	STD	Y2K	114+128	A, P, U, KS, WU, RC	P-Muxed	M24

Продолжение табл. 3.5

Тип микросхемы	Питание, В	Формат времени	Формат календаря	RAM, байт	Возможности	Тип шины	Корпус
DS1688	3/5	STD	Y2K	114	A, P, U, KS, WU, RC, VC, PC	P-Mixed	D28, S28
DS1689	3/5	STD	Y2K	114	A, P, U, KS, WU, RC, VC, PC	P-Mixed	D28, S28
DS1691	3/5	STD	Y2K	114	A, P, U, KS, WU, RC, VC, PC	P-Mixed	M28
DS1693	3/5	STD	Y2K	114	A, P, U, KS, WU, RC, VC, PC	P-Mixed	M28
DS17285	3/5	STD	Y2K	114+2K	A, P, U, KS, WU, RC	P-Mixed	D24, S24, T28
DS17287	3/5	STD	Y2K	114+2K	A, P, U, KS, WU, RC	P-Mixed	M24
DS1742, M(K)48Tx2	3.3/5	STD	Y2K	2K	NV	P-Byte-wide	M24
DS1743, M48T59	3.3/5	STD	Y2K	8K	NV	P-Byte-wide	M28, P34
DS1744, M48T35, BQ4830Y	3.3/5	STD	Y2K	32K		P-Byte-wide	M28, P34
DS1746	3.3/5	STD	Y2K	128K	NV	P-Byte-wide	M32, P34
DS1747, BQ4850Y	3.3/5	STD	Y2K	512K	NV	P-Byte-wide	M32, P34
DS17485	3/5	STD	Y2K	114+4K	A, P, U, KS, WU, RC	P-Mixed	D24, S24, T28
DS17487	3/5	STD	Y2K	114+4K	A, P, U, KS, WU, RC	P-Mixed	M24
DS17885	3/5	STD	Y2K	114+8K	A, P, U, KS, WU, RC	P-Mixed	D24, S24, T28
DS17887	3/5	STD	Y2K	114+8K	A, P, U, KS, WU, RC	P-Mixed	M24
MAX6900	2-5	STD	Y2K			SPI	S6
DS2404	3.0-5	BIN			A, I, C	3-Wire	D16, S16
MAX6901	2-5	STD	Y2K			3-Wire	S8

В таблице использованы следующие обозначения корпусов (xx — количество выводов):

Mxx — модуль DIP, включающий собственно микросхему RTC, кварцевый резонатор и встроенную (сменную или постоянную) батарею;

Dxx — DIP-корпус;

Txx — TSSOP- и TSOP-корпуса;

Sxx — SOIC-корпуса;

Qxx — PQFP-корпуса;

Cxx — CSP-корпуса;

Lxx — PLCC-корпуса;

Bxx — BGA-модули, включающие собственно микросхему RTC, кварцевый резонатор и микросхему типа DS3800 для подзарядки батареи;

Pxx — PowerCap-модули, включающие собственно микросхему RTC, кварцевый резонатор и микросхему типа DS9034PCX для подзарядки батареи.

3.6.3. Анализ и выбор микросхем RTC

Из приведенных в таблице типов микросхем RTC можно отобрать всего несколько типов, удовлетворяющих приведенным выше критериям (см. табл. 3.6):

Таблица 3.6

Тип микросхемы	Формат времени	Формат календаря	RAM, байт	Возможности	Тип шины	Корпус
DS12(C)887(A), M48T86, BQ3287(A)	STD	STD	114	A, P, U, (RC)	P-Mixed	M24
DS14287, BQ4287	STD	STD	114	A, P, U	P-Mixed	M24
DS1687, BQ3287E	STD	Y2K	114+128	A, P, U, KS, WU, RC	P-Mixed	M24
DS1286, BQ4847	STD+hh	STD	50	A, WD	P-Byte-wide	M28
DS1691/DS1693	STD	Y2K	114	A, P, U, KS, WU, RC, VC, PC	P-Mixed	M28
DS1543/DS1643, M(K)48Tx8	STD	STD	8K	A, WD, UR	P-Byte-wide	M28
DS1553/DS1743, M48T59A	STD	Y2K	8K	A, WD, UR	P-Byte-wide	M28
DS1644/DS1744, M48T35, BQ4830Y	STD	STD	32K		P-Byte-wide	M28

Мы отобрали только те микросхемы, которые имеют параллельную шину и модульное исполнение с корпусами *MDIP24* и *MDIP28*. Первые три группы микросхем выполнены в модульных корпусах *MDIP24* и имеют очень несущественные отличия (отличия в дополнительных функциональных возможностях). Все они имеют объем встроенной памяти не менее 114 байт, что вполне достаточно для большинства стандартных применений. Остальные четыре группы микросхем выполнены в модульных корпусах *MDIP28*. Из них первые две группы имеют достаточно малый объем памяти (50 и 114 байт) для относительно большого корпуса. Это обстоятельство позволяет нам в случае плохого совпадения функционального назначения выводов исключить их из дальнейшего рассмотрения.

На рис. 3.7 показано функциональное назначение выводов микросхем RTC первой группы с мультиплексированными шинами адреса и данных.

На рисунке использованы следующие обозначения:

AD0...AD7 — мультиплексированная шина адреса/данных;

Vcc — напряжение питания;

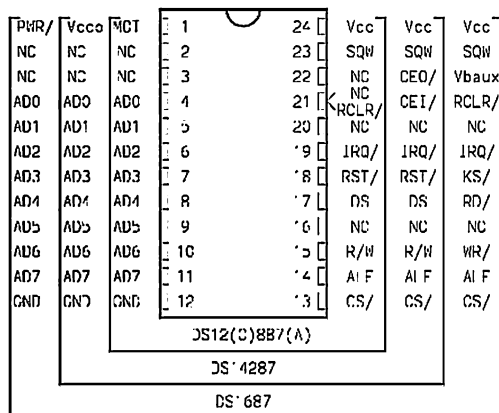


Рис. 3.7. Функциональное назначение выводов микросхем первой группы

GND — общий вывод питания;

MOT — вход выбора типа шины (Intel или Motorola);

V_{CC0} — выход для питания внешней памяти;

PWR/ — выход прерывания включения питания с открытым коллектором;

SQW — выход генератора программируемой частоты;

CEO/ — выход выборки кристалла для внешней памяти;

CEI/ — вход выборки кристалла для внешней памяти;

RCLR/ — вход стирания внутренней памяти;

IRQ/ — выход прерывания (будильника) с открытым коллектором;

RST/ — вход сброса;

DS — строб данных;

R/W — вход режима чтения/записи;

ALE — строб адреса;

CS/ — вход выборки кристалла;

V_{baux} — вход внешней батареи;

KS/ — вход сигнала Kickstart;

RD/ — вход сигнала чтения;

WR/ — вход сигнала записи;

NC — неиспользуемый вывод.

Анализ рис. 3.7 и приведенных данных о функциональном назначении выводов показывает, что микросхемы практически совпадают по основным функциональным выводам (за исключением сигналов управления чтением/записью R/W, RD/, WR/). Это позволяет при использовании минимальных внешних модификаторов устанавливать их на один и тот же сокет (панельку).

На рис. 3.8 показано функциональное назначение выводов микросхем RTC второй группы с разделенными шинами адреса и данных.

На рис. 3.8 использованы следующие обозначения:

A0...A14 — входы шины адреса;

D0...D7 — входы шины данных;

Vcc — вход питания;

GND — общий вывод питания;

CE/ — вход разрешения кристалла;

WE/ — вход разрешения записи;

OE/ — вход разрешения чтения;

RST/ — выход сброса с открытым коллектором;

IRQ/FT — выход прерывания и тестирования частоты с открытым коллектором;

SQW — выход генератора программируемой частоты;

CE2 — вход 2 разрешения кристалла;

NC — неиспользуемые выводы.

Анализ рис. 3.8 и приведенных данных о функциональном назначении выводов показывает, что микросхемы совпадают по основным функциональным выводам. Это позволяет устанавливать их на один и тот же сокет (панельку). Кроме того, разводка этой панельки позволяет при необходимости устанавливать на нее вместо микросхем *RTC* многие микросхемы постоянного запоминающего устройства (ROM).

Таким образом, мы отобрали для использования в универсальных технологических контроллерах две группы микросхем *RTC*. Первая группа — с минимальными (основными) функциями счета времени и календарем и малым объемом встроенной памяти может быть использована в более простых (в отношении решаемых программных задач) технологических контроллерах. Вторая группа — с основными функциями и встроенной энергонезависимой *RAM* довольно большого объема (8 К или 32 К) для использования в технологических контроллерах со сложным программным обеспечением и необходимостью энергонезависимого хранения больших объемов данных.

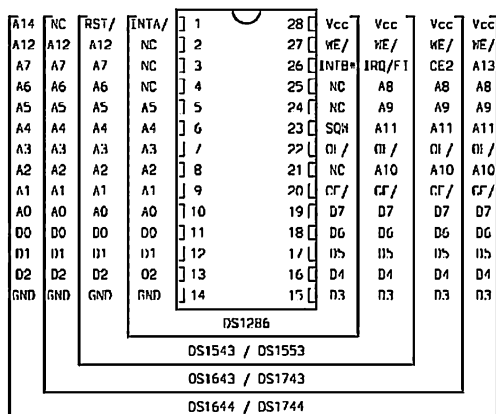


Рис. 3.8. Функциональное назначение выводов микросхем второй группы

3.6.4. Микросхемы *RAM* для технологических контроллеров

Как правило, при разработке универсальных технологических контроллеров известны их примерные области применения, соответственно известен примерный уровень сложности программного обеспечения и объемы обрабатываемых и накапливаемых данных. Если ожидается использование разрабатываемых контроллеров для обработки и хранения больших объемов данных, естественно, возникает необходимость в использовании в них микросхем оперативной памяти. К микросхемам памяти при этом предъявляются требования по соответствию быстродействию, достаточности объема, энергопотреблению и минимальности площади на печатной плате. Кроме того, учитывается то обстоятельство, что адресное пространство контроллеров ограничено и часть его отводится для работы периферийных устройств (в том числе микросхем *RTC* со встроенной энергонезависимой памятью). Учитывая эти обстоятельства, для большинства задач считается достаточным использование микросхем оперативной памяти с объемом 32 К (т. е. половины адресного пространства).

Микросхемы памяти выпускаются очень многими фирмами [84, 85]. Их параметры и схемы включения хорошо известны, поэтому приведение их данных в рамках этой книги нецелесообразно. Отметим только, что наиболее полно удовлетворяют требованиям по минимальности площади, занимаемой на печатной плате микросхемы в узких корпусах. Одной из таких микросхем *RAM* является микросхема *IS61C256A-15N* фирмы *ISSI* [86]. Она выпускается в узком корпусе *DIP28*, полностью статическая, питание микросхемы — 5 В, она имеет малое динамическое энергопотребление (400 мВт), малые времена цикла (10, 12, 15, 20 и 25 нс — обозначается после типа микросхемы), в режиме необращения потребление минимизируется и составляет при работе совместно с TTL входами 55 мВт и с CMOS входами — 250 мкВт.

3.6.5. Схемы включения микросхем *RTC*

Как итог вышесказанного рассмотрим две подсистемы памяти технологических микроконтроллеров. На рис. 3.9 показаны схемные изображения микросхем *RTC* первой группы. На рис. 3.10 показана подсистема памяти, позволяющая устанавливать все эти микросхемы и их аналоги (см. табл. 3.6) на одну панельку с реализацией практически всех основных функций.

На рис. 3.11 показаны схемные изображения микросхем *RTC* второй группы. На рис. 3.12 изображена подсистема памяти, включающая микросхему *RAM* и панельку для микросхем *RTC* с внутренней энергонезависимой *RAM*.

Изображенная на рис. 3.12 подсистема памяти содержит шинный формирователь данных *D1* (*74HC245*), регистр младшего байта адреса *D2* (*74HC573*), микросхему *RTC* второй группы *D3* и микросхему оперативной памяти *D4* (*61C256*). Функциональные назначения сигналов приведены над линиями сигналов, а перед линиями сигналов показаны названия битов портов микроконтроллера *AT89C51/52*. Модификаторы (перемычки) *JP1—JP10* предназначены для настройки панельки под конкретную микросхему *RTC* второй группы или мик-

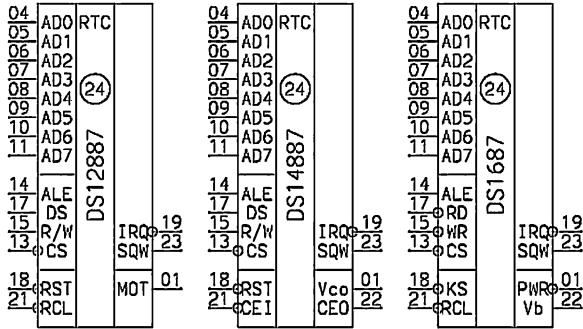


Рис. 3.9. Схемные изображения микросхем RTC первой группы

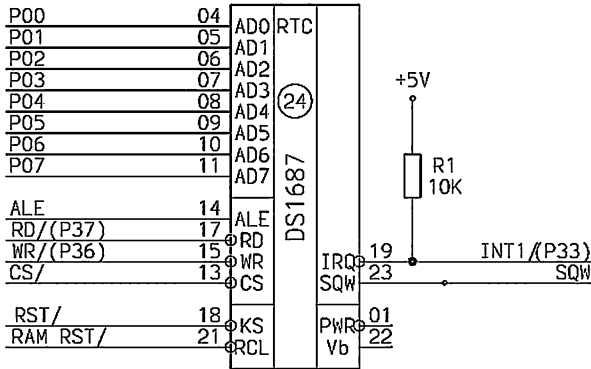


Рис. 3.10. Подсистема памяти на микросхемах RTC первой группы

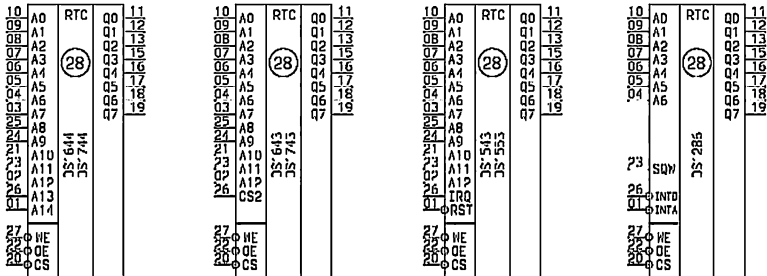


Рис. 3.11. Схемные изображения микросхем RTC второй группы

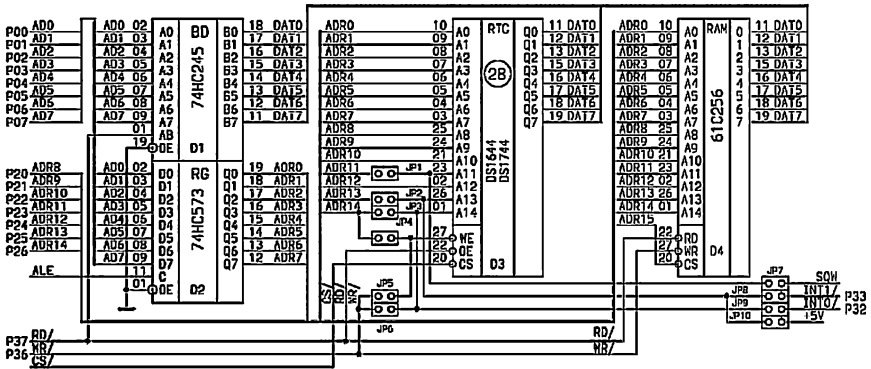


Рис. 3.12. Подсистема памяти на микросхемах RTC второй группы

роекемы постоянного запоминающего устройства ROM (Flash). Микросхема оперативной памяти 32 К D4 расположена в младшей половине адресного пространства микроконтроллера. Выборка кристалла микросхем D3 (RTC или ROM) должна поступать от системного дешифратора адреса (на схеме не приведен), который должен располагать либо ее, либо подсистему устройств ввода/вывода в старшей половине адресного пространства. К выводу 23 панельки через установленную перемычку JP1 может подаваться либо линия адреса ADR11 (DS1744, DS1644, DS1643, DS1743, DS1543, DS1553), либо при снятой перемычке JP1 и установленной JP7 с этого вывода может сниматься сигнал с выхода программируемого генератора меандра SQW (DS1286). Перемычки JP2—JP6 позволяют отключать соответствующие линии адреса или менять местами линии ADR14 и WR/ при установке на панельку микросхем ROM (Flash). Перемычки JP8, JP9 позволяют снимать сигналы прерываний при использовании микросхем DS1543, DS1553, DS1286. Перемычка JP10 необходима для подачи дополнительного сигнала выборки при использовании микросхем DS1643, DS1743. Таким образом, приведенная схема позволяет устанавливать на панельку целый ряд микросхем RTC с различными функциональными возможностями и объемом встроенной энергонезависимой памяти, что значительно повышает гибкость технологических контроллеров.

3.7. Устройства ввода/вывода и расширения

3.7.1. Последовательные интерфейсы технологических контроллеров

Современные технологические контроллеры, как правило, оснащаются одним или несколькими последовательными интерфейсами. Эти интерфейсы могут использоваться для связи контроллеров с другим оборудованием (измерительны-

ми приборами, исполнительными механизмами, персональными компьютерами). Однако наиболее часто современные технологические контроллеры используются в составе командно-информационных сетей *CI-LAN*, локальных сетей, состоящих из только одного персонального компьютера (*PC*) и нескольких технологических контроллеров, называемых в этом случае периферийными станциями (*PS*) или просто станциями.

Напомним, что основным принципом *CI-LAN* является то, что единственный в сети *PC* играет роль генератора команд для всех технологических контроллеров, а также получателя всей информации от них. При этом на *PC*, как на наиболее мощный и оснащенный периферией элемент сети, возлагаются задачи по вторичной обработке и сохранению данных, их интерпретации и визуализации.

Основными последовательными интерфейсами, используемыми в технологических контроллерах, являются *RS232C* и *RS485*. Узлы связи технологических контроллеров в *CI-LAN* называют сетевыми диспетчерами. Наиболее простые сетевые диспетчеры *CI-LAN* приведены и описаны во второй главе. В более сложных универсальных технологических контроллерах обычно используют комбинированные сетевые диспетчеры, в состав которых включают и интерфейс *RS232C*, и интерфейс *RS485*, а иногда и другие интерфейсы. Наличие в составе сетевого диспетчера нескольких интерфейсов значительно повышает гибкость систем за счет возможности подключения к *CI-LAN* дополнительного оборудования или создания многоуровневых *CI-LAN*.

На рис. 3.13 показана принципиальная схема одного из вариантов комбинированного сетевого диспетчера универсального технологического контроллера.

Принципиальная схема комбинированного сетевого диспетчера содержит следующие узлы: интерфейс *RS232C* (*D1*), интерфейс *RS485* (*D4*), настраиваемый узел оптической развязки (*D5—D7*), узел коммутации (*D2, D3*).

Диспетчер имеет независимые линии управления приема и передачи каждого интерфейса, что позволяет, например, принимать одновременно сигналы по одному или двум каналам (разумеется, с обнаружением конфликтов) и передавать данные в любой из интерфейсов или в оба сразу.

Настраиваемый узел оптической развязки (*D5—D7*) с помощью переключателей *JP3* можно модифицировать либо на работу в качестве оптически развязанных входов внешних прерываний, либо в качестве оптически изолированного интерфейса сети «*SISNET*» [39].

Штырьевые разъемы (типа *PLD*) *JP1—JP3* используются для подключения с помощью гибкого кабеля к разъемам *DB9*, которые обычно устанавливаются на корпусе станции, а также для обеспечения необходимого коммутации сигналов и земель. Переключатель *JP4* используется в сети *RS485* только тогда, когда станция подключена к концу шины (шлейфа). Переключатели *JP5—JP10* предназначены для настройки конфигурации диспетчера.

Следует также отметить, что кроме показанных на рисунке интерфейсов в последнее время практически во всех контроллерах станций устанавливаются дополнительные разъемы *PLD* для подключения к линиям интерфейса *SPI* (*Serial Peripheral Interface*), который в ранних контроллерах реализуется про-

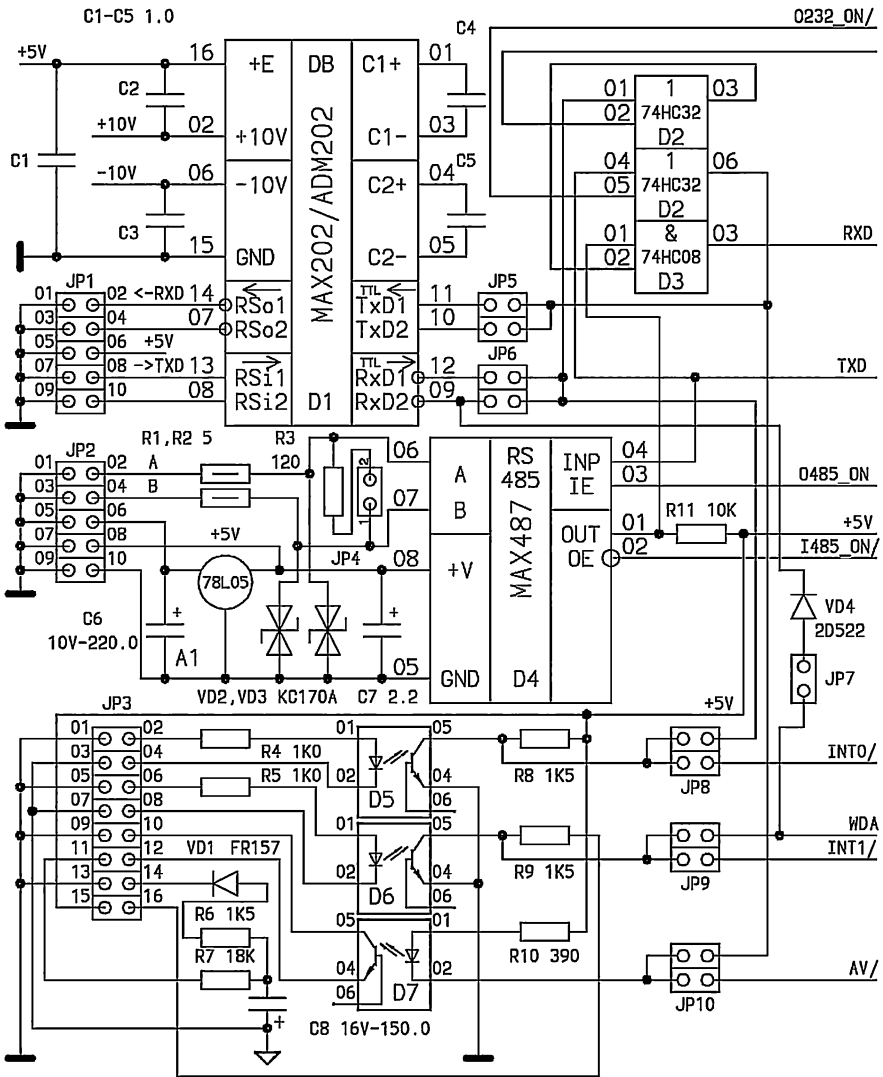


Рис. 3.13 Принципиальная схема комбинированного сетевого диспетчера

граммно, а в более старших реализован аппаратно. Для совместимости с аппаратно реализованным интерфейсом рекомендуется при программной реализации использовать линии *NSS/* (линия P14), *MOSI* (P15), *MISO* (P16), *SCK* (P17).

3.7.2. Регистры ввода-вывода технологических контроллеров

Узлы ввода-вывода технологических контроллеров чаще всего выполняются на основе регистров и шинных формирователей. В настоящее время производится достаточно большой ассортимент таких микросхем в различных сериях, с различным потреблением и быстродействием.

Отметим основные требования к регистрам и шинным формирователям:

1. Поскольку технологические контроллеры в основном оперируют байтовыми величинами, регистры и шинные формирователи также должны иметь байтовую организацию (разрядность 8).

2. Желательно, чтобы разводка выводов входов и выходов имела последовательное расположение, это значительно упрощает разводку печатной платы, позволяет снизить количество переходных отверстий, уменьшить площадь и соответственно стоимость платы, увеличивает надежность.

3. Желательно, чтобы выбранные регистры и шинные формирователи имели одинаковую разводку линий ввода-вывода и питания, это повышает гибкость технологических контроллеров за счет возможности, при необходимости, простой замены регистров на шинные формирователи и наоборот.

4. Желательно, чтобы выпускались микросхемы регистров и шинных формирователей с прямыми и инверсными выходами, что также повышает гибкость.

5. Очевидными требованиями являются малое энергопотребление и достаточно высокое быстродействие.

Всем перечисленным требованиям, кроме малого энергопотребления и высокого быстродействия, отвечают широко известные читателям регистры КР580ИР82/83 и шинные формирователи КР580ВА86/87. Однако потребление только одной такой микросхемы часто превышает потребление всех остальных микросхем технологических контроллеров и может достигать у отдельных экземпляров 100 мА. Существуют аналогичные микросхемы с низким потреблением (*i82C82/83*, *i82C86/87*), однако они получили малое распространение и практически не доступны.

Среди современных развивающихся серий микросхем наиболее широкий ассортимент имеет серия 74. Микросхемы этой серии в исполнении 74НС и 74НСТ имеют очень низкое среднее энергопотребление и высокое быстродействие.

Приведем таблицу с наиболее распространенными микросхемами восьмизрядных регистров и шинных формирователей этой серии (табл. 3.7).

Ознакомиться с техническими данными этих микросхем читатель может на сайтах фирм *Texas Instruments* [78], *Philips* [80], *Toshiba* [87], *IDT* [88], *Pericom* [89], *Hitachi* [90]. Полная номенклатура серии 74 с вариантами замены на отечественные микросхемы приведена в [91].

Анализ приведенных в таблице микросхем позволил выявить наиболее удачный комплект микросхем регистров и шинных формирователей (см. рис. 3.14).

К сожалению, не удалось подобрать комплект микросхем с полным совпадением номеров выводов входов и выходов (как это сделано в серии КР580ИР82/83 и КР580ВА86/87). Однако отобранные микросхемы наибольшим образом удовлетворяют всем остальным требованиям.

Таблица 3.7

Номер	Функциональное назначение	Отечественные аналоги
74HC240	Восьмиразрядный буфер – драйвер линии с инверсией и 3 состояниями	АП3
74HC241	Восьмиразрядный буфер – драйвер линии с 3 состояниями	АП4
74HC244	Восьмиразрядный буфер – драйвер линии с 3 состояниями	АП5
74HC245	Восьмиразрядный приемник шины	АП6
74HC273	Восьмиразрядный D-регистр со сбросом и срабатыванием по положительно-му фронту	ИР35
74HC373	Восьмиразрядный D-регистр-защелка с третьим состоянием	ИР22
74HC374	Восьмиразрядный D-регистр со срабатыванием по положительному фронту	ИР23
74HC377	Восьмиразрядный D-регистр с разрешением данных	ИР27
74HC465	Восьмиразрядный буфер с 3 состояниями	АП14
74HC534	Восьмиразрядный D-регистр со срабатыванием по положительному фронту, с инверсией и 3 состояниями	ИР41
74HC540	Восьмиразрядный буфер – драйвер линии с инверсией и 3 состояниями	АП12
74HC541	Восьмиразрядный буфер – драйвер линии с 3 состояниями	АП13
74HC543	Восьмиразрядный регистровый передатчик	
74HC563	Восьмиразрядный D-регистр-защелка с третьим состоянием и инверсией	
74HC573	Восьмиразрядный D-регистр-защелка с третьим состоянием	ИР33
74HC574	Восьмиразрядный D-регистр с третьим состоянием и срабатыванием по положительному фронту	ИР37
74HC580	Восьмиразрядный D-регистр-защелка с инверсией	
74HC640	Восьмиразрядный шинный передатчик с инверсией и 3 состояниями	АП9
74HC641	Восьмиразрядный шинный передатчик	АП7
74HC645	Восьмиразрядный шинный передатчик	АП8
74HC646	Восьмиразрядный шинный передатчик / регистр с 3 состояниями	ВА1
74HC652	Восьмиразрядный шинный передатчик / регистр с 3 состояниями	АП24

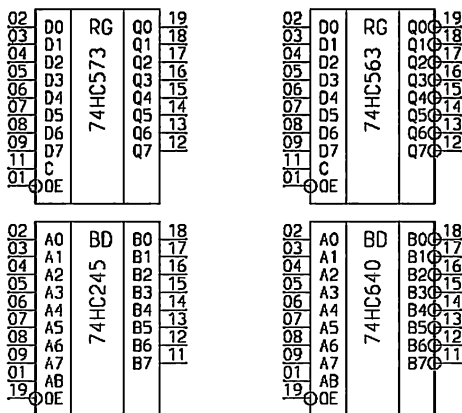


Рис. 3.14. Комплект микросхем регистров и шинных формирователей, оптимальный для применения в технологических контроллерах

3.7.3. Модификаторы адреса

В универсальных технологических контроллерах, оснащенных сетевыми диспетчерами, обязательно должен иметься модификатор адреса — узел, позволяющий установить уникальный индивидуальный адрес контроллера (станции) в сети. Существует достаточно много различных вариантов исполнения модификаторов.

1. В некоторых случаях, с целью экономии места на печатной плате, уникальный адрес прошивается в *Flash*-памяти микроконтроллера. Недостаток такой реализации модификатора очевиден — необходимо для каждой станции иметь уникальную прошивку.

2. Модификатор может быть выполнен в виде программно-доступного регистра, входы которого подтянуты резисторами к питанию, и к этим же входам подключены перемычки (см. рис. 3.15).

3. Вместо перемычек может быть использован переключатель типа ВДМ или аналогичный.

4. В последнее время в особо ответственные изделия стали встраивать новый тип модификаторов, получивший название *iButton*. Это гибридный полупроводниковый прибор, выполненный в корпусе из нержавеющей стали, внешне очень похожем на литиевую батарейку. Приборы выпускаются в двух типах корпусов (F5 и F3), диаметр обоих корпусов около 17 мм, а толщина соответственно около 6 и 3 мм. Такой корпус обеспечивает высокую механическую прочность и живучесть изделия. Чашка корпуса является общим выводом, а электрически изолированная крышка — сигнальным электродом. Прибор содержит внутри литиевую батарейку и целый набор функциональных узлов. Батарейка обеспечивает непрерывное функционирование прибора в течение не менее 10 лет.

Главным функциональным узлом прибора, безусловно, является постоянное запоминающее устройство *ROM*, содержащее 64-битный код. В состав этого кода входят: байт кода типа прибора, 6 байт уникального серийного номера и

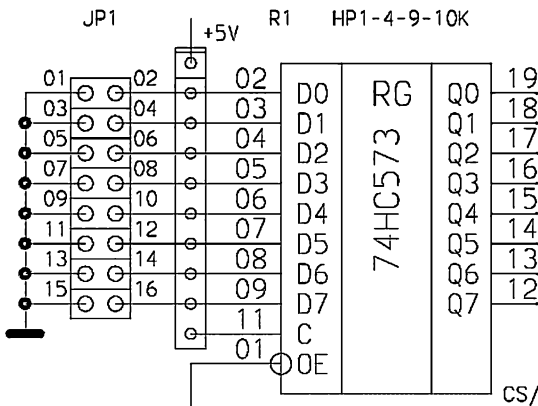


Рис. 3.15. Вариант реализации модификатора адреса

байт контрольной суммы. Уникальный серийный номер записывается с помощью лазерной технологии при изготовлении прибора. Питание *ROM* осуществляется по сигнальной линии, что экономит емкость батарейки и обеспечивает независимость считывания. В некоторых приборах имеется оперативное запоминающее устройство *RAM*, которое питается от встроенной батарейки и может хранить информацию не менее 10 лет. *RAM* имеет страничную организацию по 32 байта и может содержать от 4 до 16 страниц у различных моделей. Прибор имеет буфер данных для верификации в нем записанной информации перед записью в основное *RAM*. В некоторых, еще более сложных приборах имеется паролирование данных записи в *RAM*. Кроме того, некоторые из этих приборов имеют встроенные таймеры реального времени с отображением данных времени/календаря в дополнительную страницу памяти. Более подробно ознакомиться с характеристиками и номенклатурой этих приборов можно в [92].

Следует только заметить, что использование этого прибора влечет за собой необходимость использования специального контактного устройства и микросхемы считывания, что значительно увеличивает стоимость изделия, снижает быстродействие и увеличивает площадь печатной платы. Использование *iButton* в любой аппаратуре должно быть очень серьезно обосновано и аргументировано.

По мнению автора, реализация модификатора адреса станции с переключками, устанавливаемыми на двухрядный разъем *PLD*, является наиболее удачной, поскольку незадействованные или все входы регистра при необходимости могут быть использованы как входные линии. Кроме того, она дешевле варианта с переключателем и занимает меньше места на печатной плате.

3.7.4. Оптически развязанные узлы

Очень многие задачи, которые приходится решать универсальным технологическим контроллерам, требуют гальванической развязки входов или выходов контроллера от всевозможных исполнительных устройств, каналов ввода и датчиков. Наиболее часто для этих целей используются оптические элементы развязки — транзисторные оптопары (*Optocouplers*). Как правило, современные транзисторные оптопары изготавливаются в корпусах *DIP6* и *DIP8*. Очевидно, что корпус *DIP6* более удобен для применения в технологических контроллерах, поскольку позволяет создавать более компактные (по площади печатной платы) изделия.

Отметим также, что при создании технологических контроллеров, при выборе оптопар в первую очередь обращается внимание на следующие параметры:

- максимальное быстродействие (время включения T_{on} , время выключения T_{off}) оптронов;
- максимальное допустимое напряжение коллектор-эмиттер закрытого транзистора U_{ce} ;
- максимальное допустимое напряжение база-эмиттер закрытого транзистора U_{be} ;
- минимальное напряжение насыщения открытого транзистора $U_{ce,sat}$;
- максимальное напряжение развязки U_{oi} ;

- максимальное отношение тока коллектора открытого транзистора I_c к току светодиода I_f ;
- доступность оптопар.

Справочные данные наиболее подходящих (по приведенным выше критериям) импортных оптронов в корпусе *DIP6* приведены в табл. 3.8. С полным перечнем выпускаемых импортных оптронов в корпусах *DIP6* и *DIP8* и их характеристиками можно ознакомиться, например, в [91]. Довольно полные справочные данные на некоторые отечественные оптроны приведены в [93].

В таблицу включались только оптроны, время включения/выключения которых менее 10 мкс (за исключением широко распространенных оптронов серии 4N).

Разводка выводов оптопар в корпусе *DIP6*: 1 — положительный вывод светодиода (анод); 2 — отрицательный вывод светодиода (катод); 3 — не используется; 4 — эмиттер транзистора; 5 — коллектор транзистора; 6 — база транзистора (используется в некоторых типах оптронов).

Таблица 3.8

Тип оптрона	I_c/I_f , при $I_f > 10$ мА, %		$U_{ссe}$, max	$U_{вe}$, В	$U_{сc,сaт}$, В	T_{on}/T_{off} , Max, мкс	U_{oi} , кВ	Вывод базы
	min	max						
4N25	20	—	30	70	0,5	1,2/1,3	5,3	+
4N26	20	—	30	70	0,5	1,2/1,3	5,3	+
4N27	10	—	30	70	0,5	1,2/1,3	5,3	+
4N28	10	—	30	70	0,5	1,2/1,3	5,3	+
4N35	100	—	30	70	0,3	10/10	5,3	+
4N36	100	—	30	70	0,3	10/10	5,3	+
4N37	100	—	30	70	0,3	10/10	5,3	+
CNX83A	40	—	50	70	0,4	3/3	5,3	+
IL74	12,5	—	20	70	0,3	3/3	5,3	
MCT2	20	—	30	70	0,4	1,2/1,3	5,3	
MCT271	45	90	30	70	0,4	7/7	5,3	
MCT276	15	60	30	70	0,4	3,5/3,5	5,3	
MCT2E	20	—	30	70	0,4	1,2/1,3	5,3	
PC111L	50	—	35	70	0,2	4/3	5,0	
PC112L	40	—	70	70	0,2	4/3	5,0	
PS2601	80	600	50	—	—	3,5	5,0	
PS2651	50	400	50	—	—	3,5	5,0	
SFH600-0	40	80	70	—	0,4	3/3	5,3	
SFH600-1	63	125	70	—	0,4	3/3	5,3	
SFH600-2	100	200	70	—	0,4	3/3	5,3	
SFH600-3	160	320	70	—	0,4	3/3	5,3	

Продолжение табл. 3.8

Тип оптрона	I_F/I_F при $I_F > 10$ мА, %		$U_{ce, max}$	U_{be} , В	$U_{ce, sat}$, В	T_{on}/T_{off} , Макс, мкс	U_{off} , кВ	Вывод базы
	min	max						
SFH601-0	40	80	100	—	0,4	3/2,3	5,3	
SFH601-1	63	125	100	—	0,4	3/2,3	5,3	
SFH601-2	100	200	100	—	0,4	3/2,3	5,3	
SFH601-3	160	320	100	—	0,4	3/2,3	5,3	
SFH608-2	63	125	55	—	0,4	8/7,5	5,3	
SFH608-3	100	200	55	—	0,4	8/7,5	5,3	
SFH608-4	160	320	55	—	0,4	8/7,5	5,3	
SFH608-5	250	500	55	—	0,4	8/7,5	5,3	
SFH640-1	40	80	300	300	0,4	5/6	7,5	
SFH640-2	63	125	300	300	0,4	5/6	7,5	
SFH640-3	100	200	300	300	0,4	5/6	7,5	
TIL111	8	—	30	70	0,4	5/5	1,5	
TIL112	2	—	20	70	0,5	2/2	1,5	
TIL117	50	—	30	70	0,4	5/5	2,5	
TIL126	50	—	30	70	0,4	2/2	1,5	

Как видно из таблицы, широко распространенные оптроны 4N35 не являются лучшими в плане быстродействия.

При включении оптронов в качестве входных обычно последовательно с плюсовым выводом светодиода включается резистор, величина которого выбирается из расчета тока 10 мА при рабочем напряжении (за исключением специальных случаев, например когда ток источника сигнала лимитирован). Иногда встречно-параллельно светодиоду включается защитный диод (обычно КД522). При этом, по возможности, не следует объединять все входные минусовые линии вместе, т. к. это снижает возможность более гибкого использования оптронов. Выходные схемы включения оптронов могут выполняться с общим эмиттером и коллектором, в зависимости от требований к выходным сигналам.

При включении оптронов в качестве выходных обычно выводят свободные коллекторы и эмиттеры, которые также можно защищать встречно-параллельным включением диодов и последовательным резистором, включенным в коллектор.

Обычно в технологических контроллерах с оптической изоляцией выполняют сигналы от внешних источников прерываний, в некоторых случаях оптически развязывают последовательные интерфейсы, а также делают 1—2 оптически развязанных выхода для подключения различной сигнализации или других исполнительных устройств. С целью экономии площади печатной платы и снижения стоимости изделия входные и выходные байтовые шины на самом контроллере выполняют достаточно редко, т. к. значительно удобнее изготовить ма-

ленькую дополнительную плату развязки и при необходимости соединять ее с универсальным контроллером плоским кабелем.

3.7.5. Интерфейс LCD-индикаторов

В последнее время практически в состав любого серьезного универсального технологического контроллера включается интерфейс с жидкокристаллическими интеллектуальными символьными индикаторами (*LCD*). Интерфейс в этом случае — понятие достаточно условное, т. к. никаких дополнительных микросхем для подключения *LCD* не требуется. Достаточно на плате установить *PLD*-разъем 2×17 . Схема подключения *LCD*-индикатора показана на рис. 3.16.

Схема настолько простая, что требуется всего несколько дополнительных замечаний:

- рекомендуется использовать указанные на схеме линии портов ввода-вывода для обеспечения совместимости с другими узлами универсальных технологических контроллеров;
- рекомендуется устанавливать ограничивающий резистор *R1* для увеличения срока службы светодиодов подсветки;
- резистор *R2* необходим для настройки контрастности при различных исполнениях *LCD*. При этом в качестве отрицательного источника питания можно использовать либо соответствующий выход микросхемы *MAX232* (или аналогов) или специальную микросхему для получения отрицательного напряжения, либо усложнять источник питания (что нецелесообразно).

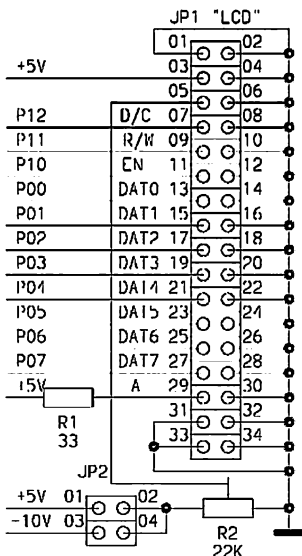


Рис. 3.16. Схема подключения *LCD*-индикатора

3.7.6. Диспетчеры памяти технологических контроллеров

Лишь в некоторых, очень сложных универсальных технологических контроллерах используются диспетчеры памяти — узлы управления внешней памятью с расширением адресного пространства. Современные микроконтроллеры имеют 64 К адресного пространства (обычно называют «кубом памяти»). Как правило, этого хватает для большинства применений, т. к. это адресное пространство используется только для организации дополнительной *RAM* и подсистемы ввода/вывода. Однако в некоторых случаях условия эксплуатации системы требуют наличия в технологических контроллерах встроенных узлов формирования телевизионных изображений. Другие типы индикаторов, в том числе и графические *LCD*, не подходят либо по быстродействию, либо по другим условиям эксплуатации. В этих случаях в состав технологического контроллера включается дополнительная память изображений на один или несколько (чаще 2) кадров. Формат изображения при этом обычно составляет 256×256 точек. На кодирование цвета точки отводится 1 байт. Нетрудно подсчитать, что на один такой кадр необходимо как раз 64 К памяти, а на 3 (2 кадра и подсистема ввода/вывода) необходимо соответственно 192 К. Вот в этих-то случаях без диспетчера памяти не обойтись. Идея диспетчера памяти достаточно проста [94]: включить в состав контроллера дополнительный программно доступный регистр, выходной байт которого сделать старшим байтом адреса (*ADDR16—ADDR23*). В этот регистр можно заранее записать адрес необходимого куба (например, одного из кубов кадров изображения), а при обращении к внешней памяти (при выработке сигналов *RD/* и *WR/*) открывать выход регистра на время обмена с памятью. Таким образом, с помощью одного регистра адресное пространство микроконтроллера можно увеличить с 64 К до 16 М.

3.8. Аналого-цифровые преобразователи

3.8.1. Критерии выбора аналого-цифровых преобразователей для *TMCS*

Аналого-цифровым преобразователем (ADC) называется схема, предназначенная для преобразования входной аналоговой величины в выходной цифровой код. Микросхемы аналого-цифровых преобразователей (далее просто *ADC*) выпускаются рядом фирм: *Analog Devices* [69], *Burr-Brown* [95], *Exar* [96], *Fairchild Semiconductor* [97], *Hitachi* [90], *Intersil* [98], *Linear Technology* [73], *Maxim* [74], *Microchip* [75], *National Semiconductor* [99], *Sipex* [76], *Technologies (Toko)* [100], *Texas Instruments* [78] и др. Данные по выпускаемым этими фирмами *ADC* можно найти на [91]. Несомненным лидером в этой области является фирма *Maxim*, которая выпускает наиболее разнообразную номенклатуру микросхем *ADC*.

Наиболее важными параметрами *ADC* являются разрядность, тип интерфейса, быстродействие, тип корпуса, характеристики питания.

Разрядность ADC определяет точность преобразования. В настоящее время выпускаются микросхемы с разрядностью от 8 до 22. Ориентировочно точность можно оценить, зная значимость младшего разряда, как величину, обратную 2^n , где n — количество разрядов (см. табл. 3.9).

Таблица 3.9

Количество разрядов	8	10	12	14	16	18
Значимость младшего разряда, %	0,39	0,098	0,024	0,006	0,0015	0,00038

Следует отметить, что реальная погрешность будет несколько выше за счет нелинейности характеристики ADC , а также возможных «пропусков кодов». Однако и приблизительной оценки погрешности вполне достаточно для того, чтобы сделать вывод о достаточной разрядности для большинства технологических применений. В настоящее время в технологических системах вполне достаточной считается разрядность 12—14 двоичных разрядов.

Выпускаемые в настоящее время ADC могут иметь параллельный или последовательный интерфейс. Параллельный интерфейс чаще всего используется в модульных микропроцессорных системах, когда подсистема аналого-цифрового преобразователя проектируется в виде отдельного модуля. Если подсистема ADC располагается на одной плате с микроконтроллером и нет особых требований по быстродействию системы, предпочтительным является последовательный интерфейс. В настоящее время существует достаточно большой набор разных последовательных интерфейсов, среди которых наиболее часто используется интерфейс SPI , так как большинство современных микроконтроллеров оснащаются аппаратной реализацией этого интерфейса.

Важное значение имеет быстродействие аналого-цифрового преобразователя. Современные ADC по быстродействию можно условно разделить на 3 группы: низкого быстродействия — до 100 ksps, среднего быстродействия — от 100 до 500 ksps, высокого быстродействия — более 500 ksps. Для универсальных технологических контроллеров наиболее часто используются ADC низкого или среднего быстродействия.

Некоторые современные микросхемы ADC имеют встроенный мультиплексор, однако количество каналов в таких микросхемах обычно невелико и управление ими достаточно громоздко, да и стоимость значительно выше, поэтому в технологических контроллерах разработчики чаще всего делают внешний (по отношению к ADC) входной мультиплексор.

Микросхемы ADC могут иметь одно или несколько различных напряжений питания. Очевидно, что для технологических микросистем желательно, чтобы микросхема имела одно питание, равное 5 В.

Важное значение при выборе микросхемы имеет тип корпуса. Как уже отмечалось в первом разделе этой главы, оптимальным для технологических контроллеров является корпус типа DIP , который позволяет устанавливать микросхему на панельку, что значительно упрощает ремонт и наладку контроллера. Кроме того, желательно, чтобы корпус имел малое количество выводов. Обычно используются микросхемы с количеством выводов от 8 до 16.

Еще одним критерием отбора микросхем *ADC* является наличие в них встроенного источника опорного напряжения, т. к. его отсутствие в некоторых микросхемах приводит к необходимости установки на плате дополнительных микросхем и пассивных элементов.

Как уже отмечалось выше, наиболее крупным и известным производителем микросхем *ADC* является фирма *Maxim* [74]. В табл. 3.10 приведен перечень интегральных аналого-цифровых преобразователей, выпускаемых этой фирмой.

Таблица 3.10

Тип микросхемы	Разрядность, бит	Количество каналов	Максимальное время преобразования, мкс	Напряжения питания, В	Ток потребления, мА	Интерфейс	Опорное напряжение, В	Диапазон входных напряжений, В	Наличие дифф. входа	Корпус
ADC0820	8	1	1,4	+5	5	μP8	Ext.	+5	—	CDIP20, DIP20, WSO20
MAX1101	8	3	0,66	4,75...5,25	20	Посл.	Ext.	±Vref	—	WSO24
MAX1106	8	1	35	2,7...5,5	0,096	Посл.	Ext., InL. +2,048	±Vref/2	√	μMAX10
MAX1107	8	1	35	4,5...5,5	0,115	Посл.	Ext., InL. +4,096	±Vref/2	√	μMAX10
MAX1108	8	2	20	2,7...5,5	0,07	Посл.	Ext., InL. +2,048	+Vref	—	μMAX10
MAX1109	8	2	20	4,5...5,5	0,095	Посл.	Ext., InL. +4,096	+Vref	—	μMAX10
MAX1110	8	8	20	2,7...5,5	0,085	Посл.	Ext., InL. +2,048	+Vref, ±Vref/2	√	DIP20, SSOP20
MAX1111	8	4	20	2,7...5,5	0,085	Посл.	Ext., InL. +2,048	+Vref, ±Vref/2	√	DIP16, QSOP16
MAX1112	8	8	20	4,5...5,5	0,135	Посл.	Ext., InL. +4,096	+Vref, ±Vref/2	√	DIP20, SSOP20
MAX1113	8	4	20	4,5...5,5	0,135	Посл.	Ext., InL. +4,096	+Vref, ±Vref/2	√	DIP16, QSOP16
MAX113	8	4	1,8	3,0...3,6	1,5	μP8	Ext.	+3, ±1,5	—	CDIP24, NDIP24, SSOP24
MAX117	8	8	1,80	3,0...3,6	1,5	μP8	Ext.	+3, ±1,5	—	SSOP28, DIP28, SSO28
MAX120	12	1	1,6	5 и -12...-15	9	μP12	InL. -5,0	±5	—	SSOP24, CDIP24, NDIP24, WSO24
MAX1202	12	8	6	+5, ±5	1,5	Посл.	Ext., InL. +4,096	+4,096, ±2,048	√	DIP20, SSOP20
MAX1203	12	8	6	+5, ±5	1,5	Посл.	Ext.	+Vref, ±Vref/2	√	DIP20, SSOP20
MAX1204	10	8	6	+5, ±5	1,5	Посл.	Ext., InL. +4,096	+Vref, ±Vref/2	√	DIP20, SSOP20
MAX121	14	1	2,9	+5 и -12...-15	9	Посл.	InL. -5,0	±5	—	SSOP20, CDIP16, DIP16, WSO16
MAX122	12	1	2,6	+5 и -12...-15	9	μP12	InL. -5,0	±5	—	SSOP24, CDIP24, NDIP24, WSO24

Продолжение табл. 3.10

Тип микро- схемы	Раз- ряд- ность, бит	Колл- чество каша- лов	Максималь- ное время преобразо- вания, мкс	Напряжения питания, В	Ток по- требле- ния, мА	Интерфейс	Опорное напряже- ние, В	Диапазон входных на- пряжений, В	Наличие дифф. входа	Корпус
MAX1240	12	1	7,5	2,7...3,6	1,4	Посл.	Ext., InL. +2,5	+Vref	—	DIP8,S08
MAX1241	12	1	7,5	2,7... 5,25	0,9	Посл.	Ext.	+Vref	—	DIP8,S08
MAX1242	10	1	7,5	2,7...3,6	1,4	Посл.	Ext., InL. +2,5	+Vref	—	DIP8,S08
MAX1243	10	1	7,5	2,7... 5,25	0,9	Посл.	Ext.	+Vref	—	DIP8,S08
MAX1245	12	8	7,5	2,375... 3,3	0,8	Посл.	Ext.	+Vref, ±Vref/2	√	DIP20,SSOP20
MAX1246	12	4	6	2,7...3,6	1,2	Посл.	Ext., InL. +2,5	+Vref, ±Vref/2	√	DIP16,QSOP16
MAX1247	12	4	6	2,7... 5,25	0,8	Посл.	Ext.	+Vref, ±Vref/2	√	DIP16,QSOP16
MAX1248	10	4	6	2,7...3,6	1,2	Посл.	Ext., InL. +2,5	+Vref, ±Vref/2	√	DIP16,QSOP16
MAX1249	10	4	6	2,7...5,25	0,8	Посл.	Ext.	+Vref, ±Vref/2	√	DIP16,QSOP16
MAX125	14	8	3	±5	17	μP/14	Ext., InL. +2,5	±5	—	SSOP36
MAX126	14	8	3	±5	17	μP/14	Ext., InL. +2,5	±2,5	—	SSOP36
MAX127	12	8	10	+5	6	Посл. 2- wire	Ext., InL. +4,096	+10, +5, ±10, ±5	—	NDIP24,SSOP28
MAX1270	12	8	3	+5	6	Посл.	Ext., InL. +4,096	+10, +5, ±10, ±5	—	NDIP24,SSOP28
MAX1271	12	8	3	+5	6	Посл.	Ext., InL. +4,096	+Vref,+Vref/2 ±Vref,±Vref/2	—	NDIP24,SSOP28
MAX128	12	8	10	+5	6	Посл. 2- wire	Ext., InL. +4,096	+Vref,+Vref/2 ±Vref,±Vref/2	—	NDIP24,SSOP28
MAX1290	12	8	1,9	4,5...5,5	1,9	μP8	Ext., InL. +2,5	+Vref, ±Vref/2	√	QSOP28
MAX1291	12	8	1,7	2,7...3,6	1,7	μP8	Ext., InL. +2,5	+Vref, ±Vref/2	√	QSOP28
MAX1292	12	4	1,9	4,5...5,5	1,9	μP8	Ext., InL. +2,5	+Vref, ±Vref/2	√	QSOP24
MAX1293	12	4	1,7	2,7...3,6	1,7	μP8	Ext., InL. +2,5	+Vref, ±Vref/2	√	QSOP24
MAX1294	12	6	2,0	4,5...5,5	2	μP12	Ext., InL. +2,5	+Vref, ±Vref/2	√	QSOP28
MAX1295	12	6	1,8	2,7...3,6	1,8	μP12	Ext., InL. +2,5	+Vref, ±Vref/2	√	QSOP28
MAX1296	12	2	2,0	4,5...5,5	2	μP12	Ext., InL. +2,5	+Vref, ±Vref/2	√	QSOP24

Продолжение табл. 3.10

Тип микро- схемы	Раз- ряд- ность, бит	Коли- чество кана- лов	Максималь- ное время преобразо- вания, мкс	Напряжения питания, В	Ток по- требле- ния, мА	Интерфейс	Опорное напряже- ние, В	Диапазон входных на- пряжений, В	Наличие дифф. входа	Корпус
MAX1297	12	2	1,8	2,7...3,6	1,8	μP12	Ext., InL. +2,5	+Vref, ±Vref/2	√	QSOP24
MAX144	12	2	7	2,7...5,25	0,9	Посл.	Ext.	+Vref	—	CDIP8,8/μMAX, DIP8
MAX145	12	1	7	2,7...5,25	0,9	Посл.	Ext.	+Vref	√	CDIP8,8/μMAX, DIP8
MAX146	12	8	6	2,7...3,6	1,2	Посл.	Ext., InL. +2,5	+Vref, ±Vref/2	√	CDIP20,DIP20, SSOP20
MAX147	12	8	6	2,7...5,25	0,9	Посл.	Ext.	+Vref, ±Vref/2	√	CDIP20,DIP20, SSOP20
MAX148	10	8	7,5	2,7...5,25	0,8	Посл.	Ext.	+Vref, ±Vref/2	√	CDIP20,DIP20, SSOP20
MAX149	10	8	7,5	2,7...3,6	1,2	Посл.	InL. +2,5	+2,5, ±1,25	√	CDIP20,DIP20, SSOP20
MAX150	8	1	1,34	4,75...5,25	5	μP8	Ext., InL. +2,5	+5	—	CDIP20,DIP20, WSO20
MAX151	10	1	2,5	±5	30	μP/10	Ext., InL. +4	+5	—	CDIP24,NDIP24, WSO24
MAX152	8	1	1,8	+3,±3	1,5	μP8	Ext.	+3,±1,5	—	CDIP20,SSOP20, DIP20,WSO20
MAX154	8	4	2,5	+5	5	μP8	Ext., InL. +2,5	+5,±2,5	—	CDIP24,SSOP24, NDIP24,WSO24
MAX155	8	8	3,6	+5,±5	18	μP8	Ext., InL. +2,5	+2,5,±2,5	√	CDIP28,DIP28, WSO28
MAX156	8	4	3,6	+5,±5	9	μP8	Ext., InL. +2,5	+2,5,±2,5	√	NDIP24,WSO28
MAX157	10	2	7	2,7...5,25	0,9	Посл.	Ext.	+Vref	√	CDIP8,8/μMAX, DIP8
MAX158	8	8	2,5	+5	5	μP8	Ext., InL. +2,5	+5,±2,5	—	SSOP28,CDIP28, DIP28,WSO28
MAX159	10	1	7	2,7...5,25	0,9	Посл.	Ext.	±Vref/2	—	CDIP8,8/μMAX, DIP8
MAX160	8	1	4	4,75... 5,25	1	μP8	Ext.	+10,±5	—	DIP18,SO18
MAX161	8	8	20	4,5...5,5	3	μP8	Ext.	+10	—	DIP28, SO28
MAX162	12	1	3,25	+5 и -12... -15	5	μP12, μP8	InL. - 5,25	+5	—	NDIP24,SO24
MAX163	12	1	8,33	+5 и -12... -15	4	μP12, μP8	InL. -5,0	+5	—	NDIP24,WSO24
MAX164	12	1	8,33	+5 и -12... -15	4	μP12, μP8	InL. -5,0	±5	—	NDIP24,WSO24
MAX165	8	1	5	+5	3	μP8	Ext., InL. +1,23	+5	—	DIP18, WSO18

Продолжение табл. 3.10

Тип микро- схемы	Раз- ряд- ность, бит	Коли- чество канал- ов	Максималь- ное время преобразо- вания, мкс	Напряжения питания, В	Ток по- требле- ния, мА	Интерфейс	Опорное напряже- ние, В	Диапазон входных на- пряжений, В	Наличие дифф. входа	Корпус
MAX166	8	1	5	+5	3	μP8	Ext., InL. +1,23	+5	√	DIP20, WSO20
MAX167	12	1	8,33	+5 и -12... -15	4	μP12, μP8	InL. -5,0	±2,5	—	NDIP24, WSO24
MAX170	12	1	5,6	+5 и -12... -15	5	Посл.	InL. - 5,25	+5	—	WSO16, DIP8
MAX172	12	1	10	+5 и -12... -15	5	μP12, μP8	InL. - 5,25	+5	—	NDIP24, WSO24
MAX173	10	1	5	+5 и -12... -15	5	μP12, μP8	InL. - 5,25	+5	—	NDIP24, WSO24
MAX174	12	1	8	+5 и ±12... ±15	3	μP12, μP8	Ext., InL. +10	+10, +20, ±10, ±5	—	DIP28, WSO28
MAX176	12	1	3,5	+5 и -12... -15	5,5	Посл.	InL. -5,0	±5	—	WSO16, DIP8
MAX177	10	1	8,33	+5 и -12... -15	6	μP12, μP8	InL. -5,0	±2,5	—	NDIP24, WSO24
MAX178	12	1	60	±5 и +15	6	μP12, μP8	Ext., InL. +5,0	+5	—	NDIP24, WSO24
MAX180	12	8	7,5	+5 и -12... -15	4,5	μP/16, μP8	Ext., InL. -5,0	+5, ±2,5	√	CDIP40, DIP40, PLCC44
MAX181	12	6	7,5	+5 и -12... -15	4,5	μP/16, μP8	Ext., InL. -5,0	+5, ±2,5	√	CDIP40, DIP40, PLCC44
MAX182	12	4	60	±5 и +15	6	μP8	Ext., InL. +5,0	+5	—	CDIP28, DIP28, WSO28
MAX183	12	1	3	+5 и -12... -15	4	μP12	ExL.	+10, +5, ±5	—	NDIP24, WSO24
MAX184	12	1	5	+5 и -12... -15	4	μP12	ExL.	+10, +5, ±5	—	CDIP24, NDIP24, WSO24
MAX185	12	1	10	+5 и -12... -15	4	μP12	ExL.	+10, +5, ±5	—	CDIP24, NDIP24, WSO24
MAX186	12	8	6	+5, ±5	1,5	Посл.	InL. +4,096	+Vref, ±Vref/2	√	SSOP20, CDIP20, DIP20, WSO20
MAX187	12	1	8,5	4,75...5,25	1,5	Посл.	InL. +4,096	+Vref	—	WSO16, DIP8
MAX188	12	8	6	+5, ±5	1,5	Посл.	ExL.	+5, ±2,5	√	SSOP20, CDIP20, DIP20, WSO20
MAX189	12	1	8,5	4,75... 5,25	1	Посл.	ExL.	+5	—	WSO16, DIP8
MAX190	12	1	7,81	+5	3	μP8, Serial	Ext., InL. +4,096	+5	√	NDIP24, WSO24
MAX191	12	1	7,5	+5, ±5	3	μP8, Serial	Ext., InL. +4,096	+5, ±2,5	√	NDIP24, WSO24
MAX192	10	8	6	+5	1,5	Посл.	Ext., InL. +4,096	+Vref, ±Vref/2	√	SSOP20, DIP20, WSO20

Продолжение табл. 3.10

Тип микро- схемы	Раз- ряд- ность, бит	Коли- чество каналов	Максималь- ное время преобразова- ния, мкс	Напряжения питания, В	Ток по- требле- ния, мА	Интерфейс	Опорное напряже- ние, В	Диапазон входных на- пряжений, В	Наличие дифф. входа	Корпус
MAX194	14	1	9,4	±5	2,5	Посл.	Ext.	+5,±5	—	SB16,DIP16, WSO16
MAX195	16	1	9,4	±5	2,5	Посл.	Ext.	+5,±5	—	DIP16, WSO16
MAX196	12	6	6	4,75...5,25	6	μP12	Ext., InL. +4,096	+10, +5, ±10, ±5	—	SSOP28,DIP28, WSO28
MAX197	12	8	6	4,75...5,25	6	μP8	Ext., InL. +4,096	+10, +5, ±10, ±5	—	SSOP28,DIP28, WSO28
MAX198	12	6	6	4,75...5,25	6	μP12	Ext., InL. +4,096	+Vref,+Vref/2 ±Vref,±Vref/2	—	SSOP28,DIP28, WSO28
MAX199	12	8	6	4,75... 5,25	6	μP8	Ext., InL. +4,096	+Vref,+Vref/2 ±Vref,±Vref/2	—	SSOP28,DIP28, WSO28
MX574A	12	1	25	+5 и ±12... ±15	3	μP12, μP8	Ext., InL. +10	+10, +20, ±10, ±5	—	PLCC28,DIP28, WSO28
MX674A	12	1	15	+5 и ±12... ±15	3	μP12, μP8	Ext., InL. +10	+10, +20, ±10, ±5	—	DIP28,WSO28
MX7572-05	12	1	5	+5 и -15	5	μP12, μP8	InL. - 5,25	+5	—	NDIP24, WSO24
MX7572-12	12	1	12	+5 и -15	5	μP12, μP8	InL. - 5,25	+5	—	NDIP24, WSO24
MX7574	8	1	15	4,75... 5,25	1	μP8	Ext.	+10,±5	—	DIP18,SO18
MX7575	8	1	5	+5	3	μP8	Ext.	+5	—	PLCC20,CDIP18, DIP18,WSO18
MX7576	8	1	10	+5	3	μP8	Ext.	+5	—	
MX7578	12	1	100	±5 и +15	5,5	μP8	Ext.	+5	—	CDIP24,NDIP24, WSO24,PLCC28
MX7581	8	8	66,6	4,5...5,5	3	μP8	Ext.	+10	—	DIP28, SO28
MX7582	12	4	100	±5 и +15	5,5	μP8	Ext.	+5	—	PLCC28,CDIP28, DIP28,SB28,WS O28
MX7672-03	12	1	3	+5 и -12	4	μP12	Ext.	+10,+5,±5	—	LCC28,NDIP24,C DIP24,PLCC28
MX7672-05	12	1	5	+5 и -12	4	μP12	Ext.	+10,+5,±5	—	
MX7672-10	12	1	10	+5 и -12	4	μP12	Ext.	+10,+5,±5	—	
MX7820	8	1	1,34	4,75...5,25	5	μP8	Ext.	+5	—	CDIP20,DIP20, 20/SO
MX7821	8	1	0,66	+5,±5	10	μP8	Ext.	+5,±5	—	PLCC20,DIP20, 20/SO
MX7824	8	4	2,5	4,75...5,25	5	μP8	Ext.	+5	—	SSOP24,CDIP24, NDIP24,WSO24
MX7828	8	8	2,5	4,75...5,25	5	μP8	Ext.	+5	—	CDIP28,SSOP28, DIP28,PLCC28, WSO28

Из приведенной таблицы в соответствии с вышеизложенными критериями можно отобрать, к сожалению, только несколько микросхем, приведенных в табл. 3.11.

Таблица 3.11

Тип микросхемы	Разрядность, бит	Количество каналов	Максимальное время преобразования, мкс	Напряжения питания, В	Ток потребления, мА	Интерфейс	Опорное напряжение, В	Диапазон входных напряжений, В	Наличие диф. входа	Корпус
MAX1240	12	1	7,5	2,7...3,6	1,4	Посл.	Ext., InL. +2,5	+Vref	—	DIP8, SO8
MAX1241	12	1	7,5	2,7...5,25	0,9	Посл.	Ext.	+Vref	—	DIP8, SO8
MAX145	12	1	7	2,7...5,25	0,9	Посл.	Ext.	+Vref	√	CDIP8, 8/11MAX,DIP8
MAX187	12	1	8,5	4,75...5,25	1,5	Посл.	Int. +4,096	+Vref	—	WSO16, DIP8
MAX189	12	1	8,5	4,75...5,25	1	Посл.	Ext.	+5	—	WSO16, DIP8

Аналогичным образом был проведен отбор «претендентов» среди микросхем других производителей (см. табл. 3.12).

Таблица 3.12

Тип микросхемы	Разрядность, бит	Максимальное время преобразования, мкс	Максимальная частота, кГц	Производительность, кSPS	Напряжения питания, В	Мощность, мВт (ток потребления, мА)	Интерфейс	Опорное напряжение, В	Диапазон входных напряжений, В	Корпус
AD7475	12	0,8	—	1000	2,7...5,25	3	SPI	Ext.	V _{ref}	DIP8
AD7495	12	0,8	—	1500	2,7...5,25	3,6	SPI	InL.	V _{ref}	DIP8
AD7851	14	3,8	—	333	5	89	SPI	InL/Ext.	V _{ref} /2	DIP24, SOIC24
AD7853	12	4,6	—	200	3...5	16,5	SPI	InL/Ext.	V _{ref} /2	DIP24, SOIC24
AD7853L	12	4,6	—	100	3...5	6,85	SPI	InL/Ext.	V _{ref} /2	DIP24, SOIC24
AD7893	12	6	—	117	5	45	Посл.	Ext.	2,5–10	DIP8
AD7895	12	3,8	—	200	5	20	Посл.	Ext.	2,5–10	DIP8
AD7896	12	8/14	—	100	3...5	10,8	Посл.	Ext.	V _{DD}	DIP8
ADC12030	12	8,8	—	—	5	10	SPI	Ext.	V _{ref}	SOIC16, TSSOP20
ADC12130	12	8,8	—	—	5	15	SPI	Ext.	V _{ref}	SOIC16, TSSOP20
ADC12H03	12	5,5	—	—	5	15	SPI	Ext.	V _{ref}	SOIC16, TSSOP20
ADS1286	12	20	—	—	5	3,5	Посл.	InL/Ext.	V _{ref}	DIP8, SO8
ADS7812	12	—	200	—	5	35	Посл.	InL.	V _{ref}	DIP16, SO16
ADS7816	12	—	200	—	5	3,5	Посл.	Ext.	0–5	DIP8, SO8, MSOP8
ADS7817	12	—	500	—	5	4	Посл.	InL.	0–2,5	DIP8, SO8, MSOP8

Продолжение табл. 3.12

Тип микро- схемы	Разряд- ность, бит	Максималь- ное время преобразо- вания, мкс	Макси- мальная частота, кГц	Произво- дитель- ность, kSPS	Напря- жения питания, В	Мощность, мВт (ток по- требления, мА)	Интер- фейс	Опорное напряже- ние, В	Диапазон входных напряже- ний, В	Корпус
ADS7818	12	—	500	—	5	15	Посл.	Ext.	0–2,5	DIP8, MSOP8
ADS7822	12	—	500	—	2,7... 3,6	1,625	Посл.	Int.	V_{ref}	DIP8, SO8, MSOP8
ADS7834	12	—	500	—	5	15	Посл.	Int./Ext.	0–2,5	DIP8, MSOP8
ADS7835	12	—	500	—	5	15	Посл.	Ext.	V_{ref}	MSOP8
HI5805	12	—	100	5	5	300	Посл.	Ext.	V_{ref}	SOIC28
HI5808	12	—	100	10	5	325	Посл.	Ext.	V_{ref}	SOIC28
HI5905	14	—	100	5	5	350	Посл.	Ext.	V_{ref}	MQFP44
LTC1286	12	100	12.5	7,5	3...6	1.25	Посл.	Ext.	V_{ref}	SO8
LTC1292	12	—	60	—	5	30	Посл.	Ext.	V_{ref}	PDIP8
LTC1297	12	—	50	—	5	30	Посл.	Ext.	V_{ref}	PDIP8
LTC1400	12	—	400	—	5, ±5	75	Посл.	Int./Ext.	V_{ref}	SO8
LTC1401	12	—	200	200	3	15	Посл.	Int./Ext.	V_{ref}	SO8
LTC1404	12	—	600	600	5	75	Посл.	Int./Ext.	V_{ref}	SO8
MCP3201	12	—	—	100	2,7...5,5	(300 мкА)	SPI	Ext.	V_{ref}	SOIC8, PDIP8
SP8528	12	30	33	—	5	(230 мкА)	Посл.	Ext.	V_{dd}	PDIP8, SOIC8
SP8531	12	4,2	235	—	5	(11,5)	Посл.	Ext.	0–2,5	PDIP8, SOIC8
TLC2551	12	—	—	400	5	15	Посл.	Ext.	V_{ref}	MSOP8, SOIC8
TLC2555	12	—	500	400	5	15	Посл.	Ext.	V_{ref}	MSOP8, SOIC8
TLV2541	12	—	500	200	2,7...5,5	2,3	Посл.	Ext.	V_{ref}	MSOP8, SOIC8
TLV2545	12	—	—	200	2,7...5,5	2,3	Посл.	Ext.	V_{ref}	MSOP8, SOIC8
XRD8794	12	—	—	2	3–5	225	Посл.	Ext.	V_{ref}	PDIP, SOIC

Дальнейший анализ микросхем-«претендентов», с учетом других параметров (не приведенных в таблице) их распространенности и наличия встроенного источника опорного напряжения, позволяет на этом этапе выделить только несколько микросхем *ADC*.

Эти микросхемы приведены в табл. 3.13.

Таблица 3.13

Тип	Разрядность, бит	Напряжения питания, В	Ток потребления, мА	Интерфейс	Опорное напряжение, В	Корпус
MAX1240	12	2,7...3,6	1,4	Посл.	Ext., Int. +2,5	DIP8, SO8
MAX187	12	4,75...5,25	1,5	Посл.	Int. +4,096	WSO16, DIP8
AD7853	12	3...5,5	16,5	SPI	Int., Ext.	DIP8
AD7495	12	2,7...5,25	3,6	SPI	Int.	DIP8

3.8.2. Сопоставительный анализ выбранных аналого-цифровых преобразователей

Рассмотрим функциональные изображения и назначение выводов выбранных аналого-цифровых преобразователей. Функциональные изображения приведены на рис. 3.17.

Приведем короткое описание назначения выводов:

Vdd — напряжение питания;

Ain — аналоговый вход;

DGND, GND — общий провод питания, «земля»;

SCLK — вход тактовых импульсов;

DOUT — последовательный выход данных;

CS / — вход выборки кристалла;

Vref — вход/выход опорного напряжения;

CONV / — вход запуска преобразования;

BUSY — занято;

AGND — аналоговая земля;

SHDN / — вход перехода в режим экономии;

VDR — вход управления на уровне выходных сигналов.

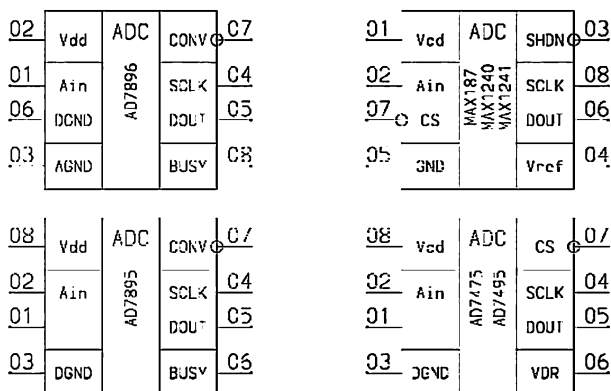


Рис. 3.17. Функциональные изображения выбранных аналого-цифровых преобразователей

Отметим некоторые особенности микросхем.

MAX187 [101] — двенадцатиразрядный аналого-цифровой преобразователь с последовательным многорежимным выходом (в том числе интерфейс *SPI*), встроенным источником опорного напряжения 4,096 В и источником питания +5 В.

MAX1240/1241 [102] — двенадцатиразрядный аналого-цифровой преобразователь с последовательным многорежимным выходом (в том числе интерфейс *SPI*). Микросхема **MAX1240** имеет встроенный источник опорного напряжения 2,5 В, а микросхема **MAX1241** не имеет встроенного источника опорного напряжения. В тоже время, микросхема **MAX1240** имеет пониженное напряжение пи-

тания +2,7—3,6 В, а микросхема *MAX1241* может работать при +5 В. Во всем остальном, в том числе и по назначению и расположению выводов, эти микросхемы полностью соответствуют микросхеме *MAX187*.

AD7475/7495 [103] — двенадцатиразрядный аналого-цифровой преобразователь с последовательным многорежимным выходом (в том числе интерфейс *SPI*). Обе микросхемы поддерживают режим питания от +5 В. Микросхема *AD7495* имеет встроенный источник опорного напряжения 2,5 В, а микросхема *AD7475* не имеет встроенного источника опорного напряжения.

AD7896 [104] — двенадцатиразрядный аналого-цифровой преобразователь с последовательным многорежимным выходом и питанием +5 В.

AD7895 [105] — аналого-цифровой преобразователь с последовательным многорежимным выходом и питанием +5 В. Не имеет встроенного источника опорного напряжения.

Таким образом, имеют встроенные источники опорного напряжения микросхемы *MAX187*, *AD7495*, *AD7896* и *MAX1240*. Однако величина опорного напряжения у *AD7495* и *MAX1240* соответствует +2,5 В, а у *MAX187* — +4,096 В, что накладывает определенные ограничения на схемотехнику входного усилителя. Кроме того, микросхема *MAX1240* работает при пониженном напряжении питания +2,7—3,6 В. Только микросхемы *MAX187*, *MAX1240* и *AD7495* имеют *SPI*-интерфейс. Следует заметить, что у этих микросхем совпадает функциональное назначение только одного из 8 выводов. Все это делает проблематичным разработку универсального посадочного места на печатной плате под все перечисленные микросхемы. Фактически разработчику остается только одно — выбрать одну из микросхем. На наш взгляд, такой микросхемой, безусловно, является *MAX187*.

3.8.3. Вспомогательные микросхемы узлов аналого-цифрового преобразования

Операционные усилители. Обычно перед аналого-цифровым преобразователем устанавливают входной усилитель с перестраиваемым (или программируемым) коэффициентом усиления. Не будем утруждать читателя сведениями о всем многообразии выпускаемых в настоящее время операционных усилителей. Отметим только, что безусловно лучшими параметрами обладают микросхемы *MAX420* [74] и *OP07* [78]. Эти усилители имеют высокий собственный коэффициент усиления, очень малый входной температурный дрейф (0,05 мкВ/°С и 0,2 мкВ/°С соответственно), выпускаются в корпусе *DIP8* и имеют соответствующую разводку выводов (см. рис. 3.18). Кроме того, микросхему *OP07* можно заменить на отечественную КР140УД17А, которая имеет такие же параметры и разводку выводов.

Мультиплексоры. Для коммутации входных сигналов, а в некоторых случаях и режимов масштабирующего усилителя часто используются аналоговые мультиплексоры. Обычно в технологических микроконтроллерах разработчик стремится организовать восемь входных каналов, что соответствует оптимальному соотношению аппаратные затраты/стоимость.

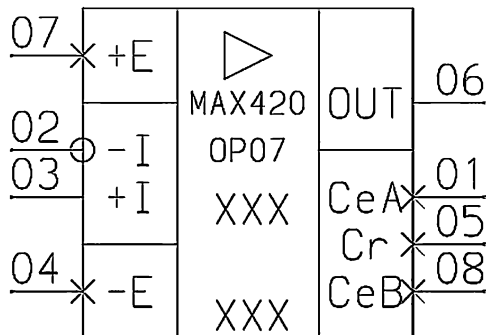


Рис. 3.18. Назначение и разводка выводов операционных усилителей *MAX420* и *OP07*

Наиболее известными, качественными и доступными, безусловно, являются мультиплексоры фирмы *MAXIM*. Эта фирма выпускает на сегодняшний день 317 типов различных мультиплексоров, отличающихся количеством мультиплексируемых каналов, числом мультиплексоров в корпусе, сигналами управления, параметрами каналов и т. п. В табл. 3.14 приведены только мультиплексоры фирмы *MAXIM*, коммутирующие 8 каналов в 1.

Таблица 3.14

Тип микросхемы	Особенности	Max R_{on} , Ом	Min/Max одиночное напряжение питания, В		Min/Max двойное напряжение питания, В		Предельное напряжение защиты, В	Max ток утечки, нА
DG408	Быстродействующий	100	4,5	30	4,5	20	—	1
DG508A		300	—	—	4,5	18	—	2
DG528	Однополярный	400	5	30	4,5	20	—	10
HI-508A	Защищенный по входу	1800	—	—	4,5	18	± 35	2
MAX308	Прецизионный быстродействующий	100	4,5	30	4,5	20	—	0,75
MAX328	Однополярный, с низкими утечками	3500	10	30	4,5	18	—	0,01
MAX338	С низкими утечками	400	4,5	30	4,5	20	—	0,05
MAX349	С последовательным управлением и низким напряжением питания	100	2,7	16	2,7	8	—	0,1
MAX354	С защитой по входу	350	4,5	36	4,5	18	± 40	0,5
MAX358	С защитой по входу	1800	—	—	4,5	18	± 35	2
MAX368	С защитой по входу и буферизацией	1800	—	—	4,5	18	± 35	5
MAX378	Высоковольтный с защитой по входу	3500	—	—	4,5	18	± 75	2
MAX382	С низким питанием и буферизацией	100	2,7	16,5	3	8	—	0,2
MAX388	Высоковольтный с защитой по входу	3000	—	—	4,5	18	± 100	2

Продолжение табл. 3.14

Тип микросхемы	Особенности	Max $R_{\text{он}}$, Ом	Min/Max одиночное напряжение питания, В		Min/Max двойное напряжение питания, В		Предельное напряжение защиты, В	Max ток утечки, нА
MAX398	Прецизионный, низковольтный	100	3	15	3	8	–	1
MAX4051	Низковольтный	100	2	16	2,7	8	–	1
MAX4051A	Низковольтный	100	2	16	2,7	8	–	0,1
MAX4508	Высоковольтный с защитой по входу	400	9	36	4,5	20	±40	2
MAX4530	Низковольтный с разрешением входов и адреса	75	2	12	2	6	–	2
MAX4539	Низковольтный с калибровкой	100	2,7	12	2,7	6	–	0,1
MAX4558	С защитой от ESD 15KV, низковольтный	160	2	12	2	6	–	1
MAX4578	Высоковольтный	350	4,5	36	4,5	20	–	0,05
MAX4581	Низковольтный	80	2	12	2	6	–	1
MAX4617	Высокоскоростной, низковольтный	10	2	5,5	–	–	–	1
MAX4638	Низковольтный	3,5	1,8	5,5	2,5	2,5	–	0,01
MAX4691	Низковольтный	70	2,7	11	2,7	5,5	–	2
MX7503	С низким потреблением	300	10	30	4,5	18	–	10

Критерии отбора мультиплексоров достаточно тривиальны.

1. Желательно, чтобы мультиплексор работал при однополярном напряжении питания +5 В или двухполярном ±10 В.

2. Микросхема должна иметь стандартный «*de facto*» для этих мультиплексоров корпус *DIP16*.

3. Желательно наличие защиты по входному напряжению.

4. Мультиплексор должен иметь высокое быстродействие переключения.

5. Желательно предельно низкое сопротивление открытого канала.

6. Мультиплексор должен иметь параллельные входы управления.

Анализ сведений, приведенных в таблице, а также более внимательное ознакомление с документацией, в частности с расположением и назначением выводов, позволил сделать заключение о наличии двух групп микросхем, отличающихся по электрическим параметрам, но совпадающих по расположению выводов (см. рис. 3.19).

Группа, представленная на рис. 3.19, *a*, является старым «*de facto*» стандартом расположения выводов. На панельку, подключенную с этим расположением выводов, можно устанавливать мультиплексоры *MAX358*, *HI-508*, *MAX354*, *ADG508F*, *MAX378*, *MAX308*, *MAX398*, *MAX333*, *KP590KH6* и ряд других. Вторая группа, представленная на рис. 3.1, *b*, выполнена в соответствии с новым промышленным стандартом на серию *74HC*. Она объединяет *MAX4051*, *MAX4581*, *74HC4051*, *74HC4052*, *74HC4053* и другие, относительно новые микросхемы мультиплексоров.

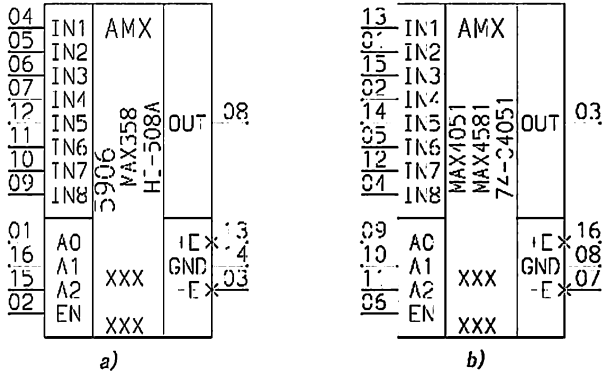


Рис. 3.19. Назначение и разводка выводов двух групп мультиплексов

3.8.4. Принципиальная схема узла аналого-цифрового преобразователя

Один из возможных вариантов принципиальной схемы узла аналого-цифрового преобразования для универсального технологического контроллера представлен на рис. 3.20.

Узел содержит всего три микросхемы: входной аналоговый мультиплексор *A1* (*MAX358*), масштабирующий усилитель *A2* (*MAX420*) и собственно аналого-цифровой преобразователь *A3* (*MAX187*). На входах аналогового мультиплексора *A1* установлена резисторная матрица *R3*. Перемычки *JP1* и *JP2* позволяют включать масштабирующий усилитель *A2* в инвертирующем и неинвертирующем включениях. Перемычки *JP3* позволяют выбирать один из двух коэффициентов усиления, задаваемых соотношением резисторов *R1* и *R2*. Резисторы *R2* могут использоваться как постоянные, так и переменные. Иногда разработчики устанавливают вместо резисторов *R2* специальный зажим для оперативной смены резистора. Аналого-цифровой преобразователь *A3* используется с внутренним источником опорного напряжения (вывод 03 соединен с +5 В). Входы управления аналогового мультиплексора *CA1:CA3*, *EN1* и выходы последова-

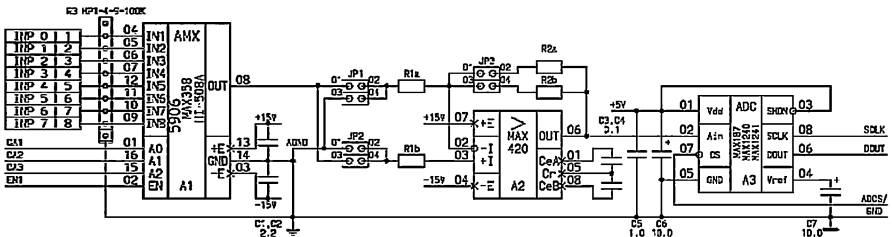


Рис. 3.20. Узел аналого-цифрового преобразования

тельного интерфейса *SPI* аналого-цифрового преобразователя *SCLK*, *DOUT*, *ADCS* / соединяются с соответствующими входами микроконтроллера. Приведенная схема конечно же не исчерпывает всех возможных вариантов исполнения узла. Например, в более сложных узлах вместо перемычек могут быть установлены аналоговые мультиплексоры. Коэффициентов усиления масштабирующего усилителя также может быть несколько, а может быть вообще один. Усилитель может содержать интегрирующие цепи, если это необходимо. Все зависит от выбранного круга задач универсального технологического контроллера и фантазии разработчика.

3.9. Цифроаналоговые преобразователи

3.9.1. Критерии выбора цифроаналоговых преобразователей для универсальных технологических контроллеров

Цифроаналоговым преобразователем (DAC) называется схема, предназначенная для преобразования входного цифрового кода в выходной аналоговый ток или напряжение. Микросхемы цифроаналоговых преобразователей (далее просто *DAC*) выпускаются рядом фирм: *Analog Devices* [69], *Burr-Brown* [95], *Exar* [96], *Fairchild Semiconductor* [97], *Hitachi* [90], *Intersil* [98], *Linear Technology* [73], *Maxim* [74], *National Semiconductor* [99], *Sipex* [76], *Signal Processing Technologies (Toko)* [100], *Texas Instruments* [78] и др. Данные по *DAC*, выпускаемым этими фирмами, можно найти на [91].

Наиболее важными параметрами *DAC* являются разрядность, тип интерфейса, быстродействие, тип корпуса, характеристики питания, тип выхода (по току или по напряжению).

В универсальных технологических контроллерах аналого-цифровые преобразователи *ADC*, как правило, используется с последовательным интерфейсом, т. к. они зачастую выполняют задачи контроля или измерения медленно изменяющихся величин. С цифроаналоговыми преобразователями ситуация несколько сложнее, т. к. они обычно управляют какими-либо исполнительными устройствами, а в конечном итоге процессами. Это обуславливает необходимость применения как медленнодействующих, так и относительно быстродействующих *DAC*.

Разрядность *DAC* определяет точность преобразования. В настоящее время выпускаются микросхемы с разрядностью от 8 до 22 и выше. Обычно в универсальных технологических контроллерах используют *DAC* с разрядностью 8 и 10, что обеспечивает шаг квантования выходной управляемой величины $1/256$ и $1/1024$ соответственно.

Выпускаемые в настоящее время микросхемы *DAC* могут иметь параллельный или последовательный интерфейс. Параллельный интерфейс чаще всего используется для более быстродействующих *DAC*, а последовательный для медленнодействующих. Причем в качестве последовательного интерфейса предпочтительно использовать интерфейс *SPI*, так как большинство современных микроконтроллеров оснащаются аппаратной реализацией этого интерфейса.

Быстродействие цифроаналогового преобразователя также важно. Современные *DAC* по быстродействию можно условно разделить на 3 группы: низкого быстродействия — до 100 ksp/s, среднего быстродействия — от 100 до 500 ksp/s, высокого быстродействия — более 500 ksp/s. Для универсальных технологических контроллеров наиболее часто используются *DAC* низкого или среднего быстродействия исходя из соображений соотношения стоимость/быстродействие.

Современные микросхемы *DAC* производятся с токовым выходом (без встроенного масштабирующего операционного усилителя) и с выходом по напряжению (со встроенным выходным операционным усилителем). Кроме того, в случае, если микросхема в одном корпусе содержит несколько *DAC*, она может иметь комбинированный выход (часть с токовым выходом, часть с выходом по напряжению) или иметь выходы по напряжению с различной нагрузочной способностью. Конечно, с точки зрения минимизации количества корпусов цифроаналогового узла более предпочтителен выход по напряжению. Однако, ввиду того что обычно разработчики универсальных контроллеров стремятся усилить выход, т. е. ставят выходной операционный усилитель, который кроме всего прочего еще и формирует выходное напряжение необходимого уровня, тип выхода микросхемы *DAC* не имеет существенного значения.

Следует также отметить, что современные микросхемы *DAC* могут использовать встроенный или внешний узел опорного напряжения.

Микросхемы *DAC* могут иметь одно или несколько различных напряжений питания. Очевидно, что для технологических микросистем желательно, чтобы микросхема имела одно питание, равное 5 В.

Суммируя вышесказанное, заметим, что обычно в универсальных технологических контроллерах используется достаточно большое количество *DAC* двух типов. Первая группа медленнодействующих преобразователей с разрядностью 10 и последовательным интерфейсом. Эта группа обычно может включать 4, 6 или 8 независимых преобразователей. Вторая группа относительно быстродействующих преобразователей с разрядностью 8. В этой группе могут использоваться микросхемы как с последовательным, так и с параллельным интерфейсом. Обычно эта группа содержит от 2 до 6 преобразователей.

Важное значение при выборе микросхемы имеет тип корпуса. Как уже отмечалось в первом разделе этой главы, оптимальным для технологических контроллеров является корпус типа *DIP*, который позволяет устанавливать микросхему на панельку, что значительно упрощает ремонт и наладку контроллера. Кроме того, желательно, чтобы корпус имел малое количество выводов. Обычно используются микросхемы с количеством выводов от 8 до 16.

3.9.2. Предварительный отбор микросхем *DAC*

В табл. 3.15 приведены наиболее распространенные *DAC* с параллельным интерфейсом (за исключением телевизионных и сверхбыстродействующих). В таблице приведены данные на микросхемы, отсортированные по нарастанию разрядности, а затем — по нарастанию количества каналов в корпусе.

Таблица 3.15

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Опорное напряжение	Корпус
HI5660	8	1	5...3	Напр.	125/60 MSPS	Int.	SOIC28, TSSOP28
AD5330	8	1	2,5...5,5	Напр.	8	Ext.	TSSOP20
AD7524	8	1	+5,+15	Ток	—	Ext.	DIP16, PLCC20, LCCC20
AD7801	8	1	2,7...5,5	Напр.	2	Int./Ext.	SOIC20, TSSOP20
AD9701	8	1	-5,2	Ток	250MSPS	Int.	DIP22, PLCC28
AD9708	8	1	2,7...5,5,+3,+3,3,+5	Ток	35	Int.	SOIC28, TSSOP28
DAC08	8	1	+15,-15	Напр.	70 нс	Ext.	DIP16
DAC0800	8	1	±5...±15	Напр.	100 нс	Ext.	SOIC16, MDIP16
DAC0802	8	1	±5...±15	Напр.	100 нс	Ext.	SOIC18, MDIP16
DAC0808	8	1	±5...±15	Напр.	150 нс	Ext.	SOIC16, MDIP16
DAC0830	8	1	5...15	Напр.	1000 нс	Ext.	MDIP20
DAC0832	8	1	5...15	Напр.	1000 нс	Ext.	SOIC, MDIP, PLCC
HA17008RFP	8	1	н/д	Напр.	150 нс	н/д	SOP16
HA17008RP	8	1	н/д	Напр.	150 нс	н/д	DIP16
HA17408P	8	1	н/д	Напр.	250 нс	н/д	DIP16
HD49307	8	1	н/д	Напр.	30 нс	н/д	SOP56
HI1171	8	1	+5	Напр.	40 MSPS	Ext.	SOIC24
HI3338, CA3338/A	8	1	+5	Напр.	50 MSPS	Ext.	PDIP16, SOIC16
MAX5018	8	1	-5,2	Ток	0,0045	Ext.	NDIP24
MAX5140	8	1	-5,2	Ток	0,003	Ext.	NDIP24
MAX5480	8	1	+5	Ток	0,4	Ext.	QSOP16
MAX7624	8	1	12...15	Ток	0,25	Ext.	DIP16, SO16
MP7523	8	1	4,5...16	Ток	н/д	н/д	PDIP, CDIP, SOIC
MP7524	8	1	4,5...16	Ток	н/д	н/д	PDIP, CDIP, SOIC, PLCC
MP7524A	8	1	4,5...16	ток	н/д	н/д	PDIP, SOIC
MX7224	8	1	15, 11,4...16,5 и -5	Ток	5 (макс)	Ext.	PLCC20, DIP18, WSO18
MX7523	8	1	5...16	Ток	0,15	Ext.	DIP16, WSO16
MX7524	8	1	5...15	Ток	0,4	Ext.	DIP16, SO16
THS5641	8	1	3,0...5,0	н/д	35 нс	н/д	н/д
THS5641A	8	1	3,0...5,0	н/д	35 нс	н/д	н/д

Продолжение табл. 3.15

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Опорное напряжение	Корпус
TLC5602	8	1	5	Напр.	30 нс	Ext.	DIP18
TLC7524	8	1	5...15	Ток	0,1	Ext.	DIP16, PLCC20
AD5332	8	2	2,5...5,5	Напр.	8	Ext.	TSSOP20
AD7302	8	2	2,7...5,5	Напр.	2	Int./Ext.	DIP20, SOIC20, TSSOP20
AD7528	8	2	5...15	Напр.	—	Ext.	DIP20, SOIC20, PLCC20
HI1177	8	2	+5	Напр.	40 MSPS	Ext.	MQFP32
HI5628	8	2	5...3	Напр.	125/60 MSPS	Int.	н/д
MP7528	8	2	4,5...16	Ток	н/д	н/д	PDIP, CDIP, SOIC, PLCC
MP7529A	8	2	4,5...16	Ток	н/д	н/д	PDIP, SOIC, PLCC
MP7529B	8	2	4,5...5,5	Ток	н/д	н/д	PDIP, SOIC, PLCC
MX7528	8	2	5...15	Ток	0,18 (max)	Ext.	DIP20, WSO20
MX7628	8	2	5...15	Ток	0,35 (max)	Ext.	DIP20, WSO20
TLC7528	8	2	5...15	Ток	0.1	Ext.	PDIP20, PLCC20
TLC7628	8	2	11...15	Ток	0.1	Ext.	PDIP20, PLCC20
FMS3810	8	3	5	Напр.	100 МГц	Int/Ext	LQFP48
FMS3815	8	3	5	Напр.	150 МГц	Int/Ext	LQFP48
HI1178	8	3	+5	Напр.	40 MSPS	Ext.	WQFP48
THS8134	8	3	3,0...5,0	Напр.	5 нс	Int.	TQFP48
THS8134A	8	3	3,0...5,0	Напр.	5 нс	Int.	TQFP48
THS8134B	8	3	3,0...5,0	Напр.	5 нс	Int.	TQFP48
TL5632	8	3	5	Напр.	10 нс	Int.	PLCC44
TMC3503x30	8	3	5	Напр.	30 МГц	Int/Ext	PLCC44, TQFP48
TMC3503x50	8	3	5	Напр.	50 МГц	Int/Ext	PLCC44, TQFP48
TMC3503x80	8	3	5	Напр.	80 МГц	Int/Ext	PLCC44, TQFP48
TMC3533x30	8	3	3,3	Напр.	30 МГц	Int/Ext	PLCC44, TQFP48
TMC3533x50	8	3	3,3	Напр.	50 МГц	Int/Ext	PLCC44, TQFP48
TMC3533x80	8	3	3,3	Напр.	80 МГц	Int/Ext	PLCC44, TQFP48
AD5334	8	4	2,5...5,5	Напр.	8	Ext.	TSSOP24
AD7305	8	4	+3/+5	Напр.	1,1	Ext.	DIP20
DAC8408	8	4	+5	Ток	1	Ext.	DIP48

Продолжение табл. 3.15

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Опорное напряжение	Корпус
MAX505	8	4	+5,±5	Напр.	6	Ext.	SSOP24, NDIP24, WSO24
MAX506	8	4	+5,±5	Напр.	6	Ext.	DIP20, WSO20
MP7226	8	4	+5/15	Напр.	н/д	н/д	PDIP, SOIC, PLCC
MP7628	8	4	4,5...5,5	Ток	н/д	н/д	PDIP, CDIP, SOIC, PLCC
MP7643	8	4	+5, +5, +10	Напр.	н/д	н/д	PDIP, SOIC
MX7225	8	4	+15, 12...15 и -5	Напр.	4 (макс.)	Ext.	PLCC28, NDIP24, WSO24
MX7226	8	4	+15, 12...15 и -5	Напр.	4 (макс.)	Ext.	PLCC20, DIP20, SB20, WSO20
TLC7225	8	4	5...15	Напр.	5	Ext.	PDIP24
TLC7226	8	4	15	Напр.	5	Ext.	DIP20
MP7228	8	8	+5/15	Напр.	н/д	н/д	PDIP, CDIP, PLCC, SOIC
MX7228	8	8	+15, 12...15 и -5, +5	Напр.	5 (макс.)	Ext.	PLCC28, NDIP24, WSO24
AD5331	10	1	2,5...5,5	Напр.	9	Ext.	TSSOP20
AD7393	10	1	+3	Напр.	60	Ext.	PDIP20
AD7533	10	1	+15	Ток	—	Ext.	DIP16, PLCC20, LCC20
AD9731	10	1	+5, -5,2	Ток	3,8 нс	Int.	SOIC28, TSSOP28
AD9732	10	1	+5	Ток	4,75 нс	Int.	PDIP28
AD9750	10	1	2,7...5,5	Ток	35	Int.	PDIP28
AD9751	10	1	+3	Ток	35	Int.	LQFP48
AD9760	10	1	2,7...5,5, +3, +3,3, +5	Ток	35 нс	Int.	DIP28
ADV7127	10	1	+3,3, +5	н/д	н/д	н/д	н/д
ADV7128	10	1	+5	н/д	н/д	н/д	н/д
HI20201	10	1	-5.2	Напр.	160 MSPS	Ext.	PDIP28, SOIC28
HI2315	10	1	+5	Напр.	80 MSPS	Int.	MQFP32
HI3197	10	1	+5, +5/-5	Напр.	125 MSPS	Int.	MQFP48
HI5721	10	1	+5/-5,2	Напр.	125 MSPS	Int.	PDIP28, SOIC28
HI5760	10	1	5...3	Напр.	125/60 MSPS	Int.	SOIC28, TSSOP28
HI5780	10	1	+5	Напр.	80 MSPS	Int.	MQFP32

Продолжение табл. 3.15

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Опорное напряжение	Корпус
MAX503	10	1	+5, ±5	Напр.	25	Ext./Int.	SSOP24, NDIP24, WSO24
MP7533	10	1	4,5...16	Ток	н/д	н/д	PDIP, CDIP, SOIC, PLCC
MP7633	10	1	4,5...16	Ток	н/д	н/д	PDIP, CDIP, SOIC, PLCC
MX7520	10	1	5...15	Ток	0.5	Ext.	SB16, DIP16, WSO16
MX7530	10	1	5...15	Ток	0.5	Ext.	SB16, DIP16, WSO16
MX7533	10	1	5...15	Ток	0.6	Ext.	SB16, DIP16, WSO16
THS5651A	10	1	3,0...5,0	Ток	35 нс	Int.	SOIC28
UCC5950	10	1	5	Напр.	2.5		н/д
AD9760-50	10	1	2,7...5,5, +3, +3,3, +5	Ток	35 нс	Int.	DIP28
AD5333	10	2	2,5...5,5	Напр.	9	Ext.	TSSOP24
AD7397	10	2	+3	Напр.	60	Ext.	PDIP24
AD9761	10	2	2,7...5,5, +3, +3,3, +5	Ток	35 нс	Int.	DIP28
AD9763	10	2	3,0...5,5, +3, +3,3, +5	Ток	35 нс	Int.	PLCC48
HI5728	10	2	5...3	Напр.	125/60 MSPS	Int.	LQFP48
FMS3110	10	3	5	Напр.	100 МГц	Int/Ext	LQFP48
FMS3115	10	3	5	Напр.	150 МГц	Int/Ext	LQFP48
HI3050	10	3	+5	Напр.	50 MSPS	Ext.	MQFP64
THS8133	10	3	3,0...5,0	Напр.	5 нс	Int.	TQFP48
THS8133A	10	3	3,0...5,0	Напр.	5 нс	Int.	TQFP48
TMC3003x30	10	3	5	Напр.	30 МГц	Int/Ext	PLCC44, TQFP48
TMC3003x50	10	3	5	Напр.	50 МГц	Int/Ext	PLCC44, TQFP48
TMC3003x80	10	3	5	Напр.	80 МГц	Int/Ext	PLCC44, TQFP48
TMC3033x30	10	3	3,3	Напр.	30 МГц	Int/Ext	PLCC44, TQFP48
TMC3033x50	10	3	3,3	Напр.	50 МГц	Int/Ext	PLCC44, TQFP48
TMC3033x80	10	3	3,3	Напр.	80 МГц	Int/Ext	PLCC44, TQFP48
AD5335	10	4	2,5...5,5	Напр.	9	Ext.	TSSOP24
AD5336	10	4	2,5...5,5	Напр.	9	Ext.	TSSOP28
HI5731	12		+5/-5,2	Напр.	100 MSPS	Int.	PDIP28, SOIC28
AD5340	12	1	2,5...5,5	Напр.	10	Ext.	TSSOP24

Продолжение табл. 3.15

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Опорное напряжение	Корпус
AD5341	12	1	2,5...5,5	Напр.	10	Ext.	TSSOP20
AD7392	12	1	+3	Напр.	60	Ext.	PDIP20
AD7541A	12	1	5...15	Ток	0,6	Ext.	SSOP16, PDIP16
AD7542	12	1	+5	Ток	2	Ext.	DIP16, SOIC20
AD7545A	12	1	5...15	Ток	0,1	Ext.	SSOP20, DIP20
AD7548	12	1	+5, +12, +15	Ток	1,5	Ext.	DIP20, PLCC20, LCCC20
AD9752	12	1	+5	Ток	35 нс	Int.	SOIC28, TSSOP28
AD9753	12	1	+3	Ток	35 нс	Int.	LQFP48
AD9762	12	1	2,7...5,5, +3, +3,3, +5	Ток	35 нс	Int.	DIP28
DAC7541	12	1	5...16	Напр.	1	Ext.	DIP18, SO18
DAC7545	12	1	5...16	Напр.	2	Ext.	DIP18, SO18
DAC7613	12	1	+5	Напр.	10	Ext.	SSOP24
DAC7621	12	1	+5	Напр.	10	Int.	SSOP20
DAC80	12	1	±15	Напр.	4	Int.	DIP24, Hermetic24
DAC811	12	1	±15	Напр.	4	Int.	DIP28, SO28
DAC813	12	1	±15	Напр.	4	Int.	DIP28, SO28
DAC8562	12	1	+5	Напр.	16	Int.	DIP20, SOL20
HA17012P	12	1	н/д	Напр.	250 нс	н/д	DIP20
HI5735	12	1	+5/-5,2	Напр.	80 MSPS	Int.	PDIP28, SOIC28
HI5860	12	1	5...3	Напр.	125 MSPS	Int.	SOIC28
HS7541A	12	1	+15 В при 2 мА	Ток	0,6		DIP18, SOIC18, PLCC20
LTC1450L	12	1	2,7...5,5	Напр.	—	Int/Ext	SSOP24, PDIP24
LTC7541A	12	1	5...15	Ток	0.6	Ext.	SSOP16, PDIP16
LTC7545	12	1	5...15	Ток	0,1	Ext.	SSOP20, DIP20
MAX501	12	1	±12...±15	Напр.	5	Ext.	NDIP24, WSO24
MAX5012	12	1	-5,2	Ток	0,013	Int.	DIP28, PLCC28
MAX5013	12	1	+5, -5,2	Ток	0,013	Int.	DIP28, PLCC28
MAX502	12	1	±12...±15	Напр.	5	Ext.	NDIP24, WSO24
MAX507	12	1	±12...±15	Напр.	5	Int.	NDIP24, WSO24
MAX508	12	1	±12...±15	Напр.	5	Int.	DIP20, WSO20
MAX530	12	1	+5, ±5	Напр.	25	Int.	SSOP24, NDIP24, WSO24

Продолжение табл. 3.15

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Опорное напряжение	Корпус
MAX555	12	1	-5,2	Напр.	0,004	Ext.	Thermally Enhanced PLCC68
MAX7645	12	1	+15	Ток	1 (макс)	Ext.	PLCC20, DIP20, WSO20
MP1230A/31A/32A	12	1	4,5...16	Ток	н/д	н/д	PDIP, CDIP, SOIC
MP7541B	12	1	4,5...16	Ток	н/д	н/д	PDIP, CDIP, SOIC
MP7542	12	1	4,5...5,5	Ток	н/д	н/д	PDIP, CDIP, SOIC
MP7545B	12	1	4,5 – 16	Ток	н/д	н/д	PDIP, SOIC, PLCC
MP7645B	12	1	4,5 – 16	Ток	н/д	н/д	PDIP, CDIP
MX565A	12	1	±15	Ток	0,25	Int.	NDIP24, WSO24
MX566A	12	1	-15	Ток	0,35	Ext.	NDIP24, WSO24
MX7245	12	1	12...15, ±15	Напр.	5 (макс.)	Int.	NDIP24, PLCC28
MX7248	12	1	12...15, ±15	Напр.	5 (макс.)	Int.	DIP20, PLCC20
MX7521	12	1	5...15	Ток	0,5	Ext.	SB18, DIP18, WSO18
MX7531	12	1	5...15	Ток	0,5	Ext.	SB18, DIP18, WSO18
MX7541	12	1	5...16	Ток	1,0 (макс.)	Ext.	DIP18, WSO18
MX7541A	12	1	5...16	Ток	0,6	Ext.	18/PLCC, DIP18, SB18, WSO18
MX7542	12	1	+5	Ток	2 (макс.)	Ext.	DIP16, WSO16
MX7545	12	1	+15, +5	Ток	2 (макс.)	Ext.	DIP20, WSO20
MX7545A	12	1	5...15	Ток	1 (макс.)	Ext.	PLCC20, DIP20, WSO20
MX7548	12	1	12...15, +5	Ток	1 (макс.)	Ext.	PLCC20, DIP20, WSO20
MX7845	12	1	±15	Напр.	5 (макс.)	Ext.	PLCC28, NDIP24, WSO24
SP7545	12	1	+15 В при 2 мА	Ток	2	Ext.	DIP20, WSO20
THS5661A	12	1	3,0...5,0	Ток	35 нс	Int.	SOIC28
TLV5613	12	1	2,7...5,5	Напр.	1	Ext.	TSSOP20
TLV5619	12	1	2,7...5,5	Напр.	1	Ext.	TSSOP20
TLV5633	12	1	2,7...5,5	Напр.	1	Int.	DIP20
TLV5639	12	1	2,7...5,5	Напр.	1	Int.	н/д
AD5342	12	2	2,5...5,5	Напр.	10	Ext.	TSSOP28
AD5343	12	2	2,5...5,5	Напр.	10	Ext.	TSSOP20
AD7396	12	2	+3	Напр.	60	Ext.	PDIP24
AD7537	12	2	+12, +15	Ток	—	Ext.	DIP24, PLCC28

Продолжение табл. 3.15

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Опорное напряжение	Корпус
AD7547	12	2	+12, +15	Ток	1,5	Ext.	DIP24, PLCC28, LCCC28
AD7549	12	2	+15	Ток	1,5	Ext.	DIP20, PLCC20, LCCC20
AD8582	12	2		Напр.	16	Int.	DIP24, SOIC24
AD9765	12	2	3,0...5,5, +3, +3,3, +5	Ток	35 нс	Int.	PLCC48
DAC7801	12	2	+5	Напр.	0,8	Ext.	DIP24
DAC7802	12	2	+5	Напр.	0,8	Ext.	DIP24
DAC8222	12	2	+5, +15	Ток	1	Ext.	DIP24, LCCC28
DAC8248	12	2	+5, +15	Ток	1	Ext.	DIP24
HI5828	12	2	5...3	Напр.	125 MSPS	Int.	LQFP48
HSP50415	12	2	+3,3	Напр.	100 MSPS	Int.	MQFP100
MX7537	12	2	12...15	Ток	1,5 (макс.)	Ext.	PLCC28, NDIP24, WSO24
MX7547	12	2	12...15	Ток	1,5 (макс.)	Ext.	PLCC28, NDIP24, WSO24
MX7549	12	2	5...15	Ток	1,5 (макс.)	Ext.	PLCC20, DIP20, WSO20
MX7837	12	2	$\pm 12... \pm 15$	Напр.	4	Ext.	NDIP24, WSO24
MX7847	12	2	± 15	Напр.	4	Ext.	NDIP24, WSO24
SP9502	12	2	+5 В при 4 мА, -5 В при 4мА	Двуп. напр.	4	н/д	н/д
SP9602	12	2	+5 В при 1,6 мА, -5 В при 1,6мА	Двуп. напр.	30	н/д	н/д
AD5344	12	4	2,5...5,5	Напр.	10	Ext.	TSSOP28
DAC7624	12	4	± 5	Напр.	10	Ext.	DIP28, SO28
DAC7625	12	4	± 5	Напр.	10	Ext.	DIP28, SO28
DAC7724	12	4	+5	Напр.	10	Ext.	SOIC24, DIP24
DAC7725	12	4	+5	Напр.	10	Ext.	SOIC28, PDIP28
MAX526	12	4	12...15 и -5	Напр.	3	Ext.	SB24, NDIP24, WSO24
MAX527	12	4	± 5	Напр.	3	Ext.	SB24, NDIP24, WSO24
MP7680	12	4	4,5...5,5	Ток	н/д	н/д	PDIP, CDIP, PQFP
SP9504	12	4	+5 В при 8 мА, -5 В при 8 мА	Двуп. напр.	4	Ext.	DIP28, SOIC28
SP9604	12	4	+5 В при 3 мА, -5 В при 3 мА	Двуп. напр.	30	Ext.	DIP28, SOIC28

Продолжение табл. 3.15

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Опорное напряжение	Корпус
MP7613	12	8	+12/-12	Напр.	н/д	н/д	PQFP, PGA, PLCC
MAX547	13	8	±5	Напр.	5	Ext.	FP44, PLCC44
AD7534	14	1	+12, +15	Ток	—	Ext.	DIP20, PLCC20
AD7536	14	1	+12, +15, 12...15	Ток	—	Ext.	DIP28, PLCC28
AD7538	14	1	+12, +15	Ток	—	Ext.	DIP24, SOIC24
AD7840	14	1	+15, -15	Напр.	4	Int./Ext.	DIP24, PLCC28
AD9754	14	1	+5	Ток	125MSPS	Int./Ext.	DIP28
AD9755	14	1	+3	Ток	300MSPS	Int./Ext.	PLCC48
AD9764	14	1	2,7...5,5, +3, +3,3, +5	Ток	35 нс	Int.	DIP28
AD9772	14	1	2,7...3,6, +3, +3,3	Ток	35 нс	Int./Ext.	PLCC48
AD9774	14	1	2,7...5,5, +3, +3,3, +5	Ток	35 нс	Int./Ext.	PLCC44
DAC14135	14	1	+5	Напр.	—	Ext.	TSSOP
HI5741	14	1	+5/-5,2	Напр.	100 MSPS	Int.	PDIP28, SOIC28
HI5960	14	1	5...3	Напр.	125 MSPS	Int.	SOIC28
HS3140	14	1	+15 В при 2 мА	Ток	2		DIP20, SOIC20
ISL5314	14	1	5...3	Напр.	125 MSPS	Int.	н/д
LTC1591	14	1	5	Ток	—	Ext.	SSOP28, PDIP28
MP7614	14	1	4,5...16	Ток	н/д	н/д	PDIP, CDIP, SOIC
MX7534	14	1	12...15	Ток	1,5 (макс.)	Ext.	PLCC20, DIP20, SB20, WSO20
MX7535	14	1	12...15	Ток	1,5 (макс.)	Ext.	PLCC28, DIP28, SB28, WSO28
MX7536	14	1	12...15	Ток	1,5 (макс.)	Ext.	PLCC28, DIP28, WSO28
MX7538	14	1	12...15	Ток	1,5 (макс.)	Ext.	PLCC28, NDIP24, SB24, WSO24
SP7514	14	1	+15 В при 2 мА	Ток	2	Ext.	DIP20, SOIC20
THS5671A	14	1	3,0...5,0	Ток	35 нс	Int.	SOIC28
AD9767	14	2	3,0...5,5, +3, +3,3, +5	Ток	35 нс	Int./Ext.	PLCC44
AD7835	14	4	+15, -15, +5	Напр.	10	Ext.	DIP28
AD7836	14	4	+15, -15, +5	Напр.	16	Ext.	MQFP44

Продолжение табл. 3.15

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Опорное напряжение	Корпус
AD7841	14	8	+12, -12, +5; +15, -15, +5	Напр.	31	Ext.	MQFP44
MP7611	14	8	+12/-12	Напр.	н/д	н/д	PQFP, PGA, PLCC
AD569	16	1	+12, -12	Напр.	3	Ext.	DIP28, PLCC28
AD660	16	1	+15, -15, +5	Напр.	13	Int./Ext.	DIP24, SOIC24
AD669	16	1	+15, -15, +5	Напр.	13	Int./Ext.	DIP28, SOIC29
AD768	16	1	+5, -5	Ток	25 нс	Int./Ext.	DIP28
AD7846	16	1	+12, -12, +5; +15, -15, +5	Напр.	6	Ext.	DIP28, PLCC28
DAC16	16	1	+5, -15	Ток	500 нс	Ext.	DIP24
DAC702	16	1	10	Напр.	8	Int.	Hermetic24
DAC703	16	1	±10	Напр.	8	Int.	DIP24, SO24
DAC707	16	1	±15	Напр.	8	Int.	DIP28
DAC708	16	1	±15	Напр.	8	Int.	Hermetic24
DAC709	16	1	±15	Напр.	8	Int.	Hermetic24
DAC712	16	1	±15	Напр.	10	Int.	DIP28, SO28
DAC715	16	1	±15	Напр.	10	Int.	DIP28, SO28
DAC7641	16	1	+5	Напр.	10	Ext.	TQFP-32
DAC7741	16	1	+5	Напр.	5	Int.	TQFP48
LTC1597	16	1	5	Ток	—	Ext.	SSOP28, PDIP28
LTC1599	16	1	5	Ток	—	Ext.	SSOP24, PDIP24
LTC1657	16	1	+5	Напр.	0,1	Int./Ext.	SSOP28, PDIP28
MP7616	16	1	4,5...16	Ток	н/д	н/д	PDIP, CDIP, SOIC
MP7626	16	1	4,5...16,5	Ток	н/д	н/д	PDIP, CDIP, PLCC
MP7636A	16	1	4,5...16,5	Ток	н/д	н/д	SOIC
SP9316	16	1	+15 В при 0,5 мА	Ток	2	н/д	н/д
DAC725	16	2	±15	Напр.	8	Int.	DIP28
DAC7644	16	4	+5	Напр.	10	Ext.	SSOP48
DAC7744	16	4	±5	Напр.	10	Ext.	SSOP48
AD760	18	1	+15, -15, +5	Напр.	13	Int./Ext.	DIP28

Примечание: н/д — означает «Нет данных»; Ext — внешнее; Int — внутреннее.

В табл. 3.16 приведены основные характеристики DAC последовательного типа.

Таблица 3.16

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Интерфейс последовательный	Опорное напряжение	Корпус
LTC1426	6	2	+5	Напр.	—	Pulse Mode	Ext.	PDIP8, SO8
AD5300	8	1	2,7...5	Напр.	4	3-Wire, SPI, μ Wire	Ext.	SOT23, MSOIC8
AD5301	8	1	+3	Напр.	8	I2C, 2-Wire	Ext.	SOT23, MSOIC8
LTC1329-10	8	1	+5	Ток	—	Pulse Mode	Ext.	PDIP8, SO8
LTC1329-50	8	1	+5	Ток	—	Pulse Mode	Ext.	PDIP8, SO8
MAX517	8	1	+5	Напр.	6	2 wire	Ext.	DIP8, SO8
MAX550A	8	1	2,5...5,5	Напр.	4	3-Wire	Ext.	DIP8, μ MAX8
TLV5623	8	1	2,7...5,5	Напр.	3	SPI	Ext.	DIP8
TLV5624	8	1	2,7...5,5	Напр.	1,0 – 3,5	SPI	Int.	DIP8, SO8
XRD5408	8	1	4,5...5,5	Напр.	13	SPI	Ext.	PDIP8, SOIC8
AD5302	8	2	+5	Напр.	8	3-Wire, SPI, μ Wire	Ext.	MSOIC8
AD5303	8	2	2,7...5	Напр.	8	3-Wire, SPI, μ Wire	Ext.	TSSOP16
AD7303	8	2	2,7...5,5	Напр.	2	3-Wire, SPI, μ Wire	Int./Ext.	DIP8, SOIC8
MAX518	8	2	+5	Напр.	6	2-wire	VDD	DIP8, SO8
MAX519	8	2	+5	Напр.	6	2-wire	Ext.	DIP16, SO16
MAX522	8	2	2,7...5,5	Напр.	70	3-Wire	Ext.	DIP8, SO8
MAX548A	8	2	2,5...5,5	Напр.	4	3-Wire	VDD	DIP8, μ MAX8
MAX549A	8	2	2,5...5,5	Напр.	4	3-Wire	Ext.	DIP8, μ MAX8
TLV5625	8	2	2,7...5,5	Напр.	2,5 или 12	SPI, 3-Wire	Ext.	DIP8
TLV5626	8	2	2,7...5,5	Напр.	1	SPI, 3-Wire	Int.	DIP8
MAX512	8	3	$\pm 4,5... \pm 5,5$, 4,5...5,5	Напр.	70	3-Wire	Ext.	DIP14, SO14
MAX513	8	3	$\pm 2,7... \pm 3,6$, 2,7...3,6	Напр.	70	3-Wire	Ext.	DIP14, SO14
AD5304	8	4	2,5...5,5	Напр.	8	3-Wire, SPI, μ Wire	Ext.	MSOIC10
AD5305	8	4	2,5...5,5	Напр.	8	2-Wire	Ext.	MSOIC10
AD5306	8	4	2,5...5,5	Напр.	8	2-Wire	Ext.	TSSOP16
AD5307	8	4	2,5...5,5	Напр.	8	3-Wire, SPI, μ Wire	Ext.	TSSOP16

Продолжение табл. 3.16

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Интерфейс последовательный	Опорное напряжение	Корпус
AD7304	8	4	3...5; ± 5	Напр.	1	3-Wire	Ext.	DIP16, SOIC16, SSOP16
MAX500	8	4	12...15 и -5	Напр.	2,5	3-Wire	Ext.	LCC20, DIP16, WSO16
MAX509	8	4	+5, ± 5	Напр.	6	3-Wire	Ext.	SSOP20, DIP20, WSO20
MAX510	8	4	+5, ± 5	Напр.	6	3-Wire	Ext.	DIP16, WSO16
MAX520	8	4	+5	Напр.	2	2-wire	Ext.	SSOP20, DIP16, WSO16
MAX533	8	4	2,7...3,6	Напр.	6	3-Wire	Ext.	DIP16, QSOP16
MAX534	8	4	+5	Напр.	8	3-Wire	Ext.	DIP16, QSOP16
MP7652	8	4	+5, +5, +10	Напр.	н/д	н/д	н/д	PDIP, SOIC
TLC5620	8	4	5	Напр.	10,0	SPI, 3-Wire	Ext.	DIP14, SO14
TLV5620	8	4	2,7...5,5	Напр.	10	Посл.	Ext.	DIP14
TLV5621	8	4	2,7...5,5	Напр.	10	2-Wire	Ext.	DIP14
TLV5627	8	4	2,7...5,5	Напр.	3	SPI, 3-Wire	Ext.	DIP16
LTC1665	8	8	2,7...5,5	Напр.	30	SPI	Ext.	PDIP16, SO16
MAX521	8	8	+5	Напр.	6	2-wire	Ext.	SSOP24, DIP20, SO24
MAX528	8	8	+5 и -15, 12...15 и -5, +15	Напр.	0,6	3-Wire	Ext.	DIP20, WSO24
MAX529	8	8	+5, ± 5	Напр.	0,6	3-Wire	Ext.	DIP20, WSO24
MP7641	8	8	+5, +5, +10	Напр.	н/д	н/д	н/д	PDIP, SOIC
MP7651	8	8	+5	Напр.	н/д	н/д	н/д	PDIP, SOIC
MP7670	8	8	-5, +5	Напр.	н/д	н/д	н/д	PDIP, SOIC
MP8840	8	8	-5, +5	Напр.	н/д	н/д	н/д	PDIP, SOIC
SP9840	8	8	+5 В при 14 мА	Напр.	0,7	н/д	н/д	н/д
SP9841	8	8	+5В при 13мА	Напр.	0,7	н/д	н/д	н/д
SP9842	8	8	+5В при 13мА	Напр.	0,7	н/д	н/д	н/д
SP9843	8	8	+5 В при 14 мА	Напр.	0,7	н/д	н/д	н/д
TLC5628	8	8	5	Напр.	10,0	Посл.	Ext.	DIP16, SO16
TLV5628	8	8	2,7...5,5	Напр.	10	Посл.	Ext.	DIP16
AD5310	10	1	2,7...5,5	Напр.	8	3-Wire, SPI, μ Wire	Ext./VDD	SOT23, MSOIC8
AD5311	10	1	+3	Напр.	9	I2C, 2-Wire	Ext.	SOT23, MSOIC8

Продолжение табл. 3.16

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Интерфейс последовательный	Опорное напряжение	Корпус
AD7391	10	1	2,7...5,5	Напр.	70	3-Wire, SPI, Ser- μ C	Ext.	DIP8, SO8, TSSOP8
LTC1427-50	10	1	+5	Ток	—	SMB/I ² C	Ext.	PDIP8, SO8
LTC1427-50	10	1	+5	Ток	—	SMB/I ² C	Ext.	PDIP8, SO8
LTC1663	10	1	2,7...5,5	Напр.	30	SPI	Int.	MS8, SOT23
MAX504	10	1	+5, \pm 5	Напр.	25	3-Wire	Ext./Int.	DIP14, SO14
MAX515	10	1	+5	Напр.	25	3-Wire	Ext.	DIP8, SO8
MAX5304	10	1	+5	Напр.	10	3-Wire	Ext.	DIP8, μ MAX8
MAX5354	10	1	+5	Напр.	10	3-Wire	Ext.	DIP8, μ MAX8
MAX5355	10	1	+3,3	Напр.	10	3-Wire	Ext.	DIP8, μ MAX8
TLC5615	10	1	5	Напр.	12,5	3-Wire	Ext.	DIP8, SO8
TLV5606	10	1	2,7...5,5	Напр.	3/9	SPI	Ext.	DIP8
XRD5410	10	1	4,5...5,5	Напр.	13	SPI	Ext.	PDIP8, SOIC8
AD5312	10	2	+5	Напр.	9	3-Wire, SPI, μ Wire	Ext.	MSOIC8
AD5313	10	2	2,7...5	Напр.	9	3-Wire, SPI, μ Wire	Ext.	TSSOP16
AD7395	10	2	2,7...5,5	Напр.	70	Посл.	Ext.	DIP14, SOIC14
LTC1661	10	2	2,7...5,5	Напр.	30	SPI	Ext.	PDIP8, SO8
LTC1662	10	2	2,7...5,5	Напр.	750	SPI	Ext.	PDIP8, SO8
MAX5158	10	2	4,5...5,5	Напр.	8	3-Wire	Ext.	DIP16, QSOP16
MAX5159	10	2	2,7...3,6	Напр.	8	3-Wire	Ext.	DIP16, QSOP16
TLC5617	10	2	5	Напр.	2,5	SPI, 3-Wire	Ext.	SO8
TLV5617A	10	2	2,7...5,5	Напр.	2,5	SPI, 3-Wire	Ext.	DIP8
TLV5637	10	2	2,7...5,5	Напр.	1	SPI, 3-Wire	Int.	DIP8
AD5314	10	4	2,5...5,5	Напр.	9	3-Wire, SPI, μ Wire	Ext.	MSOIC10
AD5315	10	4	2,5...5,5	Напр.	9	2-Wire	Ext.	MSOIC10
AD5316	10	4	2,5...5,5	Напр.	9	2-Wire	Ext.	TSSOP16
AD5317	10	4	2,5...5,5	Напр.	8	3-Wire, SPI, μ Wire	Ext.	TSSOP16
LTC1664	10	4	2,7...5,5	Напр.	19	SPI	Ext.	PDIP16, SO16
MAX5250	10	4	+5	Напр.	10	3-Wire	Ext.	DIP20, SSOP20
MAX5251	10	4	3,0...3,6	Напр.	12	3-Wire	Ext.	DIP20, SSOP20
TLV5604	10	4	2,7...5,5	Напр.	3	SPI, 3-Wire	Ext.	DIP16, SO16

Продолжение табл. 3.16

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Интерфейс последовательный	Опорное напряжение	Корпус
LTC1660	10	8	2,7...5,5	Напр.	30	SPI	Ext.	PDIP16, SO16
AD5320	12	1	2,7...5,5	Напр.	10	3-Wire, SPI, μ Wire	Ext.	SOT23, MSOIC8
AD5321	12	1	+3	Напр.	10	I2C, 2-Wire	Ext.	SOT23, MSOIC8
AD7390	12	1	2,7...5,5	Напр.	70	3-Wire, SPI, Ser- μ C	Ext.	DIP8, SO8, TSSOP8
AD7543	12	1	+5	Напр.	2	Посл.	Ext.	DIP16, PLCC20
DAC7512	12	1	н/д	Напр.	10	н/д	н/д	MSOP8, SOT23
DAC7611	12	1	н/д	Напр.	10	н/д	Int.	SO8
DAC8043A	12	1	+5	Напр.	н/д	н/д	Ext.	
DAC8143	12	1	+5	Напр.	н/д	н/д	Ext.	
LTC1257	12	1	4,75... 15,75	Напр.	6	SPI	Ext.	PDIP8, SO8
LTC1451	12	1	+5	Напр.	14	SPI	Ext.	PDIP8, SO8
LTC1452	12	1	3...5	Напр.	14	SPI	Ext.	PDIP8, SO8
LTC1453	12	1	+3	Напр.	14	SPI	Ext.	PDIP8, SO8
LTC1456	12	1	+5	Напр.	14	SPI	Int.	PDIP8, SO8
LTC1659	12	1	2,7...5,5	Напр.	14	SPI	Ext.	PDIP8, SO8
LTC7543	12	1	+5	Ток	1	Посл.	Ext.	PDIP16, SO16
LTC8043	12	1	+5	Ток	1	3-Wire	Ext.	PDIP8, SO8
LTC8143	12	1	+5	Ток	1	Посл.	Ext.	PDIP16, SO16
MAX5120	12	1	+5	Напр.	20	3-Wire	Ext./Int.	QSOP16
MAX5121	12	1	+3	Напр.	20	3-Wire	Ext./Int.	QSOP16
MAX5122	12	1	+5	Напр.	20	3-Wire	Ext./Int.	QSOP16
MAX5123	12	1	+3	Напр.	20	3-Wire	Ext./Int.	QSOP16
MAX5174	12	1	4,5...5,5	Напр.	12	3-Wire	Ext.	QSOP16
MAX5175	12	1	4,5...5,5	Напр.	12	3-Wire	Ext.	QSOP16
MAX5176	12	1	2,7...3,6	Напр.	12	3-Wire	Ext.	QSOP16
MAX5177	12	1	2,7...3,6	Напр.	12	3-Wire	Ext.	QSOP16
MAX5302	12	1	+5	Напр.	14	3-Wire	Ext.	DIP8, μ MAX8
MAX531	12	1	+5, \pm 5	Напр.	25	3-Wire	Ext./Int.	DIP14, SO14
MAX5352	12	1	+5	Напр.	14	3-Wire	Ext.	DIP8, μ MAX8
MAX5353	12	1	+3,3	Напр.	14	3-Wire	Ext.	DIP8, μ MAX8
MAX538	12	1	+5	Напр.	25	3-Wire	Ext.	DIP8, SO8

Продолжение табл. 3.16

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Интерфейс последовательный	Опорное напряжение	Корпус
MAX539	12	1	+5	Напр.	25	3-Wire	Ext.	DIP8, SO8
MAX543	12	1	+15, +5	Ток	0,25	3-Wire	Ext.	WSO16, DIP8, SO8
MAX551	12	1	+5	Ток	1,0	3-Wire	Ext.	μMAX10, DIP8
MAX552	12	1	2,7...3,6	Ток	1,0	3-Wire	Ext.	μMAX10, DIP8
MP7543	12	1	4,5...5,5	Ток	н/д	н/д	н/д	PDIP, CDIP, SOIC, PLCC
MX7543	12	1	+5	Ток	2 (max)	3-Wire	Ext.	DIP16, WSO16
SP9500	12	1	+5V@0,22ма	Напр.	7,5	3-Wire	Ext.	DIP8, SOIC8
SP9501	12	1	+5V@1,6ма, -5V@1,6ма	Двуп. напр.	4	3-Wire	Ext.	DIP8, SOIC8
SP9600	12	1	+5V@0,1ма	Напр.	20	3-Wire	Ext.	DIP8, SOIC8
SP9601	12	1	+5V@0,2ма, -5V@0,2ма	Двуп. напр.	30	3-Wire	Ext.	DIP8, SOIC8
TLV5616	12	1	2,7...5,5	Напр.	3	SPI	Ext.	DIP8, SO8
TLV5636	12	1	2,7...5,5	Напр.	1	SPI	Int.	DIP8, SOIC8
XRD5412	12	1	4,5...5,5	Напр.	13	SPI	Ext.	PDIP8, SOIC8
AD5322	12	2	+5	Напр.	10	3-Wire, SPI, μWire	Ext.	MSOIC8
AD5323	12	2	2,7...5	Напр.	10	3-Wire, SPI, μWire	Ext.	TSSOP16
AD7394	12	2	2,7...5,5	Напр.	70	Посл.	Ext.	DIP14, SOIC14
DAC7612	12	2	н/д	Напр.	10	н/д	Int.	SO8
DAC7800	12	2	н/д	Напр.	0,8	н/д	н/д	DIP16
LTC1446	12	2	+5	Напр.	14	SPI	Int.	PDIP8, SO8
LTC1446L	12	2	+3	Напр.	14	SPI	Int.	PDIP8, SO8
LTC1448	12	2	3...5	Напр.	14	SPI	Ext.	PDIP8, SO8
LTC1454	12	2	+5	Напр.	14	SPI	Int./Ext.	PDIP16, SO16
LTC1454L	12	2	+3	Напр.	14	SPI	Int./Ext.	PDIP16, SO16
LTC1590	12	2	+5	Ток	0,3	SPI	Ext.	PDIP16, SO16
MAX5104	12	2	4,5...5,5	Напр.	15	3-Wire	Ext.	DIP16, QSOP16
MAX5154	12	2	4,5...5,5	Напр.	15	3-Wire	Ext.	DIP16, QSOP16
MAX5155	12	2	2,7...3,6	Напр.	15	3-Wire	Ext.	DIP16, QSOP16
MAX5156	12	2	4,5...5,5	Напр.	15	3-Wire	Ext.	DIP16, QSOP16
MAX5157	12	2	2,7...3,6	Напр.	18	3-Wire	Ext.	DIP16, QSOP16

Продолжение табл. 3.16

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Интерфейс последовательный	Опорное напряжение	Корпус
MAX532	12	2	$\pm 12 \dots \pm 15$	Напр.	2,5	3-Wire	Ext.	DIP16, WSO16
TLC5618A	12	2	5	Напр.	2,5	3-Wire	Ext.	DIP8, SO8
TLV5618A	12	2	2,7...5,5	Напр.	2,5	SPI	Ext.	DIP8, SO8, LCCC20
TLV5638	12	2	2,7...5,5	Напр.	1	SPI	Int.	DIP8, SO8, LCCC20
AD5324	12	4	2,5...5,5	Напр.	10	3-Wire, SPI, μ Wire	Ext.	MSOIC10
AD5325	12	4	2,5...5,5	Напр.	10	2-Wire	Ext.	MSOIC10
AD5326	12	4	2,5...5,5	Напр.	10	2-Wire	Ext.	TSSOP16
AD5327	12	4	2,5...5,5	Напр.	8	3-Wire, SPI, μ Wire	Ext.	TSSOP16
DAC7614	12	4	н/д	Напр.	10	н/д	н/д	DIP16, SO16
DAC7615	12	4	н/д	Напр.	10	н/д	н/д	DIP16, SO16
DAC7714	12	4	н/д	Напр.	10	н/д	н/д	SSOP16, SOIC20
DAC7715	12	4	н/д	Напр.	10	н/д	н/д	SSOP20, PDIP20
LTC1458	12	4	+5	Напр.	14	SPI	Int./Ext.	SSOP28
LTC1458L	12	4	+3	Напр.	14	SPI	Int./Ext.	SSOP28
MAX514	12	4	+5	Ток	0,25	3-Wire	Ext.	NDIP24, WSO28
MAX525	12	4	+5	Напр.	12	3-Wire	Ext.	DIP20, SSOP20
MAX5253	12	4	3,0...3,6	Напр.	16	3-Wire	Ext.	DIP20, SSOP20
MAX536	12	4	12...15 и -5	Напр.	3	3-Wire	Ext.	SB16, DIP16, WSO16
MAX537	12	4	± 5	Напр.	3	3-Wire	Ext.	SB16, DIP16, WSO16
TLV5614	12	4	2,7...5,5	Напр.	3	SPI	Ext.	DIP16, SO16
AD7568	12	8	4,75...5,25; +5	Ток	—	Посл.	Ext.	PLCC44, PQFP44
MP7612	12	8	+12/-12	Напр.	н/д	н/д	н/д	PDIP, SOIC
MAX5130	13	1	+5	Напр.	20	3-Wire	Ext./Int.	QSOP16
MAX5131	13	1	+3	Напр.	20	3-Wire	Ext./Int.	QSOP16
MAX5132	13	1	+5	Напр.	20	3-Wire	Ext./Int.	QSOP16
MAX5133	13	1	+3	Напр.	20	3-Wire	Ext./Int.	QSOP16
MAX535	13	1	+5	Напр.	16	3-Wire	Ext.	DIP8, μ MAX8
MAX5351	13	1	+3,3	Напр.	16	3-Wire	Ext.	DIP8, μ MAX8
MAX5150	13	2	+5	Напр.	16	3-Wire	Ext.	DIP16, QSOP16

Продолжение табл. 3.16

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Интерфейс последовательный	Опорное напряжение	Корпус
MAX5151	13	2	+3	Напр.	16	3-Wire	Ext.	DIP16, QSOP16
MAX5152	13	2	+5	Напр.	20	3-Wire	Ext.	DIP16, QSOP16
MAX5153	13	2	+3	Напр.	20	3-Wire	Ext.	DIP16, QSOP16
AD5531	14	1	+12, -12; +15, -15	Напр.	1	3-Wire, SPI, μWire	Ext.	SOIC8
AD7840	14	1	+15, -15	Напр.	4	Ser + Parallel	Int./Ext.	DIP24, SOIC24, PLCC28
LTC1658	14	1	2,7...5,5	Напр.	12	SPI	Ext.	PDIP8, SO8
MAX5170	14	1	4,5...5,5	Напр.	18	3-Wire	Ext.	QSOP16
MAX5171	14	1	4,5...5,5	Напр.	18	3-Wire	Ext.	QSOP16
MAX5172	14	1	2,7...3,6	Напр.	18	3-Wire	Ext.	QSOP16
MAX5173	14	1	2,7...3,6	Напр.	18	3-Wire	Ext.	QSOP16
MAX544	14	1	+5	Напр.	1	3-Wire	Ext.	DIP8, SO8
MAX545	14	1	+5	Напр.	1	3-Wire	Ext.	DIP14, SB14, SO14
MAX5544	14	1	+5	Напр.	1	3-Wire	Ext.	DIP8, SO8
AD7244	14	2	+5, -5	Напр.	—	Посл.	Int./Ext.	
AD7834	14	4	+15, -15, +5	Напр.	—	Ser+ Parallel- Byte	Ext.	DIP28, SOIC28, PLCC44
MP7610	14	8	+12/-12	Напр.	н/д	н/д	н/д	PDIP, SOIC
AD5531/2/3	14	32	8...16,5, -4,75...-16,5, 4,75... 5,25, 2,7...5,25	Напр.	22	30(/2)	SPI	Int /Ext
AD420	16	1	12...32	Ток	2500	3-Wire	Int./Ext.	PDIP24, SOIC24
AD421	16	1	+3,0, +5	Ток	200	3-Wire	Int./Ext.	PDIP16, SOIC16
AD5541	16	1	+5	Напр.	1	3-Wire, SPI, μWire	Ext.	SOIC8
AD5542	16	1	+5	Напр.	1	3-Wire, SPI, μWire	Ext.	SOIC8
AD660	16	1	+15, -15, +5	Напр.	13	Ser+ Parallel- Byte	Int./Ext.	DIP24, SOIC24
AD766	16	1	+5, -5; +5, - 12; +12, -12	Напр.	1,5	3-Wire	Int.	DIP16
AD7849	16	1	+15, -15, +5	Напр.	10	н/д	Ext.	DIP20
DAC714	16	1	н/д	Напр.	10	н/д	Int.	DIP16, SO16
DAC716	16	1	н/д	Напр.	10	н/д	Int.	DIP16, SO16

Продолжение табл. 3.16

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Интерфейс последовательный	Опорное напряжение	Корпус
DAC7631	16	1	н/д	Напр.	10	н/д	н/д	SSOP20
DAC7731	16	1	н/д	Напр.	5	н/д	Int.	TQFP48
LTC1595	16	1	+5	Ток	1	SPI	Ext.	PDIP16, SO16
LTC1596	16	1	+5	Ток	1	SPI	Ext.	PDIP16, SO16
LTC1650	16	1	+5, ± 5	Напр.	4	SPI	Ext.	PDIP16, SO16
LTC1655	16	1	+5	Напр.	20	SPI	Int./Ext.	PDIP8, SO8
LTC1655L	16	1	2,7...5,5	Напр.	20	SPI	Int./Ext.	PDIP8, SO8
MAX541	16	1	+5	Напр.	1	3-Wire	Ext.	DIP8, SO8
MAX542	16	1	+5	Напр.	1	3-Wire	Ext.	DIP14, SB14, SO14
MAX5541	16	1	+5	Напр.	1	3-Wire	Ext.	DIP8, SO8
AD5544	16	4	+5	Ток	1/2	3-Wire, SPI	Ext.	SSOP28
DAC7634	16	4	н/д	Напр.	10	н/д	н/д	SSOP48
DAC7734	16	4	н/д	Напр.	10	н/д	н/д	SSOP48
AD760	18	1	+15, -15, +5	Напр.	13	Ser+ Parallel-Byte	Int./Ext.	PDIP28
DAC1220	20	1	н/д	Напр.	10000	н/д	н/д	SSOP16
DAC1221	20	1	н/д	Напр.	10000	н/д	н/д	SSOP16

Перед предварительным отбором микросхем сформулируем, в качестве примера, требования к подсистеме параллельных DAC. Допустим, что нам необходимо оснастить универсальный технологический контроллер шестью независимыми 8-разрядными цифроаналоговыми преобразователями (допустимо и 10-разрядными при незначительном повышении стоимости). При этом желательно обеспечить малое энергопотребление, питание предпочтительно от одного источника +5 В (в случае, если это нереализуемо, допускается использование дополнительных напряжений питания ± 15 В). Желательно также занять минимальную площадь печатной платы.

На стадии предварительного отбора сперва удалим из таблицы 3.15 все микросхемы с разрядностью 12 и выше. Затем удалим микросхемы: с питанием только 3,3 В; с питанием только выше 5 В (т. к. они, как правило, требуют дополнительных узлов согласования по входу); с питанием -5 В, т. к. это предполагает наличие дополнительного источника питания; ЭСЛ-микросхемы с питанием -5,2 В; а также микросхемы с более чем тремя напряжениями питания. Далее необходимо исключить из рассмотрения все микросхемы с высоким быстродействием из соображений высокой стоимости и высокого энергопотребления. Следует так же поступить и с микросхемами, не имеющими исполнения в DIP-корпусах.

Полученную таким образом сокращенную таблицу необходимо рассматривать исходя из соображений минимизации стоимости, количества корпусов и площади печатной платы. В нашем примере, как было указано выше, необходимо разработать подсистему из 6 независимых каналов. Очевидно, что при разработке подсистемы желательно использовать одинаковые микросхемы параллельных *DAC*. Исходя из этого, можно рассмотреть следующие варианты реализации:

- 1) шесть корпусов одноканальных 8/10-разрядных *DAC*;
- 2) три корпуса двухканальных *DAC*;
- 3) два корпуса четырехканальных *DAC* (с избыточностью в 2 канала);
- 4) один корпус восьмиканального *DAC* (с избыточностью в 2 канала).

Даже беглого ознакомления с полученной таблицей достаточно для того, чтобы определить, что все одноканальные *DAC* выпускаются в основном в корпусе *DIP16* и выше. Из этого следует, что вариант исполнения подсистемы совершенно невыгоден, т. к. подсистема будет занимать очень большую площадь печатной платы и иметь высокую стоимость. В связи с этим исключим из таблицы все одноканальные микросхемы и перегруппируем полученную таблицу по критерию количества каналов в корпусе, а затем по разрядности.

В заключение проведем анализ доступности микросхем. Для этого необходимо посетить сайты отечественных наиболее крупных представителей вышеперечисленных фирм и сайты известных отечественных поставщиков комплектации. В результате из нашей таблицы исчезнут еще несколько позиций. Кроме того, ознакомимся с документацией претендентов на сайтах производителей и заполним недостающие в таблице данные. Полученные в результате предварительного отбора микросхемы параллельных *DAC* приведены в табл. 3.17.

Таблица 3.17

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Интерфейс	Источник опорного напряжения	Корпус
AD7302	8	2	2,7–5,5	V		uP/8	Внутр./Внеш.	DIP20
AD7528, MX7528, PM7528, TLC7528	8	2	5 – 15	V	0,18	uP/8		
AD7628, MX7628, PM7628, TLC7628			12 – 15					
AD7397	10	2	2,7–5,5	V		uP/12	Внеш.	DIP24
AD7305	8	4	5	V	<2,6МГц	uP/8	Внеш.	DIP20
TLC7225	8	4	5 – 15	V	5	uP/8	Внеш.	DIP24
MAX505, MX7225	8	4	5	V	6	uP/8	Внеш.	DIP24
MAX506, MX7226								DIP20

Аналогичным образом поступим и с микросхемами последовательных *DAC*.

Сформулируем, в качестве примера, требования к подсистеме последовательных *DAC*. Допустим, что нам необходимо оснастить универсальный технологический контроллер шестью независимыми 8-разрядными цифроаналоговыми

преобразователями (допустима и более высокая разрядность при незначительном повышении стоимости), а также тремя независимыми 10-разрядными преобразователями (допустима и более высокая разрядность). При этом желательно обеспечить малое энергопотребление. Питание предпочтительно от одного источника +5 В (в случае если это нереализуемо, допускается использование дополнительных напряжений питания ± 15 В). Желательно также занять минимальную площадь печатной платы. По изложенным выше соображениям выбираем тип последовательного интерфейса — *SPI*.

На стадии предварительного отбора сначала удалим из табл. 3.16 все микросхемы с разрядностью выше 10. Далее удаляем все микросхемы, которые заведомо не поддерживают интерфейс *SPI*. Затем удалим микросхемы с питанием только 3,3 В, с питанием только выше 5 В (т. к. они, как правило, требуют дополнительных узлов согласования по входу), с питанием –5 В (т. к. это предполагает наличие дополнительного источника питания), а также микросхемы с более чем тремя напряжениями питания.

Как и в предыдущем случае, таблицу необходимо рассматривать исходя из соображений минимизации стоимости, количества корпусов и площади печатной платы. Как и в случае с параллельными *DAC*, нам не имеет смысла рассматривать одноканальные микросхемы.

Полученные в результате предварительного отбора микросхемы последовательных *DAC* приведены в табл. 3.18.

Таблица 3.18

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Интерфейс последовательный	Опорное напряжение	Корпус
AD7303	8	2	5	V		SPI	Внутр./Внеш.	DIP8
MAX522	8	2	5	V	70	SPI	Внеш.	DIP8
TLV5625	8	2	2,7...5,5	V	2,5 или 12	SPI	Внеш.	DIP8
TLV5626	8	2	2,7...5,5	V	1	SPI	Внутр.	DIP8
MAX548A, 549A	8	2	5	V	4	SPI	VDD	DIP8
MAX512	8	3	5	V	70	SPI	Внеш.	DIP14
AD7304	8	4	5	V		SPI	Внеш.	DIP16
MAX509	8	4	5	V	6	SPI	Внеш.	DIP20
MAX510	8	4	5	V	6	SPI	Внеш.	DIP16
MAX534	8	4	5	V	8	SPI	Внеш.	DIP16
TLC5620, TLV5620	8	4	2,7...5,5	V	10	SPI	Внеш.	DIP14
TLV5627	8	4	2,7...5,5	V	3	SPI	Внеш.	DIP16
LTC1665	8	8	5	V		SPI	Внеш.	DIP16
TLC5628, TLV5628	8	8	2,7...5,5	V	10	SPI	Внеш.	DIP16
LTC1661	10	2	5	V		SPI	Внеш.	DIP8
LTC1662	10	2	5	V		SPI	Внеш.	DIP8

Продолжение табл. 3.18

Тип	Разрядность	Количество DAC	Напряжение питания, В	Выход	Быстродействие, мкс	Интерфейс последовательный	Опорное напряжение	Корпус
MAX5158	10	2	5	V	8	SPI	Внеш.	DIP16
TLC5617A, TLV5617A	10	2	2,7...5,5	V	2,5	SPI	Внеш.	DIP8
TLV5637	10	2	2,7...5,5	V	1	SPI	Внеш.	DIP8
LTC1664	10	4	5	V		SPI	Внеш.	DIP16
MAX5250	10	4	5	V	10	SPI	Внеш.	DIP20
TLV5604	10	4	2,7...5,5	V	3	SPI	Внеш.	DIP16
LTC1660	10	8	5	V		SPI	Внеш.	DIP16

3.9.3. Анализ отобранных микросхем параллельных DAC

Напомним читателю, что нами были сформулированы требования к подсистеме параллельных цифроаналоговых преобразователей (DAC). В качестве примера рассматривался вариант оснащения универсального технологического контроллера шестью независимыми 8-разрядными цифроаналоговыми преобразователями (допускалось использование и 10-разрядных микросхем при незначительном повышении стоимости). При этом подсистема должна обладать малым энергопотреблением, питанием от одного источника +5 В (в случае если это не реализуемо, допускается использование дополнительных напряжений питания ± 15 В). Кроме того, подсистема должна занимать минимальный объем печатной платы. Были также определены общие требования к используемым микросхемам при проектировании универсальных технологических контроллеров: микросхемы должны иметь корпус DIP, сравнительно низкую стоимость и высокую доступность.

Для проведения последующего анализа необходимо ознакомиться с документацией на каждую из отобранных микросхем. Результаты ознакомления позволили сформировать из отобранных микросхем 6 групп по совпадению выводов микросхем. Три группы двухканальных DAC и три группы — четырехканальных. Расположение выводов отобранных микросхем показано на рис. 3.21.

Первая группа двухканальных DAC состоит только из микросхемы AD7302 — двухканального восьмиразрядного цифроаналогового преобразователя (рис. 3.21, а). Микросхема выпускается в корпусе DIP20, функционирует в диапазоне от 2,7 В до 5,5 В, имеет режим энергосбережения (*Power Down Mode*), потребление в нормальном режиме не превышает 5 мА, может работать от встроенного или внешнего источника опорного напряжения, аналоговые выходы буферизированы. Входные параллельные цифровые входы дважды буферизированы, т. е. в первые регистры информация записывается с шины по сигналам записи WR/ при подаче активного сигнала выборки кристалла CS/ и переключателя записи в канал B/A. При этом аналоговые выходы могут изменять свое состояние сразу при записи в первый регистр (при LDAC/ постоянно равным нулю) либо по стробирующему сигналу LDAC/ (активный низкий).

D7 □ 1 20 □ DGND	AGND □ 25 □ CU'B	OUTA □ 1 24 □ OUTB
D6 □ 2 19 □ OUTA	CU'A □ 2 9 □ RF3B	AGND □ 2 23 □ REF
D5 □ 3 18 □ OUTB	RF3A □ 3 8 □ REF3	DGND □ 3 22 □ VDD
D4 □ 4 17 □ AGND	REFA □ 4 7 □ VDD	LDA/ □ 4 21 □ LDB/
D3 □ 5 16 □ REFIN	DGND □ 5 6 □ WR/	SDN/ □ 5 20 □ A/B
D2 □ 6 15 □ VDD	3/A □ 6 5 □ CS/	CLR/ □ 6 19 □ CS/
D1 □ 7 14 □ CLR/	D7 □ 7 4 □ DC	NC □ 7 18 □ D9
D0 □ 8 13 □ LDAC/	D6 □ 8 3 □ D1	NC □ 8 17 □ D8
CS/ □ 9 12 □ PD/	D5 □ 9 2 □ D2	D0 □ 9 16 □ D7
WR/ □ 10 11 □ B/A	D4 □ 10 1 □ D3	D1 □ 10 15 □ D6
		D2 □ 11 14 □ D5
		D3 □ 12 13 □ D4
a) AD7302 2×DAC8	b) AD7528 2×DAC8	c) AD7397 2×DAC10
OUTB □ 1 24 □ OUTC	OUTB □ 25 □ OUTC	OUTB □ 1 20 □ OUTC
OUTA □ 2 23 □ OUTD	CU'A □ 2 9 □ CU'D	OUTA □ 2 19 □ OUTD
VSS □ 3 22 □ VDD	VSS □ 3 8 □ VDD	VSS □ 3 18 □ VDD
REFB □ 4 21 □ REFC	REF □ 4 7 □ AC	REF □ 4 17 □ AO
REFA □ 5 20 □ REFD	3ND □ 5 6 □ A1	AGND □ 5 16 □ A1
AGND □ 6 19 □ AO	LDAC/ □ 6 5 □ WR/	DGND □ 6 15 □ WR/
DGND □ 7 18 □ A1	D7 □ 7 4 □ DC	D7 □ 7 14 □ D0
LDAC/ □ 8 17 □ WR/	D6 □ 8 3 □ D1	D6 □ 8 13 □ D1
D7 □ 9 16 □ D0	D5 □ 9 2 □ D2	D5 □ 9 12 □ D2
D6 □ 10 15 □ D1	D4 □ 10 1 □ D3	D4 □ 10 11 □ D3
D5 □ 11 14 □ D2		
D4 □ 12 13 □ D3		
d) MAX505 4×DAC8	e) AD7305 2×DAC8	f) MAX506 4×DAC8

Рис. 3.21. Параллельные цифроаналоговые преобразователи

Во вторую группу двухканальных восьмиразрядных цифроаналоговых преобразователей входит 8 микросхем, совместимых по назначению выводов: *AD7528*, *MX7528*, *PM7528*, *TLC7528*, *AD7628*, *MX7628*, *PM7628*, *TLC7628* (рис. 3.21, б). Микросхемы также выпускаются в корпусе DIP20, функционируют от ~4 В до +16,5 В. Микросхема имеет отдельные управляемые входы опорного напряжения для каждого из каналов. Микросхемы не имеют двойной буферизации по входу и аналоговой — по выходу.

В третью группу входит микросхема двухканального десятиразрядного цифроаналогового преобразователя *AD7397* (рис. 3.21, с). Микросхема выполнена в корпусе DIP24. Микросхема отличается сверхнизким энергопотреблением, при питании от 2,7 В до 5,5 В ток потребления обычно составляет 0,125 мА и гарантируется его не превышение 0,2 мА. Микросхема *DAC* имеет двойную буферизацию по входу с отдельным стробированием каждого канала (сигналы *LDA/* и *LDB/*). Выходы также буферизированы и обеспечивают режим энергосбережения. Кроме того, микросхема имеет вход сброса данных для обоих цифроанало-

говых преобразователей. Следует отметить, что на цифровые входы одновременно должны подаваться 10 двоичных разрядов, что не очень удобно для микроконтроллерных систем.

В четвертую группу входят три микросхемы четырехканальных восьмиразрядных цифроаналоговых преобразователей: *MAX505*, *MX7225*, *TLC7225* (рис. 3.21, d). Эти микросхемы выпускаются в корпусе *DIP24*. Микросхемы могут работать от одного источника питания 4,5 В — 5,5 В или от двух источников (второй от –5 В до 0 В). Потребление микросхемы составляет обычно 5 мА. Микросхемы имеют отдельные входы опорного напряжения и двойную объединенную буферизацию по входу.

Пятая и шестая группы представлены микросхемами четырехканальных восьмиразрядных цифроаналоговых преобразователей: *AD7305* и группой микросхем *MAX506*, *TLC7226*, *MX7226*. Обе группы микросхем выпускаются в корпусе *DIP20*. Различие между группами фактически сводится к различию назначения одного вывода. Микросхемы могут работать от одного источника питания 4,5 В — 5,5 В или от двух источников (второй от –5 В до 0 В). Потребление микросхемы составляет обычно 5 мА. Все микросхемы буферизированы по выходу. Микросхема пятой группы *AD7305* имеет двойную буферизацию по входу со входом стробирования второго буфера *LDAC/*. Если он постоянно соединен с нулем, то микросхема не имеет режима вторичной буферизации и по выводам полностью соответствует микросхемам шестой группы.

Анализ приведенных сведений, а также состояние рынка отечественных поставщиков микросхем позволяет остановить наш выбор на микросхемах шестой группы, в частности на *MAX506*. Эта микросхема оказалась в наличии почти у всех отечественных поставщиков, а стоимость всех приведенных микросхем изменяется в диапазоне от \$6 до \$7. Принципиальная схема проектируемой подсистемы будет приведена далее.

3.9.4. Анализ отобранных микросхем последовательных *DAC*

В разделе 3.9.3 были сформулированы требования к цифроаналоговой подсистеме, выполненной на базе последовательных *DAC* с последовательным интерфейсом *SPI*. Необходимо было оснастить универсальный технологический контроллер шестью независимыми 8-разрядными цифроаналоговыми преобразователями (допустима и более высокая разрядность при незначительном повышении стоимости), а также тремя независимыми 10-разрядными преобразователями (допустима и более высокая разрядность). При этом необходимо было обеспечить малое энергопотребление, питание от одного источника +5 В, подсистема должна занимать минимальную площадь печатной платы.

Проведенный ранее предварительный отбор позволил отобрать ряд микросхем-претендентов, которые были приведены в табл. 3.17.

Рассмотрим отдельно двухканальные, четырехканальные и восьмиканальные последовательные *DAC*.

Расположение выводов последовательных двухканальных микросхем *DAC* показано на рис. 3.22.

OUTA □ 1 ~ 8 □ OUTB VDD □ 2 7 □ SYN/ GND □ 3 6 □ DIN REF □ 4 5 □ SCLK a) AD7303 2*DAC8	CS/ □ 1 ~ 8 □ DIN SCLK □ 2 7 □ REF VDD □ 3 6 □ OUTB GND □ 4 5 □ OUTA b) MAX522 2*DAC8	DIN □ 1 ~ 8 □ VDD SCLK □ 2 7 □ OUTB CS/ □ 3 6 □ REF OUTA □ 4 5 □ AGND c) TLV5625 TLV5617 2*DAC8 2*DAC10
GND □ 1 ~ 8 □ VDD OUTA □ 2 7 □ OUTB CS/ □ 3 6 □ LDAC/ DIN □ 4 5 □ SCLK d) MAX548 2*DAC8	CS/ □ 1 ~ 8 □ OUTA SCLK □ 2 7 □ GND DIN □ 3 6 □ VDD REF □ 4 5 □ OUTB e) LTC1661/1662 2*DAC10	AGND □ 1 ~ 16 □ VDD OUTA □ 2 15 □ OUTB OSA □ 3 14 □ OSB REFA □ 4 13 □ REFB CL/ □ 5 12 □ POL/ CS/ □ 6 11 □ UPO DIN □ 7 10 □ DOUT SCLK □ 8 9 □ DGND f) MAX5158 2*DAC10

Рис. 3.22. Двухканальные микросхемы цифроаналоговых преобразователей с интерфейсом SPI

Все двухканальные преобразователи разделены на шесть групп по типу корпуса и расположению выводов.

Первую группу представляет двухканальный восьмиразрядный цифроаналоговый преобразователь *AD7303*, вторую — *MAX522*. В третью группу входят два двухканальных восьмиразрядных DAC — *TLV5625*, *TLV5626* и два двухканальных десятиразрядных DAC — *TLV5617A* и *TLV5637*. Четвертую группу представляет микросхема *MAX548*. В пятую группу входят две двухканальные десятиразрядные микросхемы *LTC1661* и *LTC1662*. Все микросхемы первых пяти групп выпускаются в корпусе DIP8. В шестую группу входит только двухканальный десятиразрядный DAC — *MAX5158*, выпускаемый в корпусе DIP16. Все микросхемы имеют последовательный интерфейс SPI и соответствующие ему сигналы управления *DIN* (вход последовательных данных), *SCLK* (вход синхронизации). Конечно же, имеются и некоторые отличия. Например, все микросхемы кроме *AD7303* (первая группа) имеют обычный вход выборки кристалла *CS/*, а в микросхеме *AD7303* вход *SYN/* выполняет двойную функцию: при переходе потенциала на нем из высокого уровня в низкий происходит выборка кристалла, а по положительному фронту сигнала срабатывает вторая ступень внутренней буферизации, что приводит к обновлению данных. В микросхеме *MAX548* имеется отдельный вход управления второй ступенью внутренней буферизации — *LDAC/*. Микросхема *MAX5158* имеет расширенный набор функ-

ций, например сброс данных, отдельные входы опорного напряжения для каждого из каналов и т. п.

Микросхемы четырехканальных цифроаналоговых преобразователей показаны на рис. 3.23.

Как видно из рассмотрения рис. 3.23, разброс корпусов в этой категории достаточно велик, от DIP14 до DIP20. Следует отметить, что многие микросхемы этой категории имеют вход сброса, почти все имеют двухступенчатую буферизацию по входу. Многие микросхемы имеют отдельные входы опорного напряжения или сгруппированные по два. К сожалению, критерию доступности соответствуют только микросхемы фирмы *MAXIM*.

На рис. 3.24 представлены две группы восьмиканальных микросхем и одна трехканальная микросхема.

Наибольший интерес из них представляет первая группа микросхем, в которую входит как восьмиразрядная микросхема *LTC1665*, так и десятиразрядная микросхема *LTC1660*. Эти микросхемы выполнены в корпусе DIP16. Следует отметить, что по выводам с ними совпадает микросхема четырехканального десяти-

OUTB □ 1 16 □ OUTC OUTA □ 2 15 □ OUTD VSS □ 3 14 □ VDD REFA □ 4 13 □ REFC REFB □ 5 12 □ REF D GND □ 6 11 □ DIN LDAC/ □ 7 10 □ SCLK CLR/ □ 8 9 □ CS/	OUTB □ 20 □ OUTC OUTA □ 2 9 □ OUTD VSS □ 3 8 □ VDD REFB □ 4 7 □ REFC REFA □ 5 6 □ REF D AGND □ 6 5 □ CS/ NC □ 7 4 □ VC DGND □ 8 3 □ SCLK LDAC/ □ 9 2 □ DIN DOLT □ 10 1 □ C_R/	OUTB □ 1 16 □ OUTC OUTA □ 2 15 □ OUTD REF □ 3 14 □ AGND UPD □ 4 13 □ VDD PDE □ 5 12 □ DGND DAC/ □ 6 11 □ DIN CL_R/ □ 7 10 □ SCLK DOUT □ 8 9 □ CS/
a) AD7304 4*DAC8	b) MAX509 4*DAC8	c) MAX534 4*DAC8
GND □ 1 14 □ VDD RFFA □ 2 13 □ LDAC REFB □ 3 12 □ OUIA REFC □ 4 11 □ OUTB REF D □ 5 10 □ OUTC DIN □ 6 9 □ OUTD SCLK □ 7 8 □ LDAC	DVCD □ 6 □ AVDD PD/ □ 2 5 □ R=AB LDAC/ □ 3 4 □ OJIA DIN □ 4 3 □ OJTB SCLK □ 5 2 □ OJIC CS/ □ 6 1 □ OJTD FS □ 7 0 □ R=CD DGND □ 8 9 □ AGND	AGND □ 1 20 □ VDD FBA □ 2 19 □ FB D OUIA □ 3 18 □ OUID OUTB □ 4 17 □ OUTC FBB □ 5 16 □ FBC RFAB □ 6 15 □ RFCD CLR/ □ 7 14 □ PD/ CS/ □ 8 13 □ UPD DIN □ 9 12 □ DOUT SCLK □ 10 11 □ DGND
d) TLV5620 4*DAC8	e) TLV5627 TLV5604 4*DAC8 4*DAC1C	f) MAX5250 MAX525 4*DAC10 4*DAC12

Рис. 3.23. Четырехканальные цифроаналоговые преобразователи с интерфейсом *SPI*

GND □ 1	16 □ VDD	OUTB □ 1	16 □ OUTC	DIN □ 1	14 □ DOUT
OUTA □ 2	15 □ OUTH	OUTA □ 2	15 □ OUTD	CS/ □ 2	13 □ LC
OUTB □ 3	14 □ OUTG	GND □ 3	14 □ REF1	SCLK □ 3	12 □ RFAB
OUTC □ 4	13 □ OUTF	DIN □ 4	13 □ LDAC	CLR/ □ 4	11 □ REFC
OUTD □ 5	12 □ OUTE	SCLK □ 5	12 □ LOAD	VDD □ 5	10 □ OUTC
REF □ 6	11 □ CLR/	VDD □ 6	11 □ REF2	GND □ 6	9 □ OUTB
CS/ □ 7	10 □ DOUT	OUTE □ 7	10 □ OUTH	VSS □ 7	8 □ OUTA
SCLK □ 8	9 □ DIN	OUTF □ 8	9 □ OUTG		
a)		b)		c)	
LTC1665 8*DAC8		TLV5628 8*DAC8		MAX512 3*DAC8	
LTC1660 8*DAC10					

Рис. 3.24. Восьмиканальные и трехканальные цифроаналоговые преобразователи с интерфейсом SPI

тиразрядного DAC — LTC1664, у которой отсутствуют старшие четыре выхода микросхем LTC1660/1665. Однако все восьмиканальные микросхемы достаточно редки.

3.9.5. Варианты принципиальных схем подсистемы цифроаналоговых преобразователей

Рассмотрим несколько вариантов реализации цифроаналоговой подсистемы универсального технологического контроллера. При этом мы будем ориентироваться на первоначальное задание: наличие в подсистеме как минимум шести каналов восьмиразрядных и как минимум шести каналов десятиразрядных DAC.

Наиболее красиво с точки зрения схемотехники поставленная задача решается при выполнении цифроаналоговой подсистемы на двух восьмиканальных микросхемах DAC с последовательным интерфейсом SPI: восьмиразрядной LTC1665 и десятиразрядной LTC1660. Принципиальная схема такой подсистемы представлена на рис. 3.25. Она состоит всего из двух микросхем в корпусе DIP16. Особенностью такой подсистемы является то, что используемые восьмиразрядная и десятиразрядная микросхемы имеют одинаковое назначение выводов. Это обстоятельство позволяет легко модернизировать подсистему в подсистему 16 восьмиразрядных или десятиразрядных DAC. К сожалению, используемые в подсистеме микросхемы достаточно сложно приобрести.

Как уже указывалось ранее, при проектировании универсальных технологических контроллеров может возникнуть необходимость в обеспечении высокого быстродействия части подсистемы цифроаналогового преобразования, например, если эта подсистема предназначена для генерации сигналов специальной формы с достаточно высокой частотой. В этом случае часть подсистемы DAC или вся подсистема должна быть выполнена на более быстродействующих микросхемах с параллельным интерфейсом. Однако следует помнить, что такое решение

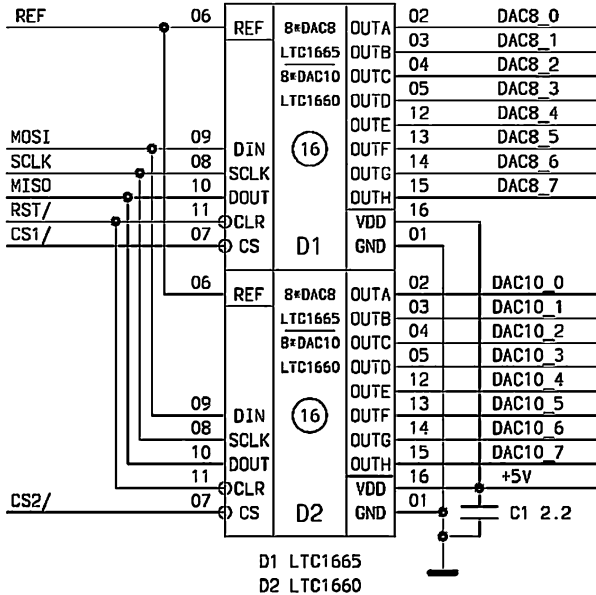


Рис. 3.25. Подсистема цифроаналоговых преобразователей на восьмиканальных микросхемах

предполагает наличие дополнительных аппаратных затрат на организацию параллельной магистрали для обмена между контроллером и подсистемой. С другой стороны, использование параллельного интерфейса значительно упрощает программное обеспечение микроконтроллера. Пример выполнения восьмиразрядной части подсистемы *DAC* на микросхемах с параллельным интерфейсом показан на рис. 3.26.

Следующий вариант реализации проектируемой подсистемы выполнен на двухканальных восьмиразрядных микросхемах *TLV5625* и десятиразрядных микросхемах *TLV5617A* (рис. 3.27). Этот вариант хорош тем, что позволяет оперативно изменять количество установленных *DAC*, а также легко модернизировать состав подсистемы в сторону 16 восьмиразрядных или десятиразрядных каналов. Недостатком варианта является более высокая стоимость по сравнению с первым вариантом (в 4—5 раз). Очевидно, что по аналогии можно собрать часть любой другой представленной подсистемы на двухканальных микросхемах. Стоит еще отметить, что двухканальные восьмиразрядные микросхемы выпускаются многими производителями, а десятиразрядные микросхемы значительно более редкие.

В заключение приведем принципиальную схему подсистемы на самых распространенных и легкодоступных микросхемах фирмы *MAXIM* (см. рис. 3.28). Эта подсистема выполнена на двух четырехканальных восьмиразрядных микро-

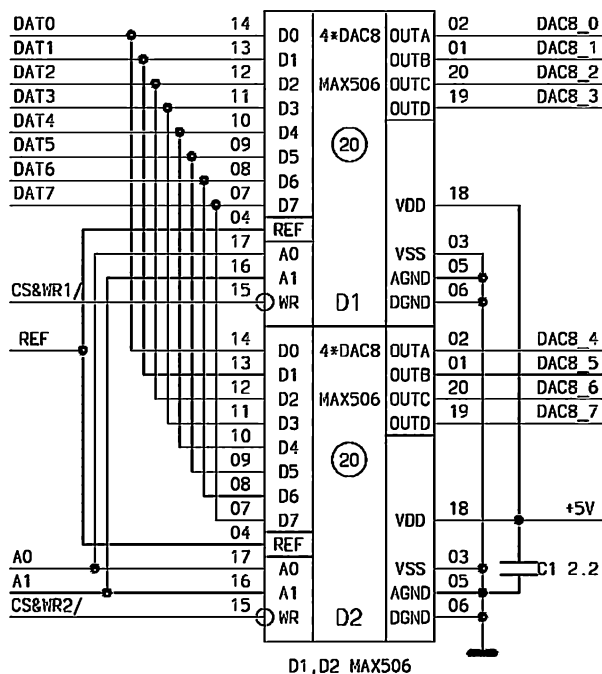


Рис. 3.26. Подсистема восьмиразрядных цифроаналоговых преобразователей на микросхемах с параллельным интерфейсом

схемах *MAX509* и двух четырехканальных десятиразрядных микросхемах *MAX5250*. Все используемые микросхемы выполнены в корпусе DIP20, что приводит к увеличению площади печатной платы, занимаемой подсистемой, примерно в 2,2 раза по сравнению с первым вариантом, увеличению стоимости примерно в 1,8 раза. Можно заменить микросхемы *MAX509* на их аналоги *MAX510*, выпускаемые в корпусе DIP16, это позволит снизить увеличение площади печатной платы до 1,9 раз. К сожалению, эта микросхема более дефицитна. Другой особенностью рассматриваемой подсистемы является то, что микросхему *MAX5250* можно заменить на *MAX525*, преобразуя, таким образом, десятиразрядную часть в двенадцатиразрядную. Несмотря на все, отмеченные выше проигрыши (в стоимости и площади), этот вариант наиболее просто реализуем, т. к. используемые микросхемы есть практически у всех крупных поставщиков микросхем.

В заключение следует отметить, что на приведенных рисунках были показаны только собственно микросхемы цифроаналоговых преобразователей. На самом деле в реальную подсистему должен входить достаточно мощный и стабильный источник опорного напряжения, выход которого подключается к сигнальной линии *REF*. В зависимости от назначения системы, таких источников

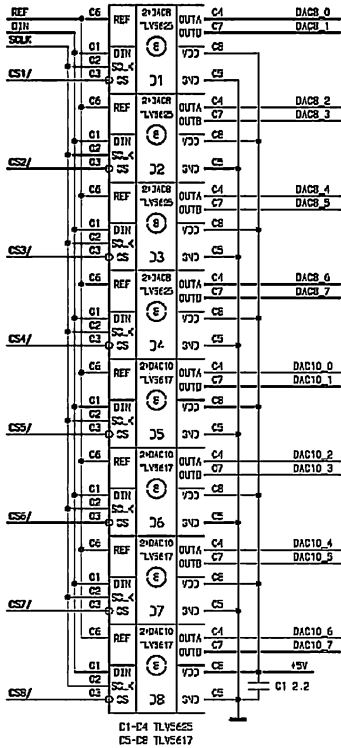


Рис. 3.27. Подсистема DAC на двух-канальных микросхемах

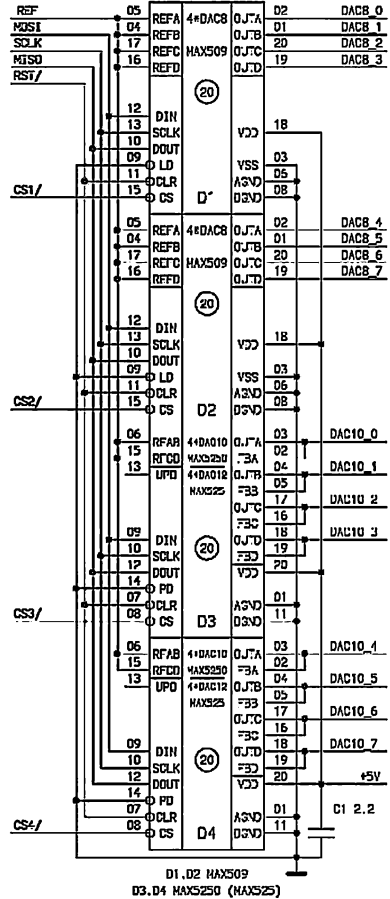


Рис. 3.28. Подсистема DAC на микросхемах фирмы MAXIM

может быть несколько, на разные напряжения. Кроме того, несмотря на то что все использованные микросхемы имеют буферизацию по выходному напряжению, в реальном изделии нам необходимо поставить еще на каждый выходной канал по масштабирующему выходному усилителю для повышения нагрузочной способности и обеспечения необходимого уровня выходного напряжения.

4. Проектирование систем малой автоматизации

4.1. Выбор микроконтроллеров для систем малой автоматизации

Проблема выбора микроконтроллера для систем малой автоматизации является, пожалуй, одной из самых главных. Выбранный микроконтроллер (или семейство микроконтроллеров) определяет не только важнейшие технические показатели системы (производительность, объем памяти программ, объем памяти данных, количество микросхем, массогабаритные показатели, стоимость и надежность системы), но и принципы конструктивного исполнения.

Как уже отмечалось в предыдущей главе, в современных универсальных технологических контроллерах и системах малой автоматизации наиболее часто используются микроконтроллеры семейства *iMCS-51TM (i8051)*, являющиеся de facto промышленным стандартом среди микроконтроллеров. Известно что несмотря на тот факт, что эти микроконтроллеры существуют уже более 20 лет, и сегодня в мире около 70 % промышленных изделий используют именно их. Они выпускаются многими фирмами в различных усовершенствованных вариантах, часто называемых x51-совместимыми микроконтроллерами.

Возникает вопрос о причинах столь высокой живучести этих микроконтроллеров. Обычно на этот вопрос разработчики приводят следующие доводы:

- во-первых, в семействе x51 реализована очень удачная и удобная архитектура;
- во-вторых, x51-совместимые микроконтроллеры очень хорошо документированы. И по аппаратной реализации этого семейства, и по схемотехнике их применения, и по их программированию опубликовано достаточно много разнообразной научно-технической литературы — книг, статей, рекомендаций по применению;
- в-третьих, за время существования этого семейства для него было разработано очень большое количество качественного и доступного программного обеспечения: компиляторов различных языков программирования (*ASM51, C, C++, PL/M51, Fort51* и т. п.), дизассемблеров, программных отладчиков, эмуляторов и т. п. Кроме того, существует большое количество программ и библиотек для различных научно-технических задач;
- в-четвертых, у большинства разработчиков, занимающихся микроконтроллерной техникой, имеется достаточно большой опыт общения с этим семейством микроконтроллеров, изучены многие особенности их поведения, выработаны приемы и способы отладки, накоплен опыт программирования.

X51-совместимые микроконтроллеры выпускаются многими известными фирмами, среди которых: *Atmel Corp.* [60—62], *Cygnal Integrated Products*

[64—66], *Dallas Semiconductor* [70], *Hyundai* [108], *Infineon (Siemens)* [109], *Integrated Circuit Solution* [110], *Integrated Silicon Solutions* [111], *Intel* [112], *MAXIM* [74], *Oki Semiconductor* [113], *Philips Semiconductors* [114], *Silicon Storage Technology* [115], *SyncMOS* [116], *Winbond* [117] и многие другие. Наибольшей популярностью в нашей стране пользуются микроконтроллеры фирм *Atmel* и *Maxim*, а в последнее время — молодой, но динамично развивающейся фирмы *Cygnal*. Познакомимся более подробно с изделиями этих фирм.

4.1.1. Последние достижения фирмы *Atmel* в области x51-совместимых микроконтроллеров

До недавнего времени наибольшей популярностью в нашей стране пользовались микроконтроллеры фирмы *Atmel* [60—62]. Рассмотрим более подробно перечень выпускаемых этой фирмой микроконтроллеров. Первое поколение микроконтроллеров фирмы *Atmel* отличалось от стандартной конфигурации i8051 практически только встроенной *Flash*-памятью программ. За последние несколько лет фирма *Atmel* создала достаточно много новых интересных микроконтроллеров семейства x51 — так называемое второе поколение. На сегодняшний день семейство представлено двенадцатью группами, включающими около 70 микроконтроллеров!

Прежде чем познакомиться с каждой из групп, необходимо знать принятую терминологию, основные сокращения и понятия.

Многие из микроконтроллеров фирмы *Atmel* оснащены встроенной *Flash*-памятью программ — *IROM (Internal ROM)* и встроенной основной оперативной памятью — *IRAM (Internal RAM)*. Некоторые современные микроконтроллеры второго поколения имеют кроме *IRAM* еще и встроенную дополнительную оперативную память — *XRAM (Extended RAM)*, что безусловно делает эти микросхемы еще более мощными и привлекательными. Кроме того, некоторые микроконтроллеры имеют встроенную *Flash*-память данных — *IDROM (Internal Data ROM)*, которая может использоваться для хранения настраиваемых параметров, сетевых адресов и другой информации.

Практически все микроконтроллеры по стандартным или расширенным набором линий ввода/вывода (или портов) обладают возможностью работы с внешней памятью программ или данных. В приведенных ниже таблицах в графе *EM (External Memory)* плюсом отмечена такая возможность. Если в этой колонке проставлен минус — микроконтроллер не может работать с внешней памятью. Как правило, это связано с уменьшенным (ограниченным) количеством линий ввода/вывода вследствие исполнения микросхем в маловыводном корпусе. Микроконтроллеры семейства x51 в стандартной конфигурации (в стандартных корпусах) обеспечивают работу с внешней памятью объемом до 64 К. Микроконтроллеры, у которых в соответствующей колонке проставлен другой меньший объем памяти имеют уменьшенное количество линий ввода/вывода. Если объем адресуемой памяти больше 64 К, микроконтроллеры имеют расширенную шину адреса или аппаратный диспетчер памяти.

Различные микроконтроллеры имеют различное количество портов ввода/вывода и, соответственно, разное количество линий ввода/вывода — *I/O (Input/Output Lines)*. Микроконтроллеры семейства x51 в стандартной конфигурации имеют 4 порта или 32 линии ввода/вывода. У других микроконтроллеров количество линий может отличаться в большую или меньшую сторону. Все микроконтроллеры семейства x51 имеют встроенный последовательный порт — *SP (Serial Port)*. Наличие плюса в одноименной колонке означает наличие стандартного последовательного порта. Некоторые микроконтроллеры в соответствующей колонке имеют вместо плюса букву «E» (*Enhanced*) — с расширенными возможностями, например, с автоматическим обнаружением ошибки фрейма, автоматическим определением адреса, с специализированным генератором скорости передачи и т. д.

Все микроконтроллеры семейства x51 имеют несколько таймеров/счетчиков общего назначения *T/C (Timer/Counter)* и источников прерывания *IS (Interrupt Source)*. Количество этих периферийных устройств указано в соответствующих колонках. В некоторых микроконтроллерах имеется также встроенный аппаратно реализованный последовательный периферийный интерфейс — *SPI (Serial Peripheral Interface)*. Для выхода из возможных состояний «зависания» микроконтроллера служит аппаратно реализованный сторожевой таймер *WDT (Watchdog Timer)*, включаемый в состав периферийных узлов многих микроконтроллеров второго поколения. В некоторых микроконтроллерах, выпускаемых в малых корпусах с сокращенным набором линий ввода/вывода (*Small Footprint*), имеется встроенный аналоговый компаратор — *AC (Analog Comparator)*. Некоторые современные микроконтроллеры оснащаются двумя регистрами-указателями данных *DPTR (Data Pointer Register)*, что позволяет создавать для них более эффективное программное обеспечение.

Большинство микроконтроллеров первого поколения программировались на специальных параллельных программаторах. Многие микроконтроллеры второго поколения позволяют производить программирование непосредственно в изделии через специальный узел — *ISP (In-System Programmable Flash)*. При этом процедура программирования предполагает подачу на микроконтроллер некоторых управляющих сигналов. Некоторые последние версии микроконтроллеров имеют так называемую загрузочную программу (*Boot*), упрощающую режим программирования.

В последнее время наряду с перепрограммируемыми *Flash*-микропроцессорами фирма *Atmel* выпускает также так называемые однократно программируемые *Flash*-микропроцессоры (без возможности стирания *Flash*-памяти) — *OTP (One-Time Programmable)*, микроконтроллеры с масочным *ПЗУ (ROM)*, запрограммированным на этапе производства и микроконтроллеры без встроенной памяти программ (*ROMless*), обеспечивая тем самым все потребности рынка.

Рассмотрим современное состояние семейства x51 микроконтроллеров фирмы *Atmel*.

Первая группа микроконтроллеров — перепрограммируемые микроконтроллеры с *Flash*-памятью программ в стандартном корпусе (см. табл. 4.1) практически не изменилась, за исключением того, что в нее включены микроконтролле-

ры *AT89C55WD* и *AT89C51RC*, содержащие 20 К и 32 К *Flash*-памяти программ, аппаратно реализованный *WDT*, оснащенные двумя *DPTR* и работающие на частотах до 33 МГц. По оценкам экспертов, наибольшей популярностью в этой группе пользуется стандартный *AT89C52*-микроконтроллер.

Таблица 4.1

Тип	ИROM, К	ИRAM, б	EM	I/O	SERIAL PORT	T/C	IS	SPI	WDT	DPTR	Vcc, V	Icc, mA	Fmax, МГц	PACKAGE
AT89C51	4	128	+	32	+	2	6	-	-	1	4,0–6,0	20	24	DIP40, PLCC44, PQFP44
AT89LV51	4	128	+	32	+	2	6	-	-	1	2,7–6,0	20	12	DIP40, PLCC44, PQFP44
AT89C52	8	256	+	32	+	3	8	-	-	1	4,0–6,0	25	24	DIP40, PLCC44, PQFP44
AT89LV52	8	256	+	32	+	3	8	-	-	1	2,7–6,0	25	12	DIP40, PLCC44, PQFP44
AT89LV55	20	256	+	32	+	3	8	-	-	1	2,7–6,0	25	12	DIP40, PLCC44, PQFP44
AT89C55WD	20	256	+	32	+	3	8	-	+	2	4,0–5,5	25	33	DIP40, PLCC44, PQFP44
AT89C51RC	32	512	+	32	+	3	8	-	+	2	4,0–5,5	30	33	DIP40, PLCC44, PQFP44

Вторая группа микроконтроллеров в малогабаритном корпусе с усеченным набором портов (табл. 4.2) практически не изменилась. Наиболее популярен в этой группе микроконтроллер *AT89C2051*, обеспечивающий наилучшее соотношение цена/качество.

Таблица 4.2

Тип	ИROM, К	ИRAM, б	EM	I/O	SERIAL PORT	T/C	IS	SPI	WDT	FUNCTION	DPTR	Vcc, D	Icc, mA	Fmax, МГц	Корпуса
AT89C1051U	1	64	-	15	+	2	6	-	-	AC	1	2,7–6,0	15	24	DIP20, SOIC20
AT89C2051	2	128	-	15	+	2	6	-	-	AC	1	2,7–6,0	15	24	DIP20, SOIC20
AT89C4051	4	128	-	15	+	2	6	-	-	AC	1	2,7–6,0	15	24	DIP20, SOIC20

Третья группа микроконтроллеров, с внутрисистемным программированием (см. табл. 4.3), пожалуй, является наиболее интересной для широкого круга разработчиков. В эту группу входят и известные читателю микроконтроллеры первого поколения *AT89S53* и *AT89S8252* (и их аналоги с пониженным напряжением питания *AT89LS53* и *AT89LS8252*), и аналоги первой группы *AT89C51/52* с добавленной возможностью внутрисистемного программирования — *AT89(L)S51/52*, и ряд новых малоизвестных микроконтроллеров *T89C51RB2/RC2/RD2/IC2/AC2*. Главными отличиями этих новых микроконтроллеров (*T89C51xxx*) являются два *DPTR*, встроенный аппаратный *WDT*,

встроенный режим *ISP*, большой объем *Flash*-памяти программ, наличие у части из них дополнительной оперативной памяти данных *XRAM* (+1 Кбайт), встроенного последовательного порта с дополнительными возможностями. Кроме того, все они оснащены так называемым *X2*-режимом ускорения (машинный цикл содержит 6 тактов, а не 12, как у стандартного семейства *x51*). Это позволяет им на тактовой частоте 20 МГц работать так же быстро, как стандартные микроконтроллеры работали бы на частоте 40 МГц. В то же время они могут работать и в стандартном режиме (12 тактов/цикл) на частотах до 40 МГц. Микроконтроллеры *RC2* и *RD2* имеют дополнительную загрузочную память с программой загрузки *Boot*, что значительно упрощает программирование. Микроконтроллер *RD2* в увеличенном корпусе имеет дополнительный пятый порт, т. е. количество линий ввода/вывода — 48. Микросхема *IC2* имеет встроенный аппаратный интерфейс *I²C*, микросхема *AC2* — встроенный восьмиканальный десятиразрядный аналого-цифровой преобразователь — *ADC*. Все перечисленные новшества в совокупности со встроенной *Flash*-памятью программ делают эти микроконтроллеры незаменимыми для многих изделий.

Таблица 4.3

Тип	IROM, К	IRAM/ XRAM, б	IDROM, К	EM	I/O	SERIAL PORT	T/C	IS	+ SPI/ *ISP Only	WDT	FUNC- TION	DPTR	Vcc, В	Fmax, МГц	Корпуса
AT89S51	4	128	—	+	32	+	2	6	*	+	—	2	4,0– 5,5	33	DIP40, PLCC44, PQFP44
AT89S52	8	256	—	+	32	+	3	8	*	+	—	2	4,0– 5,5	33	DIP40, PLCC44, PQFP44
AT89LS52	8	256	—	+	32	+	3	8	*	+	—	2	2,7– 4,0	16	DIP40, PLCC44, PQFP44
AT89S53	12	256	—	+	32	+	3	9	+	+	—	2	4,0– 6,0	33	DIP40, PLCC44, PQFP44
AT89LS53	12	256	—	+	32	+	3	9	+	+	—	2	2,7– 6,0	12	DIP40, PLCC44, PQFP44
AT89S8252	8	256	2	+	32	+	3	9	+	+	—	2	4,0– 6,0	33	DIP40, PLCC44, PQFP44
AT89LS8252	8	256	2	+	32	+	3	9	+	+	—	2	2,7– 6,0	12	DIP40, PLCC44, PQFP44
AT89S4D12	4	256	128	—	5	—	—	—	+	—	—	2	3,3	15	SOIC28, PLCC32
T89C51RB2	16	256 / 1024	—	+	32	E	3	10	+	+	X2	2	4,5– 5,5 2,7– 3,6	40	DIP40, PLCC44, PQFP44

Продолжение табл. 4.3

Тип	IROM, К	IRAM/ XRAM, b	IDROM, К	EM	I/O	SERIAL PORT	T/C	IS	+ SPI/ *ISP Only	WDT	FUNCTION	DPTR	Vcc, В	Fmax, МГц	Корпуса
T89C51RC2	32	256 / 1024	—	+	32	E	3	10	+	+	X2, Bool	2	4,5–5,5 2,7–3,6	40	DIP40, PLCC44, PQFP44
T89C51RD2	64	256/1024	2	+	32	E	3	7	*	+	X2, Bool	2	4,5–5,5 3,0–5,5	40 33	DIP40, PLCC44, PQFP44
					48										PLCC68VP QFP64
T89C51IC2	32	256/1024	—	+	32	+	3	10	+	+	X2, I2C, Bool	2	4,5–5,5 2,7–3,6	40	PLCC44, VQFP44
T89C51AC2	32	256	2	+	34	E	3	14	*	+	X2, Bool, ADC8*10	2	4,5–5,5	40	PLCC44, VQFP44

В четвертую группу семейства входят однократно программируемые *Flash*-микроконтроллеры (табл. 4.4), имеющие меньшую стоимость, чем перепрограммируемые и более высокую надежность (за счет невозможности случайного изменения содержимого памяти программ). Следует обратить внимание читателя на кристаллы *TS87C51U2*, имеющие два независимых расширенных последовательных порта, и *TS87C52X2*, способные работать на частотах до 66 МГц.

Таблица 4.4

Тип	IROM, К	IRAM/ XRAM, b	IDROM, К	EM	I/O	SERIAL PORT	T/C	IS	+ SPI/ *ISP Only	WDT	FUNCTION	DPTR	Vcc, В	Fmax, МГц	Корпуса
AT87F51	4	128	—	+	32	+	2	6	—	—	—	1	4,0–6,0	24	DIP40, PLCC44, PQFP44
AT87LV51	4	128	—	+	32	+	2	6	—	—	—	1	2,7–5,5	16	DIP40, PLCC44, PQFP44
AT87F52	8	256	—	+	32	+	3	8	—	—	—	1	4,0–6,0	24	DIP40, PLCC44, PQFP44
AT87LV52	8	256	—	+	32	+	3	8	—	—	—	1	2,7–5,5	16	DIP40, PLCC44, PQFP44
AT87F55WD	20	256	—	+	32	+	3	8	—	+	—	2	4,0–5,5	33	DIP40, PLCC44, PQFP44
AT87LV55	20	256	—	+	32	+	3	8	—	+	—	2	2,7–5,5	16	DIP40, PLCC44, PQFP44
AT87F51RC	32	512	—	+	32	+	3	8	—	+	—	2	4,0–6,0	24	DIP40, PLCC44, PQFP44

Продолжение табл. 4.4

Тип	ИРОМ, К	ИРАМ/ ХРАМ, б	ИДРОМ, К	EM	I/O	SERIAL PORT	T/C	IS	+ SPI/ * ISP Only	WDT	FUNCTION	DPTR	Vcc, В	Fmax, МГц	Корпуса
TS87C51RB2	16	256/ 1024	—	+	32	E	3	10	+	+	X2	2	4,5–5,5 2,7–3,6	40	DIP40, PLCC44, PQFP44
TS87C51RC2	32	256/ 1024	—	+	32	E	3	10	+	+	X2, Boot	2	4,5–5,5 2,7–3,6	40	DIP40, PLCC44, PQFP44
TS87C51RD2	64	256/ 1024	2	+	32	E	3	7	*	+	X2, Boot	2	4,5–5,5 3,0–5,5	40	DIP40, PLCC44, PQFP44
					48									33	PLCC68 VPQFP64
TS87C51U2	16	256	—	+	32	2E	3	7	—	+	X2	2	4,5–5,5 2,7–3,6	2x30	DIP40, PLCC44, VQFP44, CQFP44, CDIL40
T87C5112	8	256	—	+	40	E	2	8	—	+	ADC 8*10	1	4,0–5,5	33	PLCC52, LQFP48
TS87C52X2	8	256	—	+	32	E	3	6	—	—	X2	2	4,5–5,5 2,7–5,5	2x30	DIL40, PLCC44, VQFP44
TS87C54X2	16	256			32	E	3	6	—	+	X2	2	4,5–5,5 2,7–5,5	40 30	PDIL40, PLCC44, VQFP44, CQFP44, CDIL40
TS87C58X2	32	256			32	E	3	6	—	+	X2	2		40 30	PDIL40, PLCC44, VQFP44, CQFP44, CDIL40

Пятая группа микроконтроллеров (см. табл. 4.5) представлена двумя представителями. Первый из которых — T87C5101 способен работать на частотах до 66 МГц. Второй — T87C5111 имеет встроенный восьмиканальный десятиразрядный ADC в корпусе SSOP16.

Таблица 4.5

Тип	ОТР, К	ИРАМ/ ХРАМ, б	EM	I/O	SERIAL PORT	T/C	IS	SPI	WDT	FUNCTION	DPTR	Vcc, ВV	Fmax, МГц	Корпуса
T87C5101	16	256/256	16K	18	E	3	6	—	—	—	2	4,5–5,5	66	DIL24, S024, TSSOP24
				22										S028
T87C5111	4	256		16–22	E	2	8	+	+	X2, ADC 8*10	1	2,7–5,5	33	SSOP16, S024, DIL24

Шестая и седьмая группы микроконтроллеров (см. табл. 4.6 и 4.7) представлена уже знакомыми читателю кристаллами, снабжаемыми масочными ПЗУ программ.

Таблица 4.6

Тип	ROM, К	IRAM, b	IDROM, К	EM	I/O	SERIAL PORT	T/C	IS	SPI	WDT	FUNCTION	DPTR	Vcc, В	Fmax, МГц	Корпуса
TS83C51RB2	16	256/1024	-	+	32	E	3	10	+	+	X2	2	4,5-5,5 2,7-3,6	40	DIP40, PLCC44, PQFP44
TS83C51RC2	32	256/1024	-	+	32	E	3	10	+	+	X2, Bool	2	4,5-5,5 2,7-3,6	40	DIP40, PLCC44, PQFP44
TS83C51RD2	64	256/1024	2	+	32	E	3	7	*	+	X2, Bool	2	4,5-5,5 3,0-5,5	40 33	DIP40, PLCC44, PQFP44
TS83C51U2	16	256	-		32	2E	3	7	-	+	X2	2	4,5-5,5 2,7-3,6	2x30	DIP40, PLCC44, VQFP44, CQFP44, CDIL40
T83C5112	8	256	-	+	40	E	2	8	-	+	ADC 8*10	1	4,0-5,5	33	PLCC52, LQFP48
TS80C52X2	8	256	-	+	32	E	3	6	-	-	X2	2	4,5-5,5 2,7-5,5	2x30	DIL40, PLCC44, VQFP44
TS80C54X2	16	256			32	E	3	6	-	+	X2	2	4,5-5,5 2,7-5,5	40 30	PDIL40, PLCC44, VQFP44, CQFP44, CDIL40
TS80C58X2	32	256			32	E	3	6	-	+	X2	2		40 30	PDIL40, PLCC44, VQFP44, CQFP44, CDIL40

Таблица 4.7

Тип	ROM, К	IRAM/ XRAM, b	EM	I/O	SERIAL PORT	T/C	IS	SPI	WDT	FUNCTION	DPTR	Vcc, В	Fmax, МГц	Корпуса
T83C5101	16	256/256	16K	18/22	E	3	6	-	-	-	2	4,5-5,5	66	SO24, TSSOP24
T83C5102	8	256/256	8K	18/22	E	3	6	-	-	-	2	4,5-5,5	66	DIL24, SO24, TSSOP24
T83C5111	4	256		16/22	E	2	8	+	+	X2, ADC 8*10	1	2,7-5,5	33	SSOP16, SO24, DIL24

Кроме того, фирма *Atmel* выпускает ряд вышеописанных микроконтроллеров вообще без встроенной памяти программ (табл. 4.8).

Таблица 4.8

Тип	ROM, К	IRAM/ XRAM, b	EM	I/O	SERIAL PORT	T/C	IS	SPI	WDT	FUNCTION	DPTR	Vcc, В	Fmax, МГц	Корпуса
TS80C51RA2	256/256	-	+	32	E	3	7	-	+	X2	2	4,5-5,5 2,7-5,5	2x30	PDIL40, PLCC44, VQFP44
TS80C51RD2	256/1024	2	+	32	E	3	7	-	+	X2, Bool	2	4,5-5,5 3,0-5,5	40 33	DIP40, PLCC44, PQFP44
T80C5112	256	-	+	40	E	2	8	-	+	ADC 8*10	1	4,0-5,5	33	PLCC52, LQFP48

Продолжение табл. 4.8

Тип	ROM, К	IRAM/ XRAM, б	EM	I/O	SERIAL PORT	T/C	IS	SPI	WDT	FUNCTION	DPTR	Vcc, В	Fmax, МГц	Корпуса
TS80C51U2	256	—	+	32	2E	3	7	—	+	X2	2	4,5–5,5 2,7–3,6	2x30	DIP40, PLCC44, VQFP44, CQFP44, CDIL40
TS80C31X2	128	—	+	32	E	2	5	—	—	X2	2	4,5–5,5	2,7–5, 5	2x30
TS80C32X2	256	—	+	32	E	3	6	—	—	X2	2	4,5–5,5 2,7–5,5	2x30	DIL40, PLCC44, VQFP44

Девятая группа микроконтроллеров включает так называемые микроконтроллеры семейства C251 (табл. 4.9). По сравнению с обычными x51-контроллерами, C251 имеют расширенное адресное пространство внешней памяти до 256 К, 8/16 битную архитектуру, увеличенное IRAM — 1 К. Периферия микроконтроллеров дополнена SSLC-интерфейсом (*Synchronous Serial Link Controller*), обеспечивающим работу по I²C-, MicroWire- и SPI-протоколам, аппаратную поддержку реализации клавиатуры и специальный генератор скорости для последовательного канала. Еще раз подчеркнем, что эти микроконтроллеры совместимы с x51 как по разводке выводов, так и на уровне программного обеспечения (*Software*). Микроконтроллеры C251 имеют 16- или 32-битное расширенное ALU (арифметико-логическое устройство), в систему прерываний добавлено немаскируемое прерывание NMI (*Non Maskable Interrupt*), а также вход готовности WAIT (*Real-Time Wait States Input*). Эти микроконтроллеры применяются в высокоскоростных модемах, ISDN-терминалах, DVD ROM и плеерах, принтерах, сканерах и другой технике.

Таблица 4.9

Тип	IROM, К	IRAM, б	EM	I/O	SERIAL PORT	T/C	SPI	WDT	Vcc, В / Icc, мА / Fmax, МГц	Корпуса
TSC80251G2D	—	1K	256K	32	+	3	+	+	2,7–5,5В / 11мА / 16МГц 4,5–5,5В / 35мА / 24МГц	DIP40, PLCC44, PQFP44
TSC83251G2D	32	1K	256K	32	+	3	+			
TSC87251G2D	32	1K	256K	32	+	3	+			

Одна из последних разработок фирмы Atmel — микроконтроллеры, оснащенные модным сегодня аппаратным CAN-интерфейсом (версия 2.0A, 2.0B), также ожидает своих потребителей и поклонников (см. табл. 4.10).

Таблица 4.10

Тип	IROM, К	IRAM/ XRAM, б	IDROM, К	EM	I/O	SERIAL PORT	T/C	IS	SPI	WDT	DPTR	Vcc, В	Fmax, МГц	Корпуса
T89C51CC01	32	256/1K	2	+	34	+	3	14		E	2	5/3	40	PLCC44, VQFP44, CA-BGA64
T89C51CC02	16	256/256	2	+	16/20	+	3	14		E	2	5/3	40	PLCC28, SOIC28

Не менее интересна и другая новинка (табл. 4.11) — микроконтроллеры семейства x51 со встроенным *MP3* считывателем-декодером с поддержкой частот от 16 до 48 кГц. Микросхема обеспечивает отдельный цифровой амплитудный контроль правого и левого каналов, 31-ступенчатое управление амплитудно-частотной характеристикой воспроизведения на низких, средних и высоких частотах, некоторые музыкальные эффекты, индикаторы. Кроме музыкальных узлов, микросхема имеет большой набор интерфейсов *USB*, *MultiMediaCard*, *SPI*, *IDE/ATAPI*, двухканальный десятиразрядный *ADC*, стандартный последовательный порт с выделенным генератором скорости, *2Wire*-контроллер, *ISP*.

Таблица 4.11

Тип	ROM, К	IRAM, b	EM	I/O	SERIAL PORT	T/C	IS	SPI	WDT	Vcc, В	Fmax, МГц	Корпус
AT83C51SND1A	64 (ROM)	2304	+	32	+	2	8	+	+	5/3	40	PLCC84, TQFP80
AT89C51SND1A	64 (Boot Flash)											

Еще одна новинка — специализированная двенадцатая группа микроконтроллеров со встроенным декодером *Smart Card*. Ожидается, что эти микроконтроллеры найдут широкое применение в ряде кассовых аппаратов, весов, систем охранной сигнализации.

Таблица 4.12

Тип	Code Memory, К	XRAM, b	EM	I/O	SERIAL PORT	T/C	IS	SPI	WDT	DPTR	Vcc, В	Fmax, МГц	Корпус
T83C5121	16 (ROM)	256	+	14	E	2			+		2,85–5,4	16	PLCC52, SSOP24
T85C5121	16 (RAM)												
T89C5121	16 (Flash) + 16 (RAM)												

Описанные выше микроконтроллеры второго поколения семейства x51 показывают современные тенденции развития и роста этого семейства. Даже беглого взгляда на приведенные данные достаточно для того, чтобы отметить следующие тенденции.

1. Практически все модели микроконтроллеров используют стандартное ядро i8051. Повышение производительности микроконтроллеров достигается в основном за счет увеличения тактовой частоты, которая у некоторых моделей достигает 40 МГц (TS87C51RB2, RC2 и т. д.) или 66 МГц (T87C5101, T83C5101, T83C5102). Лишь в некоторых моделях реализован ускоренный механизм исполнения инструкций за 6 тактов системной частоты (T89C51RB2, T89C51RC2 и т. д.), а не за 12 тактов, как у стандартной модели i8051. При этом такие модели работают на частотах до 40 МГц. Таким образом, большинство моделей, работающих при частотах до 66 МГц, со стандартным (не ускоренным) ядром могут обеспечивать пиковую производительность до 5,5 MIPS. Модели, оснащенные механизмом ускоренного исполнения инструкций (за 6 тактов), на частоте 40 МГц способны достичь пиковой производительности 6,6 MIPS.

2. Объем оперативной памяти практически у всех моделей стандартный — 128—256 байт. Лишь несколько моделей имеют встроенную дополнительную оперативную память, составляющую 1 Кбайт (Т89С51RB2/RC2/RD2 и т. д.).

3. Основная цифровая периферия в основном стандартная.

4. Аналоговая периферия присутствует только в нескольких моделях (Т89С51АС2, Т87С5111, Т83С5111 и т. д.).

5. Практически все модели функционируют при стандартном напряжении питания +5 В.

6. Все основные модели выпускаются в корпусах DIP40.

Перечисленные качества микроконтроллеров фирмы *Atmel* позволяют с успехом использовать их в современных универсальных технологических контроллерах, реализованных в соответствии с принципами, изложенными в главе 3.

4.1.2. Достижения фирмы *MAXIM* в области x51-совместимых микроконтроллеров

В предыдущем разделе мы отмечали, что фирма *Atmel*, как и многие другие фирмы, выпускающие x51-совместимые микроконтроллеры, не уделяет особого внимания оснащению своих микроконтроллеров мощной аналоговой подсистемой. Однако в последнее время появился ряд фирм, которые на базе x51-совместимого ядра создали достаточно мощные аналого-цифровые микросистемы сбора данных.

Познакомимся с одним из таких семейств микросистем сбора данных фирмы *MAXIM* [74, 118, 119]. Семейство содержит два микроконтроллера — *MAX7651* и *MAX7652*, основные параметры которых приведены в табл. 4.13.

Таблица 4.13

Параметр	MAX7651	MAX7652
Встроенная Flash-память программ	2 × 8 К	2 × 8 К
Встроенная оперативная память данных IRAM, byte	256	256
Встроенный 12-битный аналого-цифровой преобразователь ADC (до 50 ksp/s)	√	√
Встроенный входной мультиплексор для ADC с количеством входов: однополярных дифференциальных	8 4	8 4
Широтно-импульсные 8-битные преобразователи	2	2
Источник опорного напряжения, Vref	внешний	внешний
Диапазон входных напряжений	+Vref	+Vref/2
Расширенный контроллер прерываний с количеством векторов	10	10
Таймеры общего назначения	3	3
Программный указатель данных DPTR	2	2
Программируемый охранный таймер WDT	√	√
Количество однобайтных портов ввода/вывода	4	4

Продолжение табл. 4.13

Параметр	MAX7651	MAX7652
Последовательный интерфейс передачи данных UART (до 375kb)	2	2
Максимальная тактовая частота, МГц	12	12
Напряжение питания, В	4,5...5,5	2,7...5,5
Рабочий ток потребления, мА	18	10
Потребляемая мощность, мВт	90	30
Диапазон рабочих температур, °С	-40...+85	0...+70
Тип корпуса	TQFP64	TQFP64

Микросистемы *MAX7651/2* оснащены усовершенствованным микроконтроллерным ядром, которое на уровне команд (кодов) полностью совместимо со стандартным *i8052*-ядром. Однако внутреннее строение ядра модифицировано, что позволило выполнять многие инструкции (команды) за 4 такта, а не за 12, как у стандартного *i8052*. Иными словами, если стандартное ядро при тактовой частоте 12 МГц имеет пиковую (максимально возможную) производительность примерно 1 MIPS (Million Instructions per Second) — миллионов инструкций в секунду, то ядро микроконтроллеров *MAX7651/2* будет иметь пиковую производительность примерно 3 MIPS.

Микросистемы *MAX7651/2* содержат несколько расширенный набор цифровой периферии, в состав которой входят дополнительный второй последовательный порт *UART* и охранный таймер *WDT*. Остальная цифровая периферия практически идентична стандартной. Микроконтроллеры имеют два режима энергосбережения.

К особенностям аналоговой части следует отнести то, что входной мультиплексор имеет два режима работы. Первый режим обеспечивает 4 пары полностью дифференциальных входов, а второй — 8 сигнальных входов *относительно плавающего общего входа*.

Подсистема памяти состоит из двух блоков *Flash*-памяти программ по 8 К каждый, расположенных по адресам от *0x0000* до *0x1FFF* и от *0x2000* до *0x3FBF*. Часть *Flash*-памяти верхнего блока от *0x3FC0* до *0x3FFF* зарезервирована и не доступна микроконтроллерному ядру. Адресное пространство от *0x4000* до *0xFFFF* доступно через внешний интерфейс для внешней памяти программ, если вход *EA/* = 0. Если *EA/* = 1, все внешнее адресное пространство от *0x0000* до *0xFFFF* доступно для внешней памяти программ. Программирование *Flash*-памяти программ осуществляется в параллельном режиме аналогично программированию *Flash*-памяти микроконтроллеров фирмы *Atmel*.

Как указано в таблице, микроконтроллеры выпускаются в корпусе *TQFP64*. Соответственно, нельзя говорить о совместимости выводов со стандартным *i8051*-микроконтроллером. Назначение выводов микроконтроллеров семейства *MAX7651/2* приведено в табл. 4.14.

Таблица 4.14

Вывод	Имя	Функция
1	AIN0	Аналоговый вход 0. Отрицательный дифференциальный вход относительно AIN1 или положительный дифференциальный вход относительно ACOM
2	AIN1	Аналоговый вход 1. Положительный дифференциальный вход относительно AIN0 или положительный дифференциальный вход относительно ACOM
3	AIN2	Аналоговый вход 2. Отрицательный дифференциальный вход относительно AIN3 или положительный дифференциальный вход относительно ACOM
4	AIN3	Аналоговый вход 3. Положительный дифференциальный вход относительно AIN2 или отрицательный дифференциальный вход относительно ACOM
5	AIN4	Аналоговый вход 4. Отрицательный дифференциальный вход относительно AIN5 или положительный дифференциальный вход относительно ACOM
6	AIN5	Аналоговый вход 5. Положительный дифференциальный вход относительно AIN4 или положительный дифференциальный вход относительно ACOM
7	AIN6	Аналоговый вход 6. Отрицательный дифференциальный вход относительно AIN7 или положительный дифференциальный вход относительно ACOM
8	AIN7	Аналоговый вход 7. Положительный дифференциальный вход относительно AIN6 или положительный дифференциальный вход относительно ACOM
9	AVDD	Положительное аналоговое напряжение питания для ADC и PWM. Необходимы параллельно соединенные конденсаторы 0,1 мкФ // 10 мкФ к аналоговой земле AGND
10	AGND	Аналоговая земля. Необходимо соединение PWMG и AGND
11	REF+	Положительный вход опорного напряжения для ADC и PWM (должно быть между AVDD и AGND). Необходимы параллельно соединенные конденсаторы 0,1 мкФ // 10 мкФ к аналоговой земле AGND
12	REF-	Положительный вход опорного напряжения для ADC и PWM (должно быть между AVDD и AGND). Необходимы параллельно соединенные конденсаторы 0,1 мкФ // 10 мкФ к аналоговой земле AGND
13	PWMV	Положительное аналоговое напряжение 2
14	PWMG	Аналоговая земля для PWM. Необходимо соединение PWMG и AGND
15	PWMA	PWM Выход А
16	PWMB	PWM Выход В
17	INT0	Вход внешнего прерывания 0 (активный низкий уровень)
18	INT1	Вход внешнего прерывания 1 (активный низкий уровень)
19	P3.7 / RD	P3.7: Бит 7 порта общего назначения P3 RD: Выход Read. Строб чтения внешних устройств (памяти или периферии) (активный низкий)
20	P3.6 / WR	P3.6: Бит 6 порта общего назначения P3 WR: Выход Write. Строб записи внешних устройств (памяти или периферии) (активный низкий)
21	P3.5 / T1	P3.5: Бит 5 порта общего назначения P3. T1: Внешний вход таймера 1
22	P3.4 / T0	P3.4: Бит 4 порта общего назначения P3 T0: Внешний вход таймера 0.
23	P3.3	P3.3: Бит 3 порта общего назначения P3
24	P3.2	P3.2: Бит 2 порта общего назначения P3

Продолжение табл. 4.14

Вывод	Имя	Функция
25	P3.1 / TxD0	P3.1: Бит 1 порта общего назначения P3 TXD0: Выход передатчика первого последовательного порта UART0
26	P3.0 / RxD0	P3.0: Бит 0 порта общего назначения P3 RXD0: Вход приемника первого последовательного порта UART0
27	DGND	Цифровая земля. Соединение DGND и AGND должно быть в источнике питания Соединить с выводами 27, 39 и 61
28	DVDD	Положительное цифровое напряжение питания. Необходимы параллельно соединенные конденсаторы 0,1 мкФ // 10 мкФ к земле DGND. Соединить с выводами 28, 40 и 62
29	P2.0 / A8	P2.0: Бит 0 порта общего назначения P2 A8: Бит 8 адреса внутренней Flash-памяти
30	P2.1 / A9	P2.1: Бит 1 порта общего назначения P2 A9: Бит 9 адреса внутренней Flash-памяти
32	P2.2 / A10	P2.2: Бит 2 порта общего назначения P2 A10: Бит 10 адреса внутренней Flash-памяти
32	P2.3 / A11	P2.3: Бит 3 порта общего назначения P2 A11: Бит 11 адреса внутренней Flash-памяти
33	P2.4 / A12	P2.4: Бит 4 порта общего назначения P2 A12: Бит 12 адреса внутренней Flash-памяти
34	P2.5	P2.5: Бит 5 порта общего назначения P2 Выбор младшего или старшего блока Flash-памяти
35	P2.6	P2.6: Бит 6 порта общего назначения P2 Выбор режима программирования
36	P2.7	P2.7: Бит 7 порта общего назначения P2 Выбор режима программирования
37	PSEN	Program Store Enable – строб внешней памяти программ (активный низкий)
38	ALE/PROG	ALE: строб защелки мультиплексированного адреса PROG: Импульс программирования
39	DGND	Цифровая земля
40	DVDD	Положительное цифровое напряжение питания
41	P0.0/AD0	P0.0: Бит 0 порта общего назначения P0 AD0: Бит 0 внутренней или внешней Flash-памяти
42	P0.1/AD1	P0.1: Бит 0 порта общего назначения P0 AD1: Бит 1 внутренней или внешней Flash-памяти
43	P0.2/AD2	P0.2: Бит 2 порта общего назначения P0 AD2: Бит 2 внутренней или внешней Flash-памяти
44	P0.3/AD3	P0.3: Бит 3 порта общего назначения P0 AD3: Бит 3 внутренней или внешней Flash-памяти
45	P0.4/AD4	P0.4: Бит 4 порта общего назначения P0 AD4: Бит 4 внутренней или внешней Flash-памяти
46	P0.5/AD5	P0.5: Бит 5 порта общего назначения P0 AD5: Бит 5 внутренней или внешней Flash-памяти

Продолжение табл. 4.14

Вывод	Имя	Функция
47	P0.6/AD6	P0.6: Бит 6 порта общего назначения P0 AD6: Бит 6 внутренней или внешней Flash-памяти
48	P0.7/AD7	P0.7: Бит 7 порта общего назначения P0 AD7: Бит 7 внутренней или внешней Flash-памяти
49	P1.0/T2/ T20UT/ AD0	P1.0: Бит 0 порта общего назначения P1 T2: Внешний вход прерывания 2 T20UT: Выход таймера 2 AD0: Бит 0 адреса внутренней Flash-памяти
50	P1.1/ T2EX/ AD1	P1.1: Бит 1 порта общего назначения P1 T2EX: Внешний вход триггера захвата таймера 2 AD1: Бит 1 адреса внутренней Flash-памяти
51	P1.2/ RXD1/ AD2	P1.2: Бит 2 порта общего назначения P1 RXD1: Вход приемника второго последовательного порта UART1 AD2: Бит 2 адреса внутренней Flash-памяти
52	P1.3/ TXD1/ ADS	P1.3: Бит 3 порта общего назначения P1 TXD1: Выход передатчика второго последовательного порта UART1 ADS: Бит 3 адреса внутренней Flash-памяти
53	P1.4/ AD4	P1.4: Бит 4 порта общего назначения P1 AD4: Бит 4 адреса внутренней Flash-памяти
54	P1.5/ AD5	P1.5: Бит 5 порта общего назначения P1 AD5: Бит 5 адреса внутренней Flash-памяти
55	P1.6/ AD6	P1.6: Бит 6 порта общего назначения P1 AD6: Бит 6 адреса внутренней Flash-памяти
56	P1.7/ AD7	P1.7: Бит 7 порта общего назначения P1 AD7: Бит 7 адреса внутренней Flash-памяти
57	EA/Vpp	EA: – выбор режима памяти программ. При использовании внешней ROM должен быть соединен с землей. При использовании внутренней Flash-памяти должен быть соединен с питанием. Vpp: Напряжение программирования.
58	RST	Вход сброса
59	XTAL2	Вывод для подключения кварцевого резонатора (выход)
60	XTAL1	Вывод для подключения кварцевого резонатора (вход)
61	DGND	Цифровая земля
62	DVDD	Положительное цифровое напряжение питания
63	TEST	Тестовый вывод, должен быть соединен с DGND
64	ACOM	Общий аналоговый вход мультимплексора

Обмен данными с периферийными устройствами и управление ими осуществляется через *SFR*-регистры, причем формат регистров стандартной периферии соответствует стандартному формату *SFR*-регистров стандартного микроконтроллера *i8052*.

Приведенные сведения дают лишь общее представление о внутренней структуре, разводке выводов и *SFR* (*Special Function Registers*) регистрах новых микросистемах сбора данных *MAX7651/2*. Более подробные сведения приведены в файле *MAX7651-MAX7652.pdf*, доступном на сайте [119], а также в руководстве пользователя, также имеющемся на вышеуказанном сайте.

4.1.3. Достижения фирмы *Cygnal* в области x51-совместимых микроконтроллеров

Если попросить разработчика микроконтроллерной техники перечислить основные недостатки существующих x51-совместимых микроконтроллеров, например фирмы *Atmel*, он наверняка ответит следующее:

- относительно невысокая пиковая (максимально возможная) производительность (при выполнении коротких операций), что обуславливалось особенностями архитектуры. Известно, что стандартное ядро микроконтроллеров 8051 выполняло короткие команды за 12 периодов тактовой частоты, т. е. при обычно используемой тактовой частоте — 11,0592 МГц пиковая производительность такого микроконтроллера соответствует примерно 1 MIPS (*Million Instructions per Second*) — миллиону операций в секунду. Некоторые микроконтроллеры семейства x51, например фирмы *Atmel*, были оснащены усовершенствованным ядром, а их тактовая частота была повышена до 40 МГц, однако при этом их пиковая производительность достигала всего 6,6 MIPS. На сегодняшний день это довольно мало;
- отсутствие до последнего времени в ряде микроконтроллеров встроенного монитора питания и охранного таймера *WDT*. Этот недостаток приводит к необходимости усложнения аппаратной «обвязки» контроллера;
- отсутствие в составе встроенных периферийных узлов большинства микроконтроллеров аналого-цифровых (*ADC*) и цифроаналоговых (*DAC*) преобразователей значительно усложняло схемную реализацию промышленных технологических контроллеров.

Все приведенные выше недостатки x51-совместимых микроконтроллеров удалось устранить молодой американской фирме *Cygnal Integrated Products, Inc* [64—66, 120] (далее просто *Cygnal*), которая с момента своего основания в 1999 сумела выпустить на рынок семь семейств (47 типов) новых x51-совместимых микроконтроллеров и анонсировала еще одно семейство с интерфейсом USB. Эти микроконтроллеры уже поступили на отечественный рынок [121]. Это воистину уникальные микроконтроллеры, а точнее — микроконтроллерные системы, имеющие мощное модернизированное ядро CIP-51, построенное по конвейерному принципу, которое более 70 % команд выполняет за один период тактовой частоты. Как следствие, их пиковая производительность достигает 25 MIPS (а отдельные представители имеют производительность 100 MIPS!). При этом система команд новых микроконтроллеров и карта адресов регистров специальных функций *SFR* полностью совместима со стандартным ядром *i8052*. Это позволяет использовать существующее программное обеспечение без вся-

ких ограничений. Новые микроконтроллеры имеют множество новых цифровых узлов и мощную аналоговую подсистему. Они оснащены достаточно большим объемом Flash-памяти программ, что позволяет все программное обеспечение писать на языках высокого уровня. Некоторые микроконтроллеры имеют дополнительную встроенную оперативную память. Все микроконтроллеры оснащены подсистемой внутрисхемного программирования и отладки программ в реальном времени — JTAG. Для каждого из семейств новых микроконтроллеров существуют эволюционные платы и оригинальное программное обеспечение для создания программ, программирования и отладки. Все вышесказанное позволяет с уверенностью сказать, что микроконтроллеры фирмы Cugnal на сегодняшний день являются самыми мощными в мире. Основные параметры выпускаемых микроконтроллеров приведены в таблице 4.15.

Таблица 4.15

ТИП	Номер семейства	Предельная частота, МГц	Цена, \$	Корпус	Flash память, К	Встроенная оперативная память (RAM)-УРАМ, байт	Линии ввода/вывода	Цифровая периферия	Аналоговая периферия			
									ADC0	ADC1	DAC	CPT
C8051F000	1	20	22.01	64	32	256	32	I2C+SPI+UART+WDT+4T+PCA/5	12/8G	—	12/2	2
C8051F001		20	21.13	48		256	16		12/8G	—	12/2	2
C8051F002		20	20.54	32		256	8		12/4G	—	12/2	1
C8051F005		25	25.23	64		256+2K	32		12/8G	—	12/2	2
C8051F006		25	24.35	48		256+2K	16		12/8G	—	12/2	2
C8051F007		25	24.03	32		256+2K	8		12/4G	—	12/2	1
C8051F010		20	19.07	64		256	32		10/8G	—	12/2	2
C8051F011		20	18.19	48		256	16		10/8G	—	12/2	2
C8051F012		20	17.31	32		256	8		10/4G	—	12/2	1
C8051F015		25	22.30	64		256+2K	32		10/8G	—	12/2	2
C8051F016		25	21.42	48		256+2K	16		10/8G	—	12/2	2
C8051F017		25	20.54	32		256+2K	8		10/4G	—	12/2	1
C8051F018	2	25	15.49	64	16	256+1K	32	I2C+SPI+2UART+WDT+5T+PCA/5	10/8	—	—	2
C8051F019		25	14.55	48		256+1K	16		10/8	—	—	2
C8051F020	3	25	29.17	100	64	256+4K	64	I2C+SPI+2UART+WDT+5T+PCA/5	12/8G	8/8G	12/2	2
C8051F021		25	27.64	64		256+4K	32		12/8G	8/8G	12/2	2
C8051F022		25	26.14	100		256+4K	64		10/8G	8/8G	12/2	2
C8051F023		25	24.71	64		256+4K	32		10/8G	8/8G	12/2	2
C8051F040	4	25	29.17	100	64	256+4K	64	I2C+SPI+2UART+WDT+5T+PCA/6+CAN	12/13G	8/8G	12/2	3
C8051F041		25	23.65	64		256+4K	32		12/13G	8/8G	12/2	3
C8051F042		25	26.18	100		256+4K	64		10/13G	8/8G	12/2	2
C8051F043		25	24.72	64		256+4K	32		10/13G	8/8G	12/2	2

Продолжение табл. 4.15

ТИП	Номер семейства	Предельная частота, МГц	Цена, \$	Корпус	Flash память, К	Встроенная оперативная память (RAM+XRAM, байт)	Линии ввода/вывода	Цифровая периферия	Аналоговая периферия				
									ADC0	ADC1	DAC	CPT	
C8051F206	5	25	15.20	48	8	256+1K	32	SPI+UART+WDT+3T	12/32	—	—	2	
C8051F220		25	9.24	48		256	32		8/32	—	—	2	
C8051F221		25	8.77	32		256	22		8/22	—	—	2	
C8051F226		25	11.00	48		256+1K	32		8/32	—	—	2	
C8051F230		25	7.78	48		256	32		—	—	—	2	
C8051F231		25	7.48	32		256	22		—	—	—	2	
C8051F236		25	9.54	48		256+1K	32		—	—	—	2	
C8051F300	6	25	8.48	11	4	256	8	SPI+UART+WDT+3T+PCA/3	8/8G	—	—	1	
C8051F300P		25	14.90	11		256	8		8/8G	—	—	1	
C8051F301		25	7.95	11		256	8		—	—	—	1	
C8051F302		25	7.01	11		256	8		8/8G	—	—	1	
C8051F303		25	6.48	11		256	8		—	—	—	1	
C8051F304		25	6.48	11		2	256		8	—	—	—	1
C8051F305		25	6.48	11		2	256		8	—	—	—	1
C8051F120	7	100	37.68	100	128	256+8K	64	I2C+SPI+2UART+WDT+5T+PCA/6	12/8G	8/8G	12/2	2	
C8051F121		100	36.51	64		256+8K	32		12/8G	8/8G	12/2	3	
C8051F122		100	35.28	100		256+8K	64		10/8G	8/8G	12/2	2	
C8051F123		100	32.99	64		256+8K	32		10/8G	8/8G	12/2	2	
C8051F124		50	31.45	100		256+8K	64		12/8G	8/8G	12/2	2	
C8051F125		50	30.00	64		256+8K	32		12/8G	8/8G	12/2	3	
C8051F126		50	28.88	100		256+8K	64		10/8G	8/8G	12/2	2	
C8051F127		50	27.44	64		256+8K	32		10/8G	8/8G	12/2	2	
C8051F310	8	25	?	32	16	256+1K	29	I2C+SPI+UART+WDT+4T+PCA/5	10/21G	—	—	2	
C8051F311		25	?	28		256+1K	25		10/17G	—	—	2	
C8051F060	9	25	?	100	64	256+4K	59	I2C+SPI+2UART+WDT+5T+PCA/6+CAN	16/2	10/8	12/2	3	
C8051F061		25	?	64		256+4K	24		16/2	10/8	12/2	3	
C8051F062		25	?	100		256+4K	59		16/2	10/8	12/2	3	
C8051F063		25	?	64		256+4K	24		16/2	10/8	12/2	3	
C8051F320	10	25	?	32	16	256+2K	24	I2C+SPI+UART+WDT+4T+PCA/5+USB	10/17	—	—	2	
C8051F321		25	?	28		256+2K	21		10/13	—	—	2	
C8051F330	11	25	?	20	8	256+512	17	I2C+SPI+UART+WDT+4T+PCA/3	10/16	—	10/1*	1	
C8051F331		25	?	20		256+512	17		—	—	—	1	

Микроконтроллеры условно разделены на 9 семейств (0хх, 018-9, 02х, 04х, 12х, 2хх, 30х, 31х, 33х), каждое из которых имеет общее описание в формате *pdf* [120]. Кроме того, для каждого микроконтроллера имеется краткий справочный листок (также в формате *pdf*), имя которого образуется добавлением к типу микроконтроллера записи «*short.pdf*» (например, для микроконтроллера *C8051F002* имя файла справочного листка — *C8051F002_short.pdf*).

Все микроконтроллеры имеют оригинальное ядро *CIP-51*, выполненное по конвейерному принципу. Ядро *CIP-51* имеет 109 инструкций, и по мнемонике, и по кодам полностью совместимое с инструкциями стандартного микроконтроллера *i8052*. Из 109 инструкций 26 инструкций выполняются за 1 период, 50 — за 2 периода, 26 — за 3 периода, 6 — за 4—5 периодов и лишь одна инструкция выполняется за 8 периодов тактовой частоты.

Еще одним важным достоинством микроконтроллерного ядра *CIP-51* является наличие встроенной аппаратной подсистемы отладки программного обеспечения. Связь с подсистемой микроконтроллера осуществляется через последовательный интерфейс *JTAG*, соответствующий *IEEE 1149.1*. При этом обеспечиваются как режим внутрисистемного программирования — *ISP*, так и собственно режим отладки. При программировании возможна запись как всего массива программы, так и модификация отдельных байтов. Естественно, что содержимое памяти программ может также читаться и сверяться с оригиналом. Содержимое любого байта *Flash*-памяти программ может читаться или изменяться из работающей программы с использованием инструкций *MOVC* или *MOVX*, что также позволяет осуществлять энергонезависимое хранение данных и оперативно их модифицировать под управлением программы.

Встроенная схема обеспечивает отладку в режиме реального времени выполнения программы. При этом возможны установка точек останова и контроля переменных, запуск, останов и пошаговое выполнение программы (включая подпрограммы прерываний), контроль обращений программы к стеку, контроль и модификация содержимого регистров и оперативной памяти. Используемый метод отладки ненавязчив, неагрессивен и не требует никаких ресурсов микроконтроллера (*RAM*, стека, таймеров и т. п.).

Для ядра *CIP-51* с подсистемой отладки фирмой *Cygnal* созданы программно-инструментальные средства отладки. Программное обеспечение выполнено в виде интегрированной среды развития — *IDE (Integrated Development Environment)*, включающей мощный редактор, макроассемблер, отладчик и программатор. Также имеется компилятор языка «C» фирмы Keil [122].

Важно отметить, что вся цифровая и аналоговая периферия микроконтроллера функционирует корректно в процессе отладки. В процессе останова микроконтроллера или в пошаговом режиме вся периферия (кроме *ADC*) также работает в соответствующем режиме.

К ядру внутренней шины подключается основная оперативная память *IRAM* объемом 256 байт, в некоторых моделях дополнительная оперативная память *XRAM* объемом от 1 К до 8 К, расположенная в адресном пространстве внешней памяти и *Flash*-память программ/данных объемом от 8 до 128 К. Для *Flash*-памяти существует развитая многоуровневая система защиты от чтения и

записи с программным управлением. В некоторых моделях микроконтроллеров, оснащенных интерфейсом *CAN*, имеется также дополнительная многоуровневая память для этого интерфейса.

К ядру также подключена развитая подсистема сброса с монитором питания и аппаратным охранным таймером *WDT*, который после сброса становится активным. Время срабатывания *WDT* устанавливается программно однократно после сброса (от 1—2 мс до 0,5 с). Имеется также развитая подсистема тактового генератора, программно устанавливаемая на частоты примерно 2, 4, 8, 16 МГц. Некоторые модели имеют еще режим 25 МГц. Внешний генератор может работать с кварцевым или керамическим резонатором, *RC*-цепочкой или отдельным конденсатором. Переключение между генераторами возможно «на лету», т. е. в ходе выполнения программы.

Весь микроконтроллер или его программно заданные части могут работать в двух режимах энергосбережения, при этом ток потребления снижается до единиц микроампер.

К ядру через специальную шину *SFR* подключается цифровая периферия.

Различные модели микроконтроллеров могут содержать от трех до пяти таймеров, причем первые три таймера полностью совместимы с таймерами стандартного *i8052*-микроконтроллера. Дополнительные 2 таймера используются дополнительной периферией.

Подсистема обработки прерываний значительно расширена и содержит 22 вектора прерываний, в отличие от 7 векторов стандартного *i8052*-микроконтроллера. Количество внешних входов прерывания расширено до 4.

Набор цифровой периферии также значительно расширен. Микроконтроллеры могут содержать один или два высокоскоростных последовательных порта *UART* (при частоте 11,059 МГц достигается скорость 115200), интерфейс *SMBus*, совместимый с *I2C*, интерфейс *SPI*. Все перечисленные интерфейсы допускают параллельные соединения для организации многоконтроллерной работы. Некоторые модели микроконтроллеров оснащены интерфейсом *CAN 2.0B*. Анонсировано также новое семейство с интерфейсом *USB*. Кроме этой периферии у многих микроконтроллеров имеется так называемый программируемый счетчик-массив *PCA*, представляющий собой 16-битный счетчик с пятью каналами захвата/сравнения. Этот узел позволяет генерировать и считать импульсы, работать как широко-импульсный модулятор, как дополнительный таймер или таймер реального времени *RTC*.

Различные модели микроконтроллеров могут иметь различное количество однобайтных портов ввода/вывода. Напомним, что стандартное количество портов — 4. Различные модели микроконтроллеров, в зависимости от типа корпуса, могут оснащаться 1, 2, 4, 8 портами! Возможности портов значительно расширены. На всех линиях имеется так называемая «слабая подтяжка», представляющая собой сопротивление примерно 100 кОм, соединенное с питанием. Эта «подтяжка» может включаться/отключаться программно. Кроме того, порты могут работать в обычном ключевом режиме или в режиме с открытым истоком, что используется для обеспечения совместимости в 5-вольтовых входах/выходах микросхем обвязки. Еще одной особенностью реализации портов являет-

ся то, что обеспечивается как втекающий, так и вытекающий мощный выходной ток, достигающий 25 мА.

В некоторых микроконтроллерах фиксированная часть линий портов мультиплексируется входами/выходами другой периферии. В других микроконтроллерах имеется коммутатор ресурсов *Crossbar*, позволяющий подключать к внешним линиям портов ввода/вывода только необходимую периферию и по разному определять расположение выводов дополнительной периферии. Некоторые мощные модели микроконтроллеров имеют дополнительный мультиплексированный интерфейс для подключения внешней памяти аналогично стандартному *i8051*. Этот интерфейс можно программно настроить на работу с мультиплексированной и немультимплексированной шиной.

Набор аналоговой периферии достаточно богат и может включать один или два аналого-цифровых преобразователя *ADC0* и *ADC1* с программируемым входным мультиплексором *AMUX* и входным усилителем *PGA* с программируемым коэффициентом усиления, обычно два двенадцатиразрядных цифроаналоговых преобразователя *DAC0* & *DAC1*, до трех компараторов, встроенный источник опорного напряжения 2,4 В.

Первый аналого-цифровой преобразователь *ADC0* 12-, 10- или 8-разрядный. В пяти семействах он вообще единственный. Вход первого *ADC0* обычно соединен с выходом программируемого усилителя *PGA*, который во многих семействах имеет коэффициент усиления 0,5, 1, 2, 4, 8 или 16. У некоторых моделей коэффициент усиления фиксирован и равен 1. Вход этого усилителя обычно соединен с выходом аналогового мультиплексора *AMUX*, который у различных моделей может иметь 8, 13, 16 или 32 входа. Мультиплексор управляется программно и обеспечивает либо работу в однополярном режиме, либо в дифференциальном режиме, при этом количество входных линий уменьшается в два раза. Кроме этого, практически все модели имеют встроенный датчик температуры, подключенный к дополнительному входу *AMUX*. Датчик температуры позволяет измерять температуру до 100 °С с точностью 3 °С.

В некоторых моделях имеется два *ADC*: *ADC0* — 12- или 10-разрядный, а *ADC1* — 8-разрядный. Второй *ADC1* имеет также свой программируемый усилитель *PGA1* с коэффициентами усиления 0,5, 1, 2 или 4 и свой аналоговый мультиплексор *AMUX1* на 8 входов.

Аналого-цифровые преобразователи имеют сервисный узел «окна». Имеются два 16-разрядных регистра *SFR*, в которые устанавливаются коды нижнего и верхнего порогов (границ) «окна». Узел может генерировать прерывания при обнаружении выходного кода *ADC* как внутри заданного окна, так и вне его. Запуск *ADC* может осуществляться программно, от таймера или от внешнего сигнала.

Многие модели микроконтроллеров имеют два 12-разрядных цифроаналоговых преобразователя с выходом по напряжению.

Аналого-цифровые и цифроаналоговые преобразователи могут работать от встроенного источника опорного напряжения 2,4 В или от внешнего источника, который при этом не может быть больше напряжения питания (тач 3,6 В).

Кроме этого, микроконтроллеры могут иметь от 3 до 1 компаратора (обычно 2). Компараторы имеют возможность независимой установки величины поло-

жительного и отрицательного гистерезиса. Первый компаратор может служить источником сигнала сброса, т. е. осуществлять мониторинг внешнего источника питания.

В заключение описания аналоговой подсистемы следует сказать, что некоторые из микроконтроллеров имеют индивидуальные выводы для входов и выходов аналоговых сигналов. У других моделей часть аналоговых сигналов мультиплексирована с выводами цифровых портов.

Все микроконтроллеры фирмы *Cygnal* работают при напряжении питания от 2,6 В до 3,6 В в промышленном диапазоне температур от -40°C до $+85^{\circ}\text{C}$. Микроконтроллеры фирмы *Cygnal* выпускаются в девяти типах корпусов: *TQFP100*, *TQFP64*, *TQFP48*, *LQFP32*, *MLP11*, *MLP20*, *MLP28*, *MLP32*, *DIP14*. В предыдущем предложении нет ошибки! Все эти корпуса предназначены только для пайки (за исключением одного микроконтроллера *C8051F300P* в корпусе *DIP14*). Габаритные размеры очень маленькие. Шаг между выводами составляет 0,5 мм для корпусов *TQFP* и 0,8 мм для корпусов *LQFP*. Габаритные размеры корпуса *MLP11* для микроконтроллеров *C8051F300-305* с 11 выводами составляют всего 3×3 мм. Это самый маленький в мире корпус микроконтроллера!

Вышеперечисленные достоинства микроконтроллеров фирмы *Cygnal* позволяют создавать достаточно простые *TMCS*, т. к. почти все необходимые узлы уже имеются на кристалле. Примеры таких технологических микроконтроллеров мы рассмотрим немного дальше.

А сейчас отметим те сложности, с которыми придется столкнуться разработчику при создании универсальных технологических контроллеров на базе микроконтроллеров фирмы *Cygnal*. Этих сложностей всего две.

Первая из них заключается в низковольтном питании микроконтроллеров. Эта проблема решается довольно просто добавлением одного линейного малошумящего стабилизатора, который преобразует основное напряжение питания системы +5 В в напряжение в диапазоне от 2,6 В до 3,6 В. При этом потребление собственно микроконтроллера, даже самого мощного, не превышает 25 мА. Необходимо только иметь в виду, что от этого же стабилизатора питается и адаптер-программатор *JTAG*. Таким образом, суммарное потребление может достигать 70—80 мА. Обычно для питания таких микроконтроллеров используются стабилизаторы с рабочим током от 150 до 500 мА.

Вторая сложность заключается в используемых малогабаритных корпусах, для которых не существует панелек. Это не позволяет оперативно менять микроконтроллер, что снижает ремонтпригодность системы. Выход из этой ситуации находится в определенном конструктивном исполнении, называемым «слотовым». Смысл этого исполнения заключается в том, что и микроконтроллер, и интерфейс *JTAG*, и низкочумящий стабилизатор размещаются на отдельной печатной плате, снабженной унифицированным (для данной системы) разъемом. Эта плата называется «слотом» и устанавливается в разъем на плате главного контроллера *TMCS*.

микроконтроллеры. Полученные данные необходимо учитывать при проектировании конкретных систем малой автоматизации.

Завершив этими сведениями ознакомление с наиболее используемыми микроконтроллерами, перейдем к рассмотрению некоторых микроконтроллерных систем.

4.2. Эволюционный контроллер

Описываемый вариант контроллера предназначен в первую очередь для макетирования и отладки различных периферийных аналоговых узлов, для этого на печатной плате изготовлено специальное макетное поле достаточно больших размеров. Кроме того, он может быть также применен для учебных целей — обучения схемотехнике и принципам работы таких важных для любых универсальных технологических контроллеров узлов, как интерфейсы *RS232C* и *RS485*, супервизор питания и охранный таймер *WDT*, таймер реального времени *RTC* и «интеллектуальный» знаковосинтезирующий жидкокристаллический индикатор *LCD*. Контроллер имеет соответствующий разъем для внутрисистемного программирования *ISP*. Все микросхемы контроллера установлены на панельках, все входы/выходы контроллера и важные сигналы других микросхем выведены на соответствующие штыревые разъемы *PLS* (однорядные), *PLD* (двухрядные) и контактные площадки, расположенные вокруг макетного поля. Еще одной важной особенностью описываемого эволюционного контроллера является то, что он может работать с целым рядом микроконтроллеров фирмы *Atmel* семейств *x51* и *AVR*: *AT89C51/52/55WD/51RC*, *AT89S51/52/53/8252*, *T89C51RB2/RC2/RD2*, *TS87C51RB2/RC2/RD2/U2*, *TS87C52X2/54X2/58X2*, *TSC80251G2D*, *AT89C1051U/2051/4051*, *AT90S2313*, *AT90S8515/8535*. Микроконтроллеры в корпусах *DIP20* устанавливаются на специальную панельку, расположенную внутри панельки *DIP40* для основного (стандартного) типа микроконтроллера. При работе с микроконтроллерами с усеченными портами ввода/вывода (*AT89C1051U/2051/4051* и *AT90S2313*) узлы *RTC* и *LCD* напрямую не подсоединены к микроконтроллеру. Но их можно подключить на свободные выводы через штыревые разъемы или контактные площадки.

Принципиальная схема эволюционного контроллера представлена на рис. 4.2. Он содержит следующие функциональные узлы:

- 1) интерфейс *RS232C* на микросхеме *MAX202/232* или *ADM202 (D1)*;
- 2) интерфейс *RS485* на микросхеме *MAX485/487* или *ADM485 (D2)*;
- 3) узел управления интерфейсами: *74HC32 (D4)* и *74HC03 (D5)*;
- 4) супервизор питания с охранным таймером *WDT* на микросхеме *TL7705 (D6)*;
- 5) собственно микроконтроллер (один из перечисленных выше типов) (*D7*);
- 6) таймер реального времени на микросхеме *DS12(C)887(A) (D8)*;
- 7) регистры вывода *74HC573 (D9, D10)*;
- 8) управляемый канал ввода/вывода на микросхеме *74HC245 (D11)*.

На плате также находятся штыревые разъемы и контактные площадки для подключения к макетируемым узлам (см. табл. 4.16).

Таблица 4.16

Элемент	Назначение
JP1	Штыревой разъем для подключения к интерфейсу RS232C
JP2	Штыревой разъем для подключения к интерфейсу RS485
JP3	Штыревой разъем для подключения напряжения питания +5 В
JP4	Штыревой разъем для перемычки включения согласующего резистора R3 120 Ом. Устанавливается только на последних станциях сети RS485
JP5	Штыревой разъем для выбора перемычкой источника дополнительного сброса от интерфейсов RS232C или RS485
JP6	Контактные площадки для подключения периферийных устройств ко второму каналу интерфейса RS232C (RxD2, TxD2)
JP7	Штыревой разъем для подключения кабеля программатора ISP
JP8	Штыревой разъем для выбора полярности сигнала сброса RST
JP9	Контактные площадки порта P1
JP10	Контактные площадки порта P3
JP11	Контактные площадки порта P0
JP12	Контактные площадки порта P2
JP13	Штыревой разъем для подключения LCD
JP14	Контактные площадки для подключения прерывания IRQ и генератора SQW таймера реального времени RTC (D8) к макетируемым узлам
JP15	Штыревой разъем для выбора перемычкой полярности напряжения контрастности LCD
JP16	Контактные площадки выходов регистра RG1 (D9)
JP17	Контактные площадки выходов регистра RG2 (D10)
JP18	Контактные площадки входов/выходов шинного формирователя BD (D11)

Узел интерфейса RS232C выполнен на микросхеме MAX232 (D1) в стандартном включении. Первые сигналы приема RxD1 (D1/11) и передачи TxD1 (D1/12) данных используются для обмена данными по интерфейсу RS232C. Вторые сигналы приема RxD2 (D1/09) и передачи TxD2 (D1/10) могут использоваться разработчиком для обмена служебными сигналами или других целей. Для подсоединения эти сигналы выведены на контактные площадки JP6. Кроме того, сигнал RxD2 может использоваться для сброса микроконтроллера D7 от подсоединенного персонального компьютера (PC), если входной контакт разъема JP1/10 соединен с линией DTR или RTS персонального компьютера. Для этого необходимо установить перемычку JP5 в положение 01—02. Контакты выходных разъемов (DB9 и DB25) интерфейса RS232C персонального компьютера

с международными названиями связей приведены в таблице 4.17. Следует напомнить, что информационные контакты кабеля связи персонального компьютера с эволюционной платой должны быть соединены следующим образом. Выходной контакт *TxD* разъема персонального компьютера (*DB9/3* или *DB25/2*) необходимо соединить с контактом *JP1/08* эволюционной платы, а входной контакт *RxD* разъема персонального компьютера (*DB9/2* *DB25/3*) — с контактом *JP1/02* эволюционной платы. Кроме того, общий провод на разъеме *PC* (*DB9/05* или *DB25/07*) должен быть соединен с контактами *JP1/01* (*03*, *05*, *07* или *09*).

Таблица 4.17

Сигнал	Имя цепи		Номер контакта		Назначение	Направление относительно РС
	ЕА	ССИТ	DB9	DB25		
DCD	CF	109	1	8	Connect	Вход
RxD	BB	104	2	3	Input Data	Вход
TxD	BA	103	3	2	Output Data	Выход
DTR	CD	108/2	4	20	Ready to Work	Выход
GND	AB	102	5	7	GND	Общий
DSR	CC	107	6	6	Modem Ready	Вход
RTS	CA	105	7	4	Transmit Ask	Выход
CTS	CB	106	8	5	Modem Transmit Ready	Вход
RI	CE	125	9	22	Ring	Вход
FG	AA	101	BOX	1	E	Экран

Узел интерфейса *RS485* может быть выполнен на взаимозаменяемых микросхемах *MAX485* (допускает параллельное соединение до 32 станций) или *MAX487* (допускает параллельное соединение до 128 станций). Узел имеет стандартную защиту от перенапряжений в линии (резисторы *R1*, *R2* 5,1 Ома и стабилитроны *VD1*, *VD2* *KC191A*). Кроме того, имеется согласующий резистор *R3* 120 Ом и перемычка *JP4*. Следует помнить, что при организации сети на базе интерфейса *RS485* согласующие резисторы должны устанавливаться только на крайних станциях (путем установки перемычки *JP4*). На остальных станциях перемычку *JP4* необходимо снять. Вывод 02 микросхемы *D2* соединен с общим проводом (землей). Это означает, что канал ввода интерфейса *RS485* всегда открыт. Канал вывода управляется уровнем напряжения на выводе *D2/03*.

Управление интерфейсами *RS232C* и *RS485* осуществляется четырьмя сигналами микроконтроллера *I232_OFF*, *I485_OFF*, *O232_OFF* и *O485_ON* с помощью узла управления интерфейсами, реализованного на микросхемах *74HC32* (*D4*) и *74HC03* (*D5*). Узел позволяет реализовать гибкое управление интерфейсами. Входы и выходы интерфейсов управляются независимо. Табл. 4.18 иллюстрирует состояния узла управления.

Таблица 4.18

I232_OFF	I485_OFF	O232_OFF	O485_ON	Состояние
0	0			Входы обоих интерфейсов открыты и объединены по функции логического «И»
0	1			Открыт только вход RS232C
1	0			Открыт только вход RS485
1	1			Входы обоих интерфейсов закрыты. Состояние после сброса
		0	0	Открыт только выход RS232C
		0	1	Выходы обоих интерфейсов открыты. Передача данных в обе сети
		1	0	Выходы обоих интерфейсов закрыты
		1	1	Открыт только выход RS485. Состояние после сброса

Узел супервизора питания с охранным таймером *WDT* реализован на микросхеме *TL7705 (D6)*. К входу *O2* микросхемы *D6 (TL7705)* подключен конденсатор *C9*, который подзаряжается через резистор *R9* во время импульса сброса и через цепочку *C8, R8, VD3* в рабочем режиме, при этом, если интервал подачи подзаряжающих импульсов с микроконтроллера *D7* (порт *P3.5*, вывод *15*) не превышает определенную временную величину (в данном случае около 20 мс), напряжение на конденсаторе продолжает оставаться выше уровня восприятия логической единицы (на выводе *2* микросхемы *D6*). В случае «зависания» микроконтроллера подача заряжающих импульсов прекращается и напряжение на конденсаторе *C9* снижается, как следствие, вырабатывается импульс сброса микроконтроллера. Таким образом, в рассматриваемой схеме аппаратно реализована функция *WDT*.

Кроме этого, как уже упоминалось выше, в описываемом эволюционном контроллере предусмотрена возможность внешнего сброса от одного из интерфейсов *RS232C* или *RS485*. Это достигается установкой соответствующей перемычки *JP5*. Если установлена перемычка в позиции *O1—O2*, то управление сбросом осуществляется потенциалом сигнала *RxD2*. На плате имеется также разрезное поле, позволяющее устанавливать в линию *RxD2* конденсатор емкостью ~0,1 (на схеме не показано), при этом возможно импульсное (а не потенциальное) управление сбросом. Если установлена перемычка в позиции *O3—O4*, то управление сбросом осуществляется импульсом сигнала, поступающего от разьема *JP2/06* через конденсатор *C15*, а если вместо конденсатора установлена перемычка, то управление сбросом может осуществляться потенциалом. Следует отметить, что при потенциальном управлении сбросом функции *WDT* не выполняются. Штыревой разъем *JP7* позволяет выбирать полярность сигнала сброса: положительную при положении перемычки *O3—O4* для *x51*-совместимых микроконтроллеров и отрицательную при положении перемычки *O5—O6* для *AVR*-микроконтроллеров. Положение перемычки *O1—O2* необходимо для осуществления функций внутрисхемного программирования *ISP* (при этом перемычки *O3—O4* и

05—06 должны быть удалены), а функции сброса возлагаются на *ISP*-программатор, который управляет состоянием линии через контакт 05 разъема *JP7*.

Собственно узел микроконтроллера имеет конструктивную особенность, заключающуюся в том, что с целью экономии места на печатной плате внутри панельки для стандартных типов микроконтроллеров в корпусе *DIP40* расположена панелька для микроконтроллеров с усеченными линиями ввода/вывода *AT89C1051U/2051/4051* и *AT90S2313* в корпусе *DIP20*. Соединение выводов этих панелек показано на рис. 4.3.

К порту *P0* стандартных контроллеров (в корпусе *DIP40*) подключены: микросхема *D8* таймера реального времени *RTC* — *DS12887(A)*, интерфейс *LCD* (разъем *JP13*), два регистра вывода на микросхемах *74HC573* (*D9*, *D10*) и управляемый канал ввода/вывода на микросхеме шинного формирователя *74HC245* (*D11*). Все перечисленные функциональные узлы управляются линиями порта *P2* микроконтроллера.

Выводы микросхемы (*D8*) таймера реального времени *RTC* — *IRQ* (прерывание от будильника) и *SQW* (программируемый генератор меандра) выведены

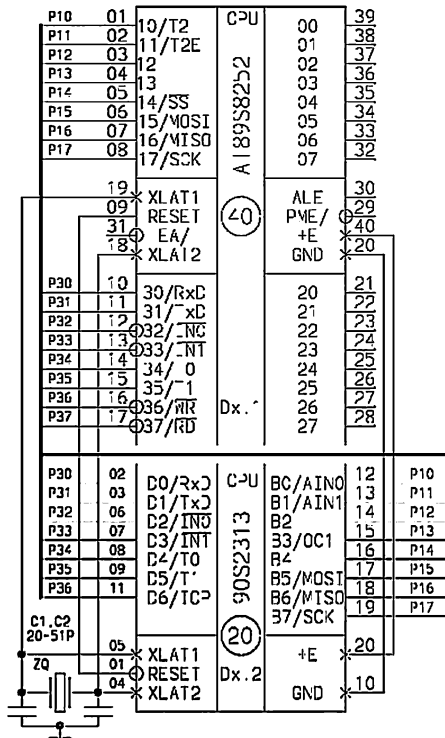


Рис. 4.3. Соединение выводов панелек микроконтроллеров

на контактные площадки *JP14*. Они могут быть подключены к исследуемым узлам или свободным входам микроконтроллера.

Штыревой разъем *JP15* и резистор *R14* служат для выбора полярности и установки контрастности *LCD*. При этом отрицательное напряжение -10 В используется от микросхемы *D1* (*MAX232*). Резистор *R13* служит для ограничения (регулирования) тока подсветки *LCD*. Разводка штыревого разъема *JP13* соответствует стандартной для *LCD* разводке, что позволяет использовать для связи с *LCD* стандартный 40-проводный плоский компьютерный кабель *HDD* (для подключения жестких дисков).

Регистры (*D9*, *D10*) работают только на вывод, их выходы всегда открыты. После сброса выходы устанавливаются в единичное состояние. Регистры предназначены для управления исследуемыми периферийными узлами.

Управляемый канал ввода/вывода на микросхеме шинного формирователя *74HC245* (*D11*) может работать как на ввод, так и на вывод данных, в зависимости от состояния линии *P2.6*. После сброса регистр устанавливается в положение на вывод и переводится в высокоимпедансное состояние. Выходы регистра подтянуты к $+5$ В через матрицу резисторов *R15*. При установке линий $P2.6 = 0$ и $P2.7 = 0$ регистр открывается и устанавливается на ввод, что позволяет использовать разъем *JP18* в качестве модификатора для установки, например, сетевого адреса контроллера или других параметров.

И, конечно же, эволюционный контроллер имеет достаточно большое макетное поле, на которое при желании можно установить по два корпуса *DIP40* и *DIP16* (или эквивалентное количество более мелких корпусов). Контактные площадки сгруппированы по три, что позволяет устанавливать корпуса с различной шириной (300, 450 и 600), а также при установке узких корпусов устанавливать в оставшиеся контактные отверстия необходимые дискретные элементы. Естественно, что все установочные отверстия металлизированные. Макетное поле окружено мощной земляной шиной, а по центру проведена шина питания $+5$ В.

Описанный эволюционный контроллер выполнен в виде печатной платы с размерами 100×100 мм. Область установки основных элементов покрыта зеленым защитным лаком. Имеются крепежные отверстия. Эволюционный контроллер позволяет макетировать аналоговые подсистемы ввода/вывода, а также некоторые специфические цифроаналоговые радиочастотные узлы.

4.3. Вариант универсального контроллера

Универсальные технологические контроллеры, ориентированные на работу в технологических системах управления, как правило, выполняются в виде магистрально-модульных систем. В таких системах все модули (платы) имеют одинаковые размеры и оснащаются системным разъемом. В последнее время наиболее популярным размером печатных плат в таких конструкциях является 220×100 мм (хотя иногда используется и размер 100×100 мм). В качестве разъемов магистрали наиболее часто используются отечественные 96-контакт-

ные разъемы *СНП96* (или импортные аналоги *С96* с дюймовым шагом). Ответные части разъемов соединяются между собой через кросс-плату. Все контакты разъемов, установленных на кросс-плате, соединены параллельно и образуют системную магистраль. Обычно используется модифицированная (усеченная) магистраль *И41* (отечественный аналог магистрали *Multibus-I*). Модули (платы) располагаются в специальной конструкции с направляющими, называемой корзиной. Корзины могут быть рассчитаны на различное число модулей, но обычно используются корзины на 10 модулей. Кросс-плата крепится на заднюю стенку корзины, а платы вставляются в корзину спереди по направляющим вертикально. На передних сторонах модулей закреплены передние панели одинакового размера, которые кроме органов управления имеют элементы фиксирования модулей в корзине и специальные ручки для извлечения плат. Все передние панели установленных модулей образуют общую «сегментированную» переднюю панель корзины. Корзина боковыми поверхностями крепится к двум легким (обычно из дюралюминия) боковым вертикальным стенкам с ручками для переноски. За корзиной обычно располагается блок питания. Под корзиной и блоком питания располагаются несколько мощных компьютерных вентиляторов, постоянно обдувающих вертикально стоящие (в корзине) модули и блок питания снизу вверх. Вся эта конструкция обычно называется крейтом.

Обычно система управления сложной технологической установки содержит один крейт и может содержать ряд других электронных узлов, например выносные блоки предварительных усилителей, располагающиеся непосредственно возле датчиков, или автономные мощные аналоговые узлы управления (например прецизионные терморегуляторы) и т. п.

В состав крейта обычно входит один модуль главного контроллера и несколько периферийных модулей.

В описываемом случае, например, в состав системы входят:

- несколько модулей периферийных регистров с мощными высоковольтными ключами, необходимыми для управления силовыми агрегатами системы;
- модуль аналоговых усилителей, необходимых для фильтрации и предварительного усиления аналоговых сигналов с датчиков перед подачей их на аналого-цифровой преобразователь;
- модуль мощных выходных нормирующих усилителей, необходимых для обеспечения требуемых уровней и мощности выходных сигналов цифроаналоговых преобразователей.

Эти модули выполнены традиционно и в данном разделе не рассматриваются.

Перейдем к рассмотрению собственно главного контроллера. Он содержит три подсистемы: подсистему аналогового ввода/вывода, подсистему интерфейсов и подсистему контроллера.

Подсистема аналогового ввода/вывода представлена на рис. 4.4. Она содержит всего три микросхемы: восьмиканального 12-разрядного аналого-цифрового преобразователя *D1 (MAX1270)* и двух микросхем восьмиканальных цифроаналоговых преобразователей 8-разрядного *D2 (LTC1665)* и 10-разрядного *D3 (LTC1660)*. Подсистема управляется микроконтроллером при помощи интерфейса *SPI*. Цифро-аналоговые преобразователи *D2* и *D3* используют опорное напря-

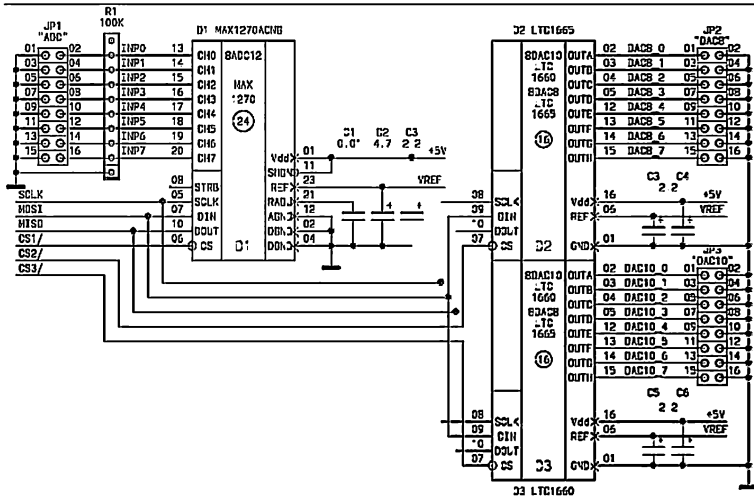


Рис. 4.4. Подсистема аналогового ввода/вывода универсального технологического контроллера

жение $VREF$, вырабатываемое аналого-цифровым преобразователем $D1$. Вся аналоговая подсистема питается от одного источника +5 В с дополнительной LC-фильтрацией (на схеме цепи питания не показаны). Все микросхемы выполнены в DIP корпусах и установлены на панельки. Входы аналого-цифрового преобразователя выведены на штыревой двухрядный разъем $JP1$ ($PLD-16\ 2 \times 8$) и соединены с общим проводом через резисторную сборку $R1$. Выходы цифроаналоговых преобразователей также выведены на штыревые двухрядные разъемы $JP2$ и $JP3$ ($PLD-16\ 2 \times 8$). Следует заметить, что обе микросхемы цифроаналоговых преобразователей имеют одинаковую разводку выводов, что позволяет при необходимости оснастить контроллер шестнадцать 8-разрядными или шестнадцать 10-разрядными цифроаналоговыми преобразователями.

Подсистема интерфейсов показана на рис. 4.5. Она содержит два интерфейса: $RS232C$ и оптически изолированный $RS485$, а также две микросхемы логики управления. В качестве интерфейса $RS232C$ использована микросхема $MAX202$ ($D7$), ее выводы выведены на штыревой разъем $JP6$. Интерфейс $RS485$ выполнен на микросхеме $MAX1480A$ ($D6$) со встроенной схемой развязки питания и оптической изоляцией сигналов. Интерфейс имеет цепи защиты от импульсных помех $R12$, $R13$, $VD1$, $VD2$, согласующий резистор $R11$ с перемычкой $JP4$. Сигналы интерфейса выведены на разъем $JP5$. Логика управления реализована на микросхемах $D4$, $D5$. Она позволяет осуществлять раздельное управление вводом и выводом, а также выбирать один или два активных интерфейса.

Подсистема контроллера представлена на рис. 4.6. Она содержит ряд узлов: супервизор питания с функциями охранного таймера WDT на микросхеме $D8$ ($TL7705$), входной регистр модификатора $D9$ ($74HC245$), дешифратор сигналов чтения/записи памяти и узлов ввода/вывода на микросхеме дешифратора $D10$

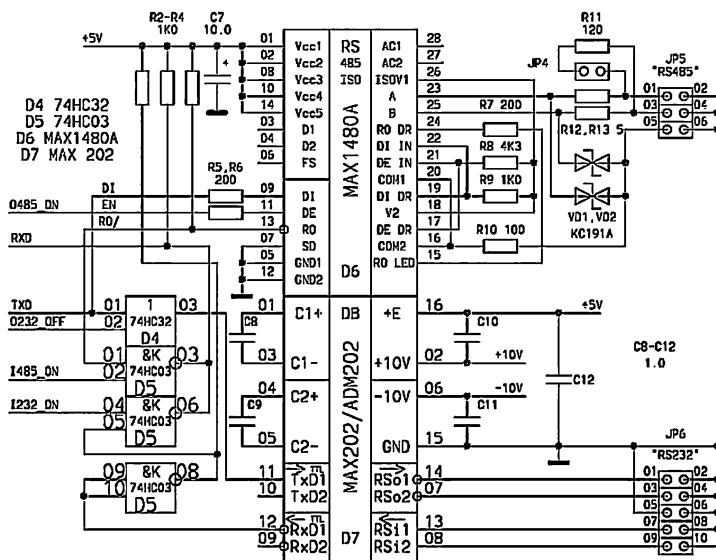


Рис. 4.5. Подсистема интерфейсов универсального технологического контроллера

(74HC138), собственно микроконтроллер D11, микросхему оперативной памяти D12 (61C256) с объемом 32K×8, микросхемы шинного формирователя D13 (74HC245) и регистра защелки младшего байта адреса D14 (74HC573), микросхему таймера реального времени со встроенной литиевой батарейкой и энерго-независимой оперативной памятью объемом 32K×8 — D15 (DS1644), логику выборки старшего адреса D16 (74HC00), выходной регистр управления D17 (74HC374), логику выбора адресов регистров (D9 и D17) на микросхемах D18 (74HC30) и D19 (74HC32), два выходных изолированных оптически сигнала на оптронах D20, D21 (4N35), двухцветный индикатор состояния на светодиоде VD6, модификатор сетевого адреса JP8, системный разъем XC1, разъем для подключения жидкокристаллического индикатора JP10, разъем JP9 и переменный резистор R23 для выбора и установки напряжения контрастности. В качестве микроконтроллера может использоваться ряд микроконтроллеров фирмы Atmel семейств x51 и AVR в стандартном корпусе DIP40: AT89C51/52/55WD/51RC, AT89S51/52/53/8252, T89C51RB2/RC2/RD2, TS87C51RB2/RC2/RD2/U2, TS87C52X2/54X2/58X2, TSC80251G2D и AT90S8515/8535.

Особенностью данного варианта микроконтроллера является полное отсутствие внешних прерываний. Оперативная память D12 занимает в адресном пространства первые 32 К адресов, а энергонезависимая память таймера реального времени D15 — вторые 32 К. Регистры D9 и D17 занимают адреса 0xFE и 0xFF. Остальные 253 адреса ввода/вывода могут использоваться внешними периферийными платами.

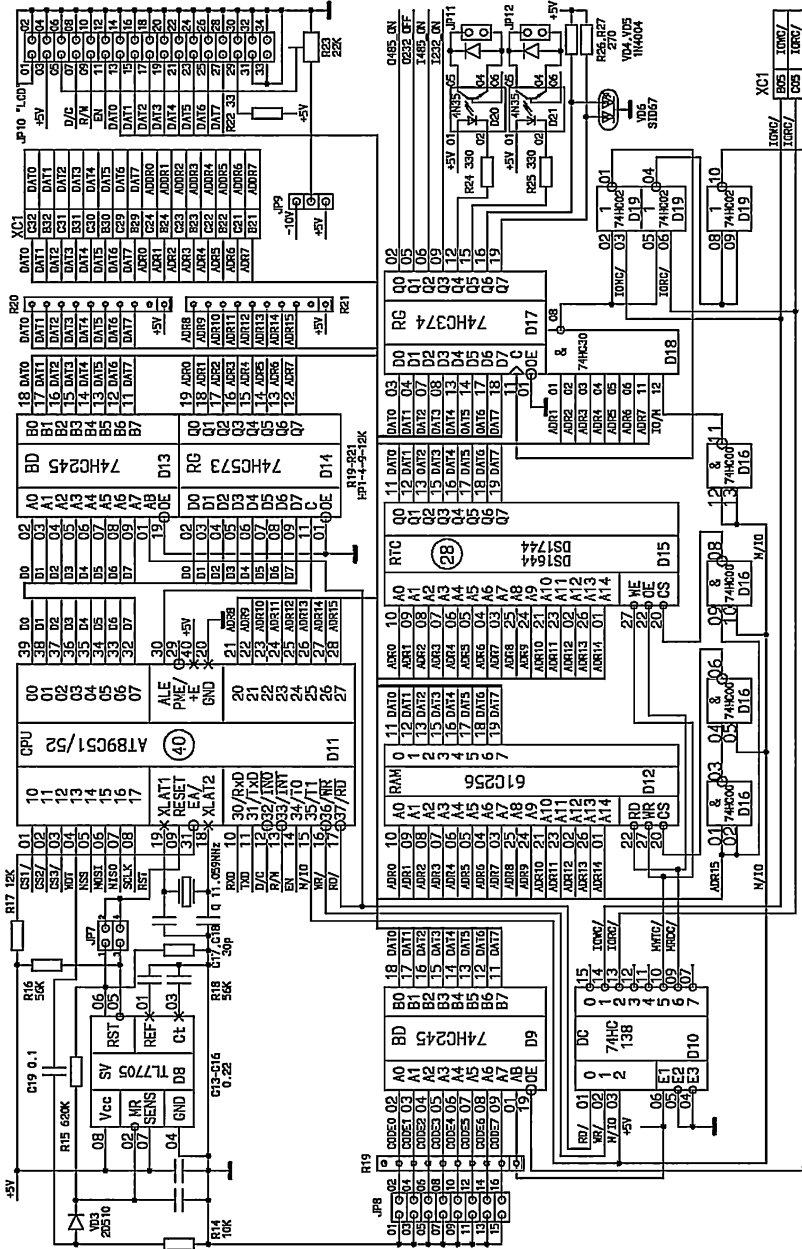


Рис. 4.6. Подсистема контроллера

Описанный вариант контроллера используется в технологических системах синтеза полупроводниковых материалов.

4.4. Мощный универсальный технологический контроллер на микроконтроллере *Cygnal*

В предыдущем разделе была описана принципиальная схема универсального технологического контроллера, выполненного на традиционных микроконтроллерах фирмы *Atmel* серий *AT89/T89/AVR*. Однако в последние три года молодая фирма *Cygnal* смогла создать семь семейств (46 типов) новых x51-совместимых микроконтроллеров [64—66]. Эти микроконтроллеры обладают уникальными характеристиками и набором периферийного оборудования, как аналогового, так и цифрового. К наиболее важным характеристикам в первую очередь следует отнести высокую производительность микроконтроллеров, достигающую 25 MIPS (миллионов инструкций в секунду), а для одного из семи семейств — 100 MIPS. В качестве сравнения можно привести пиковую производительность известной микросхемы фирмы *Atmel AT89C52*, которая при тактовой частоте 12 МГц имеет пиковую производительность всего 1 MIPS! Все микроконтроллеры фирмы *Cygnal* работают при напряжениях питания 2,7—3,6 В, при этом ток потребления большинства из них составляет 10—12 мА. Все микроконтроллеры оснащены системой внутрисистемного программирования *JTAG* с возможностью отладки программ в режиме реального времени. Набор периферии этих микроконтроллеров просто уникален. Большинство из них имеет: 12-/10-битный аналого-цифровой преобразователь со встроенным источником опорного напряжения, входной аналоговый мультиплексор на 8 (до 32) входов, программируемый предварительный усилитель с коэффициентами усиления от 0,5 до 16, два 12-битных цифроаналоговых преобразователя с выходом по напряжению, два компаратора напряжения и встроенный датчик температуры. Все эти узлы имеют большой набор сервисных функций, как, например, программируемые положительный и отрицательный гистерезисы компараторов, формирование прерываний при выходе выходного аналогового кода аналого-цифрового преобразователя за пределы заданного диапазона и т. п. Некоторые семейства имеют еще и второй 8-разрядный встроенный аналого-цифровой преобразователь со своим аналоговым мультиплексором и программируемым предварительным усилителем. В набор цифровой периферии входят такие полезные функциональные узлы, как таймеры (от 3 до 5), мощный контроллер прерываний на 22 вектора (для сравнения напомним, что стандартный 8051 имеет всего 7 векторов прерываний), охраняемый таймер *WDT*, монитор контроля питания, встроенный генератор, *Flash*-память программ и данных с возможностью записи из работающей программы (от 8 до 128 К), встроенная оперативная память (от 1 до 8 К). Состав интерфейсов также необычайно велик. Многие микроконтроллеры имеют интерфейсы *SMBus* (I2C), *SPI*, один или два высокоскоростных (до 115200) интерфейса *UART*. Одно из семейств имеет также интерфейс *CAN 2.0B*. Все микроконтроллеры работают в промышленном диапазоне температур. Перечисленные достоинства

микроконтроллеров фирмы *Cygnal* делают их просто незаменимыми для создания универсальных технологических контроллеров.

На рис. 4.7 показана принципиальная схема универсального технологического контроллера на базе одного из мощных микроконтроллеров фирмы *Cygnal* — *C8051F020*.

Как и контроллер, описанный в предыдущем разделе, описываемый контроллер выполнен в виде модуля магистрально-модульной системы с размерами 100 × 100 мм. Модуль оснащен системным разъемом *СНП96* с модифицированной магистралью *И41*. Модуль представляет собой центральное контроллерное устройство. Подразумевается, что система содержит еще ряд модулей, на которых расположены следующие функциональные узлы:

- входные аналоговые усилители, необходимые для фильтрации и предварительного усиления аналоговых сигналов с датчиков перед подачей их на аналого-цифровой преобразователь;
- мощные выходные нормирующие усилители, необходимые для обеспечения требуемых уровней и мощности выходных сигналов цифроаналоговых преобразователей;
- силовые и высоковольтные ключи с соответствующими регистрами;
- узлы оптической изоляции выходных сигналов.

Перейдем к рассмотрению собственно главного контроллера.

Используемый микроконтроллер фирмы *Cygnal C8051F020 (D3)* имеет 64 К Flash-памяти программ и 4,25 К оперативной памяти. В состав цифровой периферии микроконтроллера входят аппаратный охранный таймер *WDT*, пять таймеров/счетчиков общего назначения, программируемый счетчик-массив, встроенный тактовый генератор до 16 МГц, подсистема сброса со встроенным монитором питания, развитый контроллер прерываний на 22 вектора, два универсальных высокоскоростных интерфейса *UART* и последовательный периферийный интерфейс *SPI*.

Приведенная на рис. 3.34 схема мощного универсального контроллера содержит три подсистемы: подсистему аналогового ввода/вывода, подсистему интерфейсов и подсистему контроллера.

Подсистема интерфейсов содержит:

- драйвер интерфейса *RS485* с оптической изоляцией и низковольтным питанием (3,3 В) — *D1 (MAX3480A)*, подключенный к первому последовательному интерфейсу *UART0*;
- драйвер интерфейса *RS232C* с низковольтным питанием *D2 (MAX3223E)*, подключенный ко второму последовательному интерфейсу *UART1*;
- интерфейсный разъем *JP10* для подключения «интеллектуального» алфавитно-цифрового жидкокристаллического индикатора *LCD* с интерфейсом *L2462*, например фирмы *Seiko Instruments*, подключенный к линиям портов *P2* и *P3*.

Подсистема аналогового ввода/вывода содержит:

- первый встроенный 12-разрядный аналого-цифровой преобразователь с входным мультиплексором. Мультиплексор может работать в двух режимах: однополярном восьмиканальном режиме или дифференциальном че-

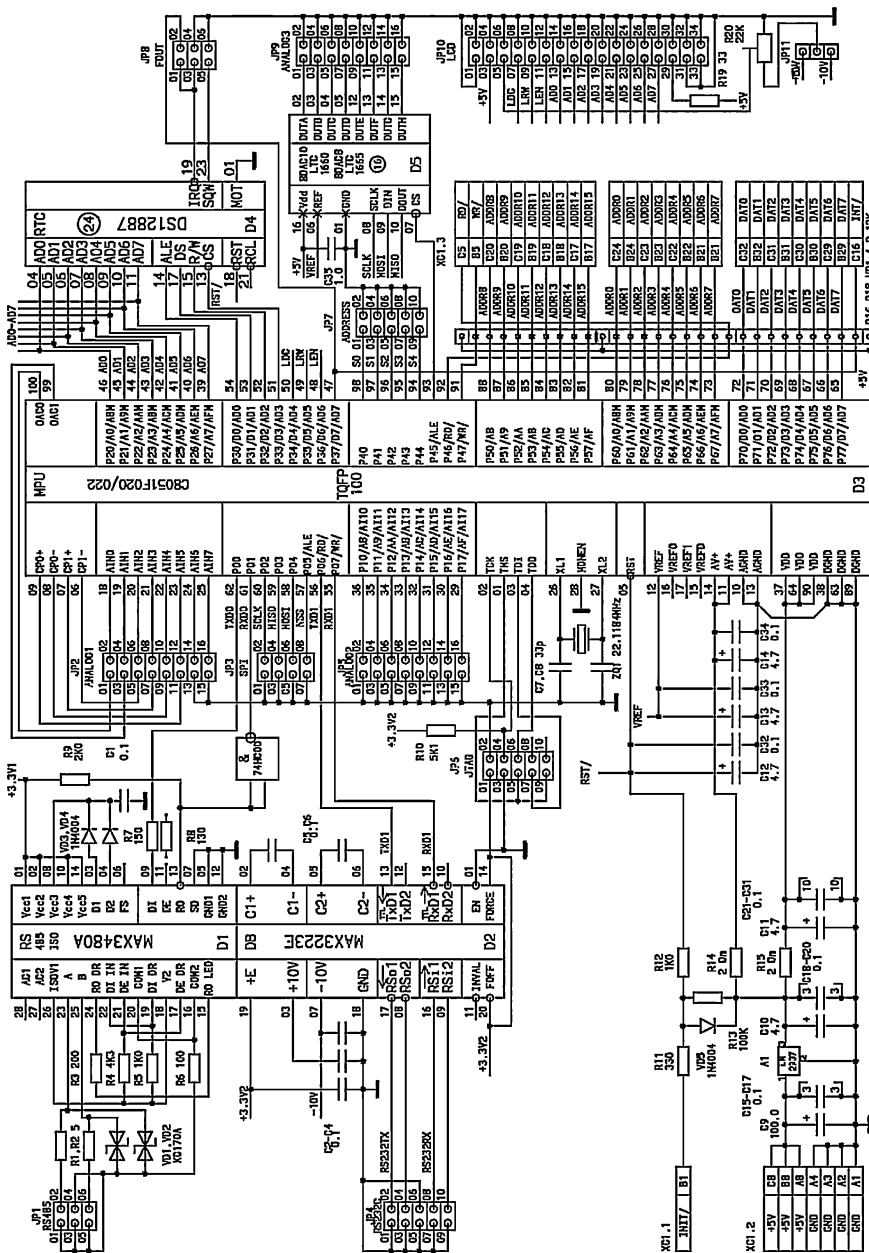


Рис. 4.7. Универсальный технологический контроллер на базе микроконтроллера *C8051F020*

тырехканальном. Между выходом мультиплексора и входом аналого-цифрового преобразователя подключен масштабирующий усилитель с программируемым коэффициентом усиления, который может иметь значения 0,5, 1, 2, 4, 8 и 16. Входы мультиплексора первого аналого-цифрового преобразователя соединены с контактами штыревого двухрядного разъема *JP2 (PLD16)*;

- второй встроенный 8-разрядный быстродействующий аналого-цифровой преобразователь с входным мультиплексором. Мультиплексор может работать в двух режимах: однополярном восьмиканальном или дифференциальном четырехканальном. Между выходом мультиплексора и входом аналого-цифрового преобразователя подключен масштабирующий усилитель с программируемым коэффициентом усиления, который может иметь значения 0,5, 1, 2, 4. Входы мультиплексора второго аналого-цифрового преобразователя соединены с контактами штыревого двухрядного разъема *JP5 (PLD16)*;
- два встроенных 12-разрядных цифроаналоговых преобразователя. Выходы обоих преобразователей соединены с контактами штыревого двухрядного разъема *JP2 (PLD16)*;
- два встроенных компаратора напряжения с возможностью независимой программной установки положительного и отрицательного порогов. Входы обоих компараторов также выведены на штыревой двухрядный разъем *JP2 (PLD16)*;
- встроенный источник опорного напряжения на 2,4 В, от которого работает вся встроенная аналоговая периферия и внешний десятиразрядный восьмиканальный аналого-цифровой преобразователь *D5 (LTC1660)*;
- микросхема восьмиканального десятиразрядного цифроаналогового преобразователя *D5 (LTC1660)*. Эта микросхема использует интерфейс *SPI* микроконтроллера, который также выведен на штыревой двухрядный разъем *JP3 (PLD8)*;
- в аналоговую подсистему также входит преобразователь питания 5 В в 3,3 В, выполненный на микросхеме *LM2937-3.3 (A1)*, предназначенный для питания микроконтроллера *D3* и микросхем с низковольтным питанием *D1, D2*.

Подсистема микроконтроллера включает в себя собственно микроконтроллер *D3* и микросхему таймера реального времени *D4 (DS12887)*. Микроконтроллер имеет восемь портов. На выводы первого порта *P0* с помощью коммутатора ресурсов *Crossbar* выведены сигналы интерфейсов *UART0 (P0.0, P0.1)*, *SPI (P0.2—P0.5)* и *UART1 (P0.6, P0.7)*. На выводы второго порта *P1* настроены входы второго 8-разрядного аналого-цифрового преобразователя. Третий и четвертый порты (*P2* и *P3*) используются для программного доступа к таймеру реального времени *D4* и интерфейсу *LCD*. Пять младших разрядов пятого порта *P4* соединены со штыревым разъемом *PLD10*, который используется для задания (путем установок перемычек) сетевого адреса станции (контроллера). Линия *P4.5* используется для программной выборки внешней микросхемы цифроаналогового преобразователя. Остальные порты *P7—P5* и *P4.6, P4.7* настроены на ра-

боту в режиме открытого истока (т. е. внутренние подтягивающие резисторы выключены) для обеспечения 5-вольтового режима работы и функционирования в качестве не мультиплексированного интерфейса внешних ресурсов (памяти и устройств ввода/вывода). Подтяжку уровней до 5 В обеспечивают внешние резисторные матрицы *R16—R18*. Все линии интерфейса внешних ресурсов (т. е. магистрали) соединены с контактами системного разъема *XC1*. С этого же разъема поступает питание. Штырьковый двухрядный разъем *JP6 (PLD10)* предназначен для подключения *JTAG*-адаптера-программатора (*EC2*). Штырьковый двухрядный разъем *JP8 (PLD6)* предназначен для коммутации выходных сигналов таймера реального времени. Весь контроллер выполнен всего на 6 микросхемах. Потребление от источника питания +5 В составляет 195 мА, из которых 180 мА приходится на микросхему *D1*.

4.5. Универсальная «слотовая» система сбора данных на микроконтроллерах фирмы *Cygnal*

Главным недостатком описанного в предыдущем разделе мощного универсального технологического контроллера является его низкая ремонтпригодность, обусловленная тем, что используемый микроконтроллер припаян непосредственно к плате, т. е. использовано в чистом виде модульное исполнение. Оперативная замена или модернизация такого контроллера связана со значительными технологическими сложностями.

Вообще говоря, главными факторами, определяющими конструктивное исполнение современных микроконтроллерных систем, являются их назначение, области применения и в некоторой степени — используемая элементная база, а точнее — ее конструктивное исполнение.

В настоящее время в микроконтроллерной технике используется четыре основных варианта конструктивного исполнения:

- в виде одноплатных контроллеров;
- в виде многоплатных немагистральных контроллеров;
- магистрально-модульное исполнение;
- «слотовое» исполнение.

Исполнение в виде одноплатных контроллеров, как правило, используется в узкоспециализированных системах. Основными преимуществами такого исполнения являются минимальная стоимость и высокая надежность за счет отсутствия промежуточных разъемных соединений. Недостатками такого типа исполнения являются: низкая гибкость, ремонтпригодность, модифицируемость.

Многоплатное, немагистральное исполнение используется в основном в модифицированных изделиях, в которых к готовому одноплатному контроллеру добавляются дополнительные узлы на дополнительных платах. Недостатками таких конструкций являются: несколько большая стоимость, немного меньшая надежность, но по-прежнему низкая гибкость, ремонтпригодность и модифицируемость.

Магистрально-модульное исполнение используется в универсальных системах. О достоинствах и недостатках таких систем мы достаточно подробно говорили ранее.

Частным случаем магистрально-модульного исполнения является «слотовое» исполнение. В этом случае используется не кросс-плата, а более сложная несущая (материнская) плата, на которой кроме собственно магистрали могут располагаться и другие узлы системы, как правило, имеющие универсальное назначение и большой срок морального устаревания. Этот вид конструктивного исполнения широко используется, например, в современных персональных компьютерах.

Таким образом, при создании универсальных микроконтроллерных систем наибольший интерес представляют магистрально-модульный и «слотовый» варианты исполнения. Еще большую актуальность приобретают эти варианты исполнения при использовании современных быстроразвивающихся микроконтроллерных систем сбора данных, ориентированных на работу в распределенных системах управления, построенных на базе командно-информационных сетей. Например, магистрально-модульное исполнение узла главного микроконтроллера позволяет легко модифицировать систему и наращивать ее вычислительную мощность путем замены модуля микроконтроллера на модуль с более мощным и производительным микроконтроллером.

Естественно, что при создании универсальной модульной микроконтроллерной системы, претендующей на широкий спектр областей применения, необходимо использовать наиболее мощные высокопроизводительные и перспективные микроконтроллеры. При этом желательно, чтобы выбранные микроконтроллеры выпускались с различным набором встроенных функциональных узлов. Поставленным условиям как нельзя лучше удовлетворяют самые мощные и высокопроизводительные микроконтроллеры фирмы *Cygnal*, описанные в разделе 4.1.3 (см. табл. 4.15).

Для выбора приемлемых типов микроконтроллеров опишем критерии отбора:

1. Первым критерием отбора является пиковая производительность. Обычно при наличии достаточно широкого диапазона этой величины (от 20 до 100) выбирается значение, которое присуще большинству типов микроконтроллеров. В нашем случае это 25 *MIPS*.

2. Вторым важным критерием является количество линий ввода/вывода (или байтных портов), которое в универсальных системах не должно быть меньше, чем у стандартного микроконтроллера *i8051*. Иными словами, количество портов ввода/вывода не должно быть меньше 4 (соответственно, линий ввода/вывода не должно быть меньше 32). Количество линий ввода/вывода более 32 в универсальных модульных системах избыточно и приводит к увеличению количества выводов корпуса, количеству контактов универсальной магистрали, а соответственно, сложности, площади и стоимости печатной платы.

3. Третьим критерием является обязательное наличие стандартных интерфейсов *UART* и *SPI*. Что касается интерфейсов *SMBus (I2C)* и *CAN*, то они используется в настоящее время достаточно редко и их наличие не обязательно.

4. Четвертый критерий — желательно наличие хотя бы одного многоканального аналого-цифрового преобразователя *ADC*. Причем чем большее коли-

чество входов имеет *ADC*, тем больше универсальность проектируемой микроконтроллерной системы. Наличие двух *ADC* для универсальных систем не является обязательным.

5. С целью повышения универсальности системы желательно наличие в системе цифроаналоговых преобразователей.

6. С целью повышения универсальности желательно иметь большой объем встроенной оперативной памяти.

7. Программное обеспечение современных микроконтроллерных универсальных систем редко превышает 8 К, поэтому объем встроенной *Flash*-памяти не является определяющим фактором.

8. С целью снижения сложности и стоимости печатной платы желательно использовать микроконтроллеры с меньшим количеством выводов.

9. Очевидным критерием отбора является стоимость микроконтроллера.

С учетом приведенных критериев модифицируем табл. 4.15. При этом удалим из нее все микроконтроллеры с пиковой производительностью менее 25 MIPS, с количеством линий ввода/вывода меньшее 32. С целью повышения читаемости удалим из таблицы колонки наличия компараторов, второго аналого-цифрового преобразователя *ADC1*, а также колонки охранного таймера *WDT* и интерфейса *SPI*, подразумевая, что они есть во всех рассматриваемых моделях. Также исключим все типы микроконтроллеров, имеющие число портов линий ввода/вывода более 32 и корпус более 64 выводов. Кроме того, добавим колонку со значением цены [121] и отсортируем полученные данные сперва по возрастанию стоимости, а затем по семействам. Полученный результат представлен в табл. 4.19.

Таблица 4.19

Тип	Цена, \$	Корпус	FLASH, К	RAM	Port I/O	UART	ADC0	DAC-12
C8051F230-25	7,78	TQFP-48	8	256	32	1	–	–
C8051F220-25	9,24	TQFP-48	8	256	32	1	8/32	–
C8051F236-25	9,54	TQFP-48	8	256+1 К	32	1	–	–
C8051F226-25	11,00	TQFP-48	8	256+1 К	32	1	8/32	–
C8051F206-25	15,20	TQFP-48	8	256+1 К	32	1	12/32	–
C8051F018-25	15,49	TQFP-64	16	256+1 К	32	1	10/8	–
C8051F015-25	22,30	TQFP-64	32	256+2 К	32	1	10/8G	2
C8051F005-25	25,23	TQFP-64	32	256+2 К	32	1	12/8G	2
C8051F023-25	24,71	TQFP-64	64	256+4 К	32	2	10/8G	2
C8051F021-25	27,64	TQFP-64	64	256+4 К	32	2	12/8G	2
C8051F041-25	23,65	TQFP-64	64	256+4 К	32	2	12/13G	2
C8051F043-25	24,72	TQFP-64	64	256+4 К	32	2	10/13G	2
C8051F127-50	27,44	TQFP-64	128	256+8 К	32	2	10/8G	2
C8051F125-50	30,00	TQFP-64	128	256+8 К	32	2	12/8G	2
C8051F123-100	32,99	TQFP-64	128	256+8 К	32	2	10/8G	2
C8051F121-100	36,51	TQFP-64	128	256+8 К	32	2	12/8G	2

В результате первичного отбора мы получили таблицу, включающую 16 типов микроконтроллеров, входящих в состав 6 семейств, пригодных для создания универсальной модульной системы. Дальнейший анализ полученной таблицы с использованием описания выводов семейств позволяет сделать следующие выводы:

1. Отобранные типы микроконтроллеров выпускаются в двух типах корпусов: *TQFP48* и *TQFP64*.

2. Первые пять микроконтроллеров *C8051F206/220/226/230/236* входят в одно семейство *C8051F2xx*. В соответствии с технической документацией [123] они имеют: одинаковую разводку выводов и выпускаются в корпусе *TQFP48*; одинаковый объем *Flash*-памяти, равный 8 К. Основные их отличия заключаются в наличии у микроконтроллеров *C9051F206/226/236* встроенной дополнительной оперативной памяти *XRAM* объемом в 1 К, расположенной в пространстве адресов внешней памяти. Кроме того, у микроконтроллеров *C8051F220/226* имеется восьмиразрядный аналого-цифровой преобразователь (*ADC*), а у микроконтроллера *C8051F206* двенадцатиразрядный *ADC*. Аналого-цифровые преобразователи имеют встроенный входной мультиплексор на 32 входа. Особенностью этого семейства является фиксированная привязка входов/выходов цифровой периферии к определенным линиям портов ввода/вывода и, соответственно, к выводам корпуса, а также возможность использования всех линий портов ввода/вывода в качестве аналоговых входов *ADC*. Кроме того, микроконтроллеры этого семейства имеют достаточно низкую стоимость. Вышеперечисленные качества этой группы микроконтроллеров позволяют использовать их для разработки простого модуля главного контроллера для низко-стоимостных универсальных микроконтроллерных систем.

3. Следующие три типа *C8051F005/015/018* принадлежат к двум семействам микроконтроллеров *C8051F0xx* [124] и *C8051F018-9* [125]. Они имеют одинаковую разводку выводов и выпускаются в корпусе *TQFP64*. Основные их отличия заключаются в объеме *Flash*-памяти 16 К (*C8051F018*) и 32 К (*C8051F005/015*), объеме встроенной дополнительной оперативной памяти *XRAM*, расположенной в пространстве адресов внешней памяти — 1 К (*C8051F018*) и 2 К (*C8051F005/015*). Микроконтроллеры *C8051F018/015* имеют встроенный десятиразрядный *ADC*, а микроконтроллер *C8051F005* — двенадцатиразрядный *ADC*. Все микроконтроллеры имеют также встроенный аналоговый восьмиканальный мультиплексор *AMUX* на восемь входов. Микроконтроллеры *C8051F005/015* имеют также встроенный программируемый усилитель *PGA* с коэффициентами усиления 0,5, 1, 2, 4, 8, 16 и два двенадцатиразрядных цифро-аналоговых преобразователя (*DAC*). Все микроконтроллеры этой группы имеют индивидуальные аналоговые входы *ADC* и компараторов *CPT* и выходы *DAC*. Кроме того, особенность встроенного коммутатора ресурсов *Crossbar* обеспечивает компактную «плавающую» (программируемую) привязку входов/выходов цифровой периферии к определенным линиям портов ввода/вывода и, соответственно, к выводам корпуса. Эта группа микроконтроллеров имеет среднюю стоимость. Вышеперечисленные качества этой группы микроконтроллеров позволяют использовать их для разработки модуля главного

контроллера средней мощности для универсальных микроконтроллерных систем средней сложности.

4. Остальные микроконтроллеры *C8051F021/023/041/043/121/123/125/127* принадлежат к трем наиболее мощным семействам *C8051F02x* [126], *C8051F04x* [127] и *C8051F12x* [128] микроконтроллеров фирмы *Cygnal*. Они имеют одинаковую разводку выводов и выпускаются в корпусе *TQFP64*. Важно отметить, что микроконтроллеры *C8051F021/023/041/043*, *C8051F125/127* и *C8051F121/123* способны работать на частотах до 25, 50 и 100 МГц соответственно, при этом их пиковая производительность составляет соответственно 25, 50 и 100 MIPS! Основные их отличия заключаются в объеме Flash-памяти 64 К (*C8051F021/023/041/043*) и 128 К (*C8051F121/123/125/127*), объем встроенной дополнительной оперативной памяти *XRAM*, расположенной в пространстве адресов внешней памяти — 4 К (*C8051F021/023/041/043*) и 8 К (*C8051F121/123/125/127*). Микроконтроллеры *C8051F023/043/123/127* имеют встроенный десятиразрядный *ADC1*, а микроконтроллер *C8051F021/041/121/125* — двенадцатиразрядный *ADC1*. Все микроконтроллеры имеют второй восьмиразрядный *ADC2*. Все микроконтроллеры имеют также встроенные аналоговые восьмиканальные мультиплексоры *AMUX1* и *AMUX2* на восемь входов (тринадцать входов для *C8051F041/043*). Все микроконтроллеры имеют также встроенные программируемые усилители *PGA1* с коэффициентами усиления 0,5, 1, 2, 4, 8, 16 и *PGA2* с коэффициентами усиления 0,5, 1, 2, 4, а также два двенадцатиразрядных *DAC*. Все микроконтроллеры этой группы имеют индивидуальные аналоговые входы *ADC1* и компараторов *CPT*, интерфейса *CAN* (только для *C8051F041/043*) и выходы *DAC*. Кроме того, особенность встроенного коммутатора ресурсов *Crossbar* обеспечивает компактную «плавающую» (программируемую) привязку входов/выходов цифровой периферии к определенным линиям портов ввода/вывода и, соответственно, к выводам корпуса. Эта группа имеет высокую стоимость. Вышеперечисленные качества данной группы микроконтроллеров позволяют использовать их для разработки модуля главного контроллера большой мощности для высокопроизводительных универсальных микроконтроллерных систем большой сложности.

5. Таким образом, на базе отобранных микроконтроллеров возможно создание трех вариантов модулей главного контроллера для универсальной модульной микроконтроллерной системы, отличающихся производительностью, составом периферийного оборудования и стоимостью.

Рассмотрим основные требования к разрабатываемым модулям главного контроллера универсальной микроконтроллерной системы:

1. Исходя из основных принципов построения магистрально-модульных систем, все модули должны иметь одинаковые размеры по ширине с установленным одинаковым магистральным разъемом. Магистральный разъем должен иметь одинаковые линии связи для всех исполнений. Количество линий связи должно быть минимально возможное. Высота модулей может незначительно отличаться.

2. Дополнительные входы/выходы модулей, связанные, например, с аналоговыми узлами, или являющиеся входами/выходами дополнительных модулей,

должны иметь дополнительный штыревой разъем (или разъемы), устанавливаемый вдоль широкой стороны поля печатной платы модуля.

3. Функциональный состав модуля должен обеспечивать максимальные функциональные возможности при минимальной его стоимости.

Исходя из приведенных основных требований к модулям главного микроконтроллера, а также функциональных особенностей примененных микроконтроллеров предлагается следующая номенклатура связей магистрального разъема (см. табл. 4.20).

Таблица 4.20

Наименование цепи	Номер контакта		Номер контакта	Наименование цепи
Общий провод GND	01		02	Общий провод GND
Линия порта P3.7	03		04	Линия порта P0.7
Линия порта P3.6	05		06	Линия порта P0.6
Линия порта P3.5	07		08	Линия порта P0.5
Линия порта P3.4	09		10	Линия порта P0.4
Линия порта P3.3	11		12	Линия порта P0.3
Линия порта P3.2	13		14	Линия порта P0.2
Линия порта P3.1	15		16	Линия порта P0.1
Линия порта P3.0	17		18	Линия порта P0.0
Линия порта P2.7	19		20	Линия порта P1.7
Линия порта P2.6	21		22	Линия порта P1.6
Линия порта P2.5	23		24	Линия порта P1.5
Линия порта P2.4	25		26	Линия порта P1.4
Линия порта P2.3	27		28	Линия порта P1.3
Линия порта P2.2	29		30	Линия порта P1.2
Линия порта P2.1	31		32	Линия порта P1.1
Линия порта P2.0	33		34	Линия порта P1.0
Упр. опорным напряжением MONEN	35		36	Линия сброса RST/
Вход опорного напряжения VREF	37		38	Напряжение питания +E
Общий провод GND	39		40	Общий провод GND

В качестве магистрального разъема, устанавливаемого на кросс-плате при магистрально-модульном исполнении или на материнской плате при слотовом исполнении, используется двухрядный гнездовой разъем *PBD-40* (2×20). В качестве системного магистрального разъема *J1* на модулях используется угловой двухрядный штыревой разъем *PLD-R-40* (2×20). Все контакты разъемов магистралей, за исключением 1, 2 и 35, имеют параллельное соединение. Контакты 1, 2 и 35 могут в различных модулях использоваться для различных целей, например для подачи на модули необходимых напряжений питания. Эти контакты на

магистральной в системе соединяются необходимым образом с помощью «струнного» монтажа.

На рис. 4.8 показана принципиальная схема простого модуля главного контроллера (МГК), выполненная на базе микроконтроллеров первой группы *C8051F206/220/226/230/236*. Она содержит собственно сам микроконтроллер D1, стабилизатор напряжения на 3,5 В — A1 (*R1X2*, 3,5 В, 150 мА), кварцевый резонатор Q1 (22,1182 или 11,0592 МГц) с конденсаторами C1 и C2, разъем программирования по интерфейсу *JTAG* — J2 и магистральный разъем J1. На рис. 4.9 показан чертеж расположения элементов на печатной плате с габаритными размерами 52 × 24 мм. На рис. 4.10 и 4.11 показаны лицевая и обратная (зеркально) стороны печатной платы. Обратная сторона платы имеет малую плотность связей, а лицевая — высокую, что связано с разводкой планарного корпуса *TQFP48* на плате довольно малой высоты. Для программирования по интерфейсу *JTAG* использован двухрядный штыревой разъем *PLD-10* (2 × 5) со стандартной разводкой для программатора *EC-2*.

На рис. 4.12 представлена принципиальная схема МГК, выполненного на базе микроконтроллеров второй группы *C8051F005/015/018*. По сравнению с предыдущей схемой, в рассматриваемую добавлен штыревой разъем J3 — *PLD16* (2 × 8), с контактами которого соединены все аналоговые входы/выхо-

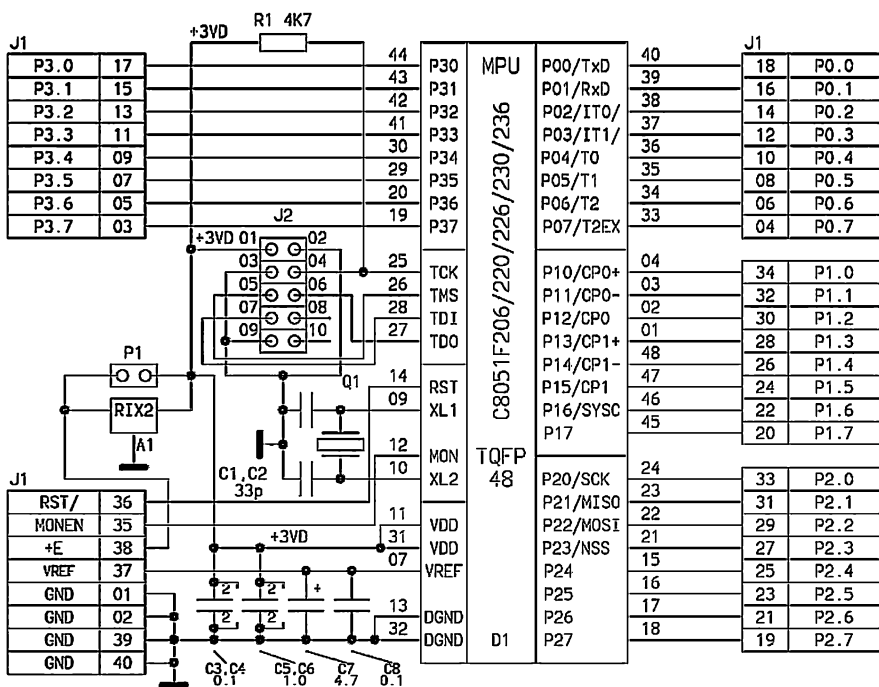


Рис. 4.8. Принципиальная схема простого модуля главного контроллера (МГК)

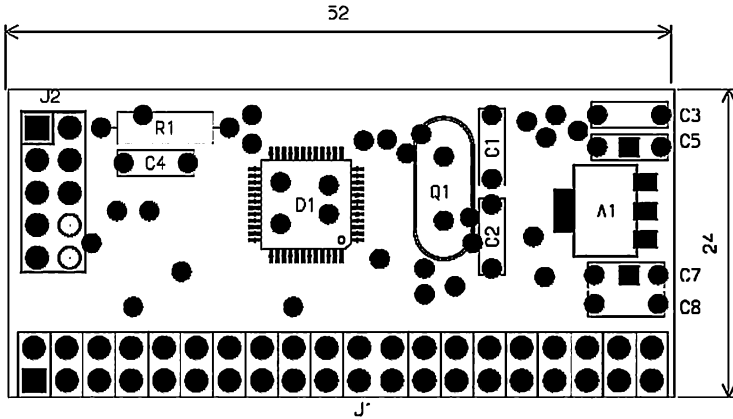


Рис. 4.9. Схема расположения элементов на печатной плате простого МГК

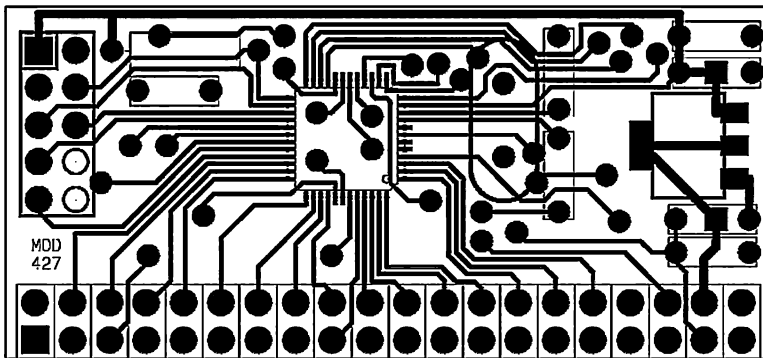


Рис. 4.10. Лицевая сторона печатной платы простого МГК

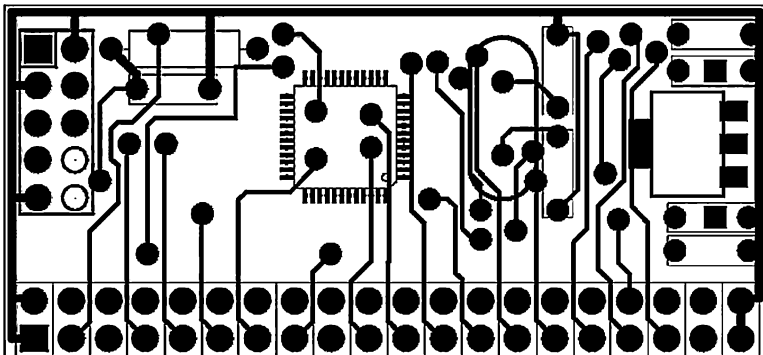


Рис. 4.11. Обратная сторона печатной платы простого МГК

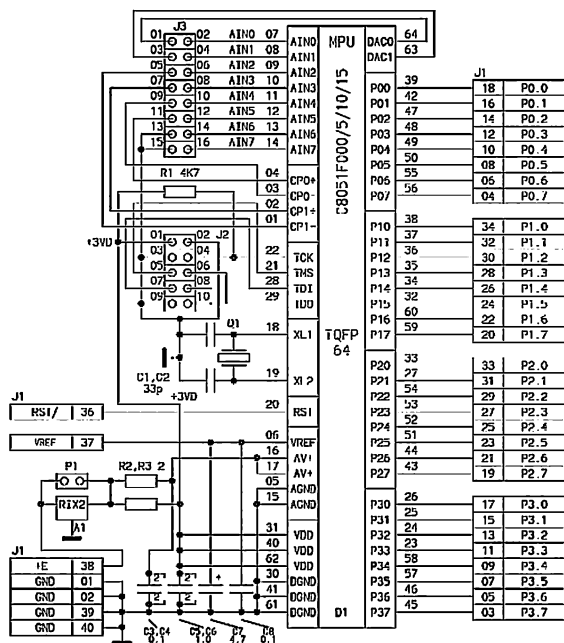


Рис. 4.12. Принципиальная схема *МГК* средней мощности

ды, а также два резистора *R2* и *R3* по 2 Ом каждый и конденсатор *C9* для формирования цифрового и аналогового напряжений питания.

На рис. 4.13 представлена принципиальная схема *МГК*, выполненного на высокопроизводительных мощных микроконтроллерах *C8051F021/023/041/043/121/123/125/127*. Она практически не отличается от предыдущей схемы.

Модули главного контроллера средней и большой мощности выполнены на печатных платах с размером 52 × 32 мм. Чертеж расположения элементов на печатной плате этих модулей показан на рис. 4.14 (положение ключа микросхемы *D1* показано для *МГК* средней мощности).

Выше были рассмотрены три варианта модулей главного контроллера. Естественно, что в универсальной модульной системе кроме этих основных модулей должны быть и другие модули. Однако при их разработке возникают определенные сложности, связанные с наличием у микроконтроллеров второй и третьей групп встроенного коммутатора ресурсов *Crossbar*, который осуществляет «плавающую» привязку внутренних периферийных узлов к выводам корпуса. Для выработки рекомендаций по разработке периферийных узлов следует составить таблицу (см. табл. 4.21) распределения входов/выходов обязательных периферийных узлов для различных групп. При этом к обязательным следует отнести интерфейсы *UART* (в семействах, содержащих два интерфейса *UART*, следует

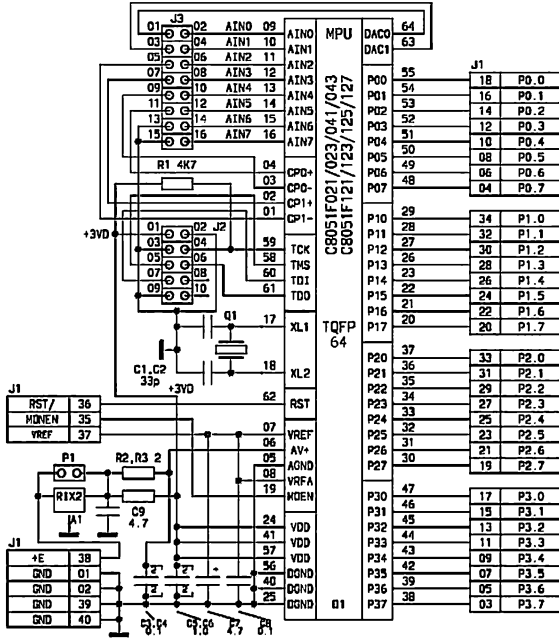


Рис. 4.13. Принципиальная схема высокопроизводительного МГК большой мощности

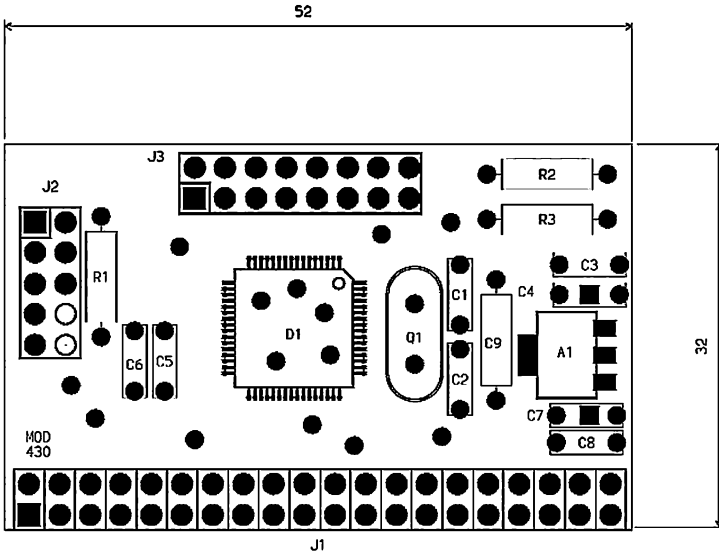


Рис. 4.14. Схема расположения элементов на печатных платах МГК средней и большой мощности

указать оба интерфейса), *SPI*, входов внешних прерываний и интерфейса внешней памяти в мультиплексированном режиме (если он аппаратно реализован в данной группе).

Таблица 4.21

Периферия	Группа	Порт P0							Порт P1							Порт P2				Порт P2			
		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4-7	
UART1	1	U	U																				
SPI																							
INT 0&1				I	I																		
UART1	2					U	U																
SPI		S	S	S	S																		
INT 0&1																							
UART1	3	U	U																				
SPI				S	S	S																	
UART2																							
INT0																							
INT1																							
EMI							ALE	RD/	WR/														
		Address High																	Mixed Data \$ Address Low				

Проще всего дело обстоит с первой группой, у которой входы/выходы внутренних периферийных узлов фиксированы на определенные выводы портов. Правда, расположение выводов интерфейса *SPI* в диапазоне выводов порта *P2* не совсем понятно. Во второй группе все входы/выходы необходимой периферии расположены в диапазоне выводов порта *P0*, а все остальные порты свободны. В третьей группе порт *P3* занят мультиплексированными линиями данных, младшего байта адреса интерфейса внешней памяти *EMI*, порт *P2* также занят линиями старшего байта адреса *EMI*, кроме этого, сигналы *EMI* (*ALE*, *RD/*, *WR/*) занимают фиксированные линии порта *P0.5—P0.7*. Остальные периферийные устройства этой группы располагаются на оставшихся свободными линиях портов *P0* и *P1*.

Следует также учитывать, что аналоговые входы *ADC* в микроконтроллерах первой группы могут занимать любые линии портов ввода/вывода. Во второй группе единственный *ADC* имеет индивидуальные линии входов. В третьей группе имеется два *ADC*, 0 и 1. *ADC0* имеет индивидуальные линии входов, а *ADC1* может использовать для аналоговых входов только линии порта *P1*.

Анализ сведений, приведенных в таблице 4.21 и описанных выше, позволяет сделать следующие рекомендации, которые необходимо учитывать при разработке дополнительных модулей универсальной микроконтроллерной системы:

1. Шину данных аппаратного или программного интерфейса внешней памяти (*EMI*), мультиплексированную с младшим байтом адреса, следует располагать на линиях порта *P3*. Шину старшего байта адреса в семействах два и три

рекомендуется располагать на линиях порта *P2*. В первом семействе с целью экономии линий ввода/вывода рекомендуется использовать двойное мультиплексирование адреса на линиях порта *P3*. Исходя из вышесказанного, при разработке универсального модуля внешней оперативной памяти рекомендуется: использовать мультиплексирование данных и младшего байта адреса на линиях порта *P3*. Что касается старшего байта адреса, то следует иметь возможность подключать дополнительный регистр-защелку для старшего байта адреса, либо на порт *P2* (при отключенной линии управления захвата и прямом пропускании адреса), либо на порт *P3* с захватом мультиплексированного адреса по сигналу дополнительной линии управления.

2. При создании модуля универсальных интерфейсов *RS232C* и/или *RS485* следует помнить, что для универсальности необходимо иметь на модуле два канала интерфейса, а также обеспечить возможность коммутации внутренних входов/выходов на линии *P0.0, P0.1, P0.4, P0.5, P1.1, P1.2*.

3. При создании универсальных модулей, связанным с интерфейсом *SPI*, необходимо помнить о необходимости коммутации внутренних сигналов на линии портов *P0.0—P0.5, P1.0, P2.0—P2.3*.

4. При создании универсальных модулей, связанным с прерываниями, следует помнить о необходимости коммутации внутренних сигналов на линии портов *P0.2, P0.3, P0.6, P0.7, P1.3, P1.4*.

5. Исходя из замечаний, приведенных в пп. 1—4, следует отметить, что значительно упростить коммутацию внутренних сигналов можно, если разрабатывать универсальные дополнительные модули не для всех трех групп, а для двух. Например, модуль дополнительной оперативной памяти для второй и третьей групп будут отличаться только одним сигналом *ALE*. Аналогичные компромиссы можно найти и для других модулей.

В заключение следует отметить, что возможно создание магистрально-модульных микроконтроллерных систем с другими типами и принципами создания магистралей. Например, можно облегчить задачу совместимости интерфейсов в периферийных дополнительных модулях, если на печатных платах *МГК* все обязательные интерфейсы выводить и на дополнительные унифицированных штыревые разъемы (кроме магистрали). При этом подключение внутренних сигналов основных интерфейсов к дополнительным модулям может осуществляться через плоские дополнительные кабели. Еще один вариант заключается в разводке магистрали не по признаку линий портов, а по признаку назначения сигналов обязательных интерфейсов. Очевидно, что все это должен учитывать разработчик при создании своей проблемно-ориентированной или универсальной системы.

4.6. Специализированные технологические контроллеры-фотодатчики

Наше ознакомление со схемотехникой технологических микроконтроллеров было бы неполным, если бы мы не рассмотрели несколько вариантов специализированных технологических контроллеров, а также не привели несколько при-

меров программирования. В этом разделе мы ознакомимся с двумя вариантами реализации специализированных технологических контроллеров-фотодатчиков, работающих на отражение.

При создании различных систем автоматизации производства, систем складского учета, систем учета движения и маркировки готовой продукции одной из основных задач является разработка фотодатчика движения продукции по транспортерной ленте, работающего на отражение (далее просто *ФД*). К *ФД* предъявляются достаточно жесткие требования. Он должен работать «на отражение», т. е. излучать определенное излучение в сторону продвигающихся по транспортеру объектов и воспринимать отраженное от объектов излучение. Он должен быть малочувствителен к возможным внешним засветкам и фоновому облучению. Схема *ФД* должна позволять включать (выключать) его в работающую систему. Схема должна быть устойчива к сбоям по питанию, иметь защиту от зависаний. Зачастую *ФД* должен обеспечивать возможность работы в составе локальной сети. Кроме того, *ФД* должен иметь достаточно малые размеры и малое энергопотребление, а также питаться от одного источника питания. На рис. 4.15 показан первый вариант основной принципиальной схемы *ФД*. Приведенная принципиальная схема реализует только основную функцию — определение наличия объекта перед фотодатчиком и выдачу стробирующего импульса.

Принципиальная схема содержит следующие основные элементы: излучающий сверхяркий красный светодиод *SD1* со схемой управления на транзисторе *VT1* и резисторах *R4—R6*; принимающий фотодиод *FDI*; предварительный усилитель фотодиода *A1* (*TBA2800*), супервизор питания со схемой защиты от зависаний (*WDT*) *D1* (*TL7705*), микроконтроллер *D2* (*AT89C2051*).

В качестве излучающего сверхяркого красного светодиода *SD1* применен *L93-SR-C*. На светодиод *SD1* подаются импульсы тока через транзистор *VT1* и резистор *R4* (1,5 Ом) с амплитудой около 400 мА (длительность импульса

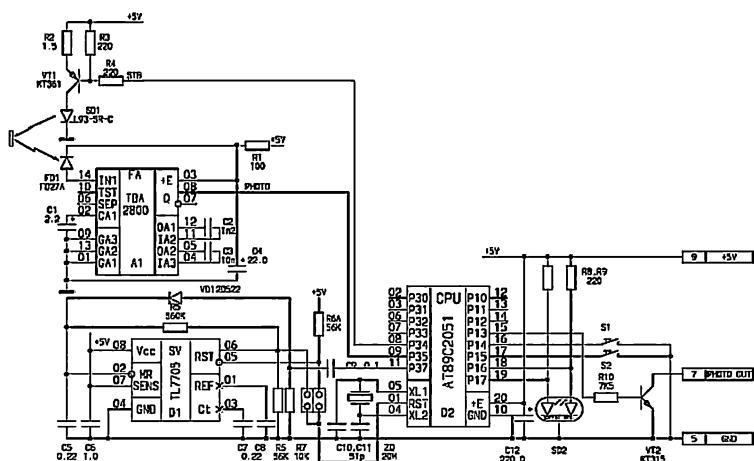


Рис. 4.15. Основная принципиальная схема первого варианта фотодатчика

~75 мкс (при скважности около 10). Светодиод *SD1* и фотодиод *FD1* расположены соосно окнами в одну сторону (сторону объекта) в дюралеовой крышке цилиндрического корпуса, которая одновременно служит теплоотводом. Желательно, чтобы светодиод и фотодиод были расположены под углом 1—2 градуса. Фотодиод *FD1* соединен со входом инфракрасного предварительного усилителя *D1*.

Инфракрасный предварительный усилитель *TBA2800* фирмы *Micronas Intermetall* [129] предназначен для предварительного усиления инфракрасных импульсных сигналов. Усилитель содержит четыре узла: управляемый усилитель I, усилитель II, усилитель выделения импульсов III и инвертор IV (функциональная схема микросхемы *TBA2800* показана на рис. 4.16).

Назначение выводов микросхемы *TBA2800*: 1 — общий вывод входного сигнала; 2 — вывод подключения конденсатора к усилителю I; 3 — вывод напряжения питания; 4 — вход усилителя III; 5 — выход усилителя II; 6 — вывод для регулировки разделения сигнал/шум; 7 — выход отрицательных импульсов; 8 — выход положительных импульсов; 9 — общий вывод выхода; 10 — тестовый вывод; 11 — вход усилителя II; 12 — выход усилителя I; 13 — общий вывод усилителя II; 14 — вход усилителя I.

Усилитель I имеет большой динамический диапазон. Благодаря этому гарантируется работоспособность при ярком окружающем свете и в условиях засветки модулированным светом с частотой 50 Гц от флуоресцентных ламп (ламп дневного света). Кроме того, работоспособность сохраняется и при засветке интенсивным красным излучением. Такая ситуация, например, возможна, когда отражающая поверхность объекта находится в непосредственной близости от излучающего и приемного диодов. Усилитель II служит для дальнейшего усиления сигнала, а усилитель III — для отделения импульсной составляющей от шумов. Инвертор IV обеспечивает дополнительное инвертирование отрицательных импульсов с выхода усилителя III (вывод 7) в положительные импульсы (вывод 8). Введение дополнительного резистора между выводом 6 и общим проводом увеличивает шумовую устойчивость, но снижает чувствительность по входу. Вывод 10 используется как тестовый.

Конденсатор, подключаемый между выводом 2 и общим проводом питания, влияет на автоматическое изменение динамического диапазона усилителя I. Для устойчивой работы усилителя I емкость конденсатора должна быть больше 1 мкФ. Микросхема сохраняет работоспособность от 4,5 до 6 В, максимальный ток потребления не более 2 мА.

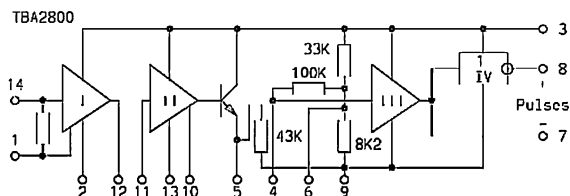


Рис. 4.16. Функциональная схема микросхемы *TBA2800*

В описываемой схеме ФД микросхема *TVA2800* используется для усиления принятого фотодиодом *FD1* красного модулированного сигнала, излучаемого светодионом *SD1*, и отраженного от движущегося мимо ФД объекта. Выходной сигнал *PHOTO* с микросхемы *D1* (вывод 8) поступает на вход *P3.5* (вывод 09) микроконтроллера *D2*.

На микросхеме *D1 (TL7705)* собран супервизор питания со схемой защиты от зависаний *WDT*.

В описываемой схеме ФД использован микроконтроллер *AT89C2051* фирмы *Atmel*. Отметим только некоторые особенности включения. Микроконтроллер работает на тактовой частоте 20 МГц.

Линии порта 3 используются следующим образом:

- линии *P3.2* и *P3.3* — тестовые выходы, на которых формируются эха импульсов опознавания предмета;
- линия *P3.4* — выход отрицательных включающих импульсов для управления светодиодом *SD1*;
- линия *P3.5* — вход для анализа сигналов от предварительного усилителя *A1* фотодиода *FD1*;
- линия *P3.7* — выход подзаряжающих импульсов на *WDT*.

Линии порта 1 используются следующим образом:

- линия *P1.7* — управление зеленым светодиодом;
- линия *P1.6* — управление красным светодиодом;
- линии *P1.5* и *P1.4* — анализируют состояние кнопок установки порога срабатывания фотодатчика;
- линия *P1.3* — является линией выходного импульса.

Общий алгоритм работы ФД:

1. Микроконтроллер *D2* переводит линию порта *P3.4* в состояние логического нуля, это состояние открывает транзистор *VT1* и включает светодиод *SD1*.

2. Фотодиод *FD1* принимает отраженный сигнал. Если перед ФД на расстоянии примерно 6—7 см имеется отражающая поверхность объекта, на выходе предварительного усилителя *A1* (и соответственно, на входной линии порта *P3.5* микроконтроллера длительность импульса достигает 40—70 мкс (в зависимости от наличия паразитных засветок ФД, расстояния до объекта, его формы, скорости и отражающих свойств поверхности). Если перед ФД нет объекта (или он находится слишком далеко от ФД), длительность импульса изменяется от нуля до 20 мкс.

3. Микроконтроллер несколько раз опрашивает состояние входной линии порта *P3.5* и считает количество принятых импульсов, затем переводит линию порта *P3.4* в состояние логической единицы (отключает импульс на красный светодиод).

4. Далее микроконтроллер выдает на тестовые выходы серии коротких положительных импульсов, количество которых равно количеству принятых импульсов и установленному порогу. Длительность импульса — 1 мкс, период следования — 10 мкс.

5. Микроконтроллер анализирует количество принятых импульсов и сравнивает это количество с определенной пороговой величиной. Если превышение

пороговой величины происходит впервые, то устанавливается внутренний флаг наличия объекта, если же наблюдается снижение количества принятых импульсов ниже пороговой величины, флаг наличия объекта сбрасывается.

6. Микроконтроллер проверяет взведение флага наличия объекта и на выходе *PI.3* формирует положительный выходной импульс длительностью примерно 400 мкс. Одновременно микроконтроллер устанавливает соответствующим образом выходные линии *PI.6* и *PI.7*, включая зеленый светодиод двоянного индикатора *SD2*.

7. Далее микроконтроллер анализирует активность флага наличия объекта, если флаг взведен, микроконтроллер включает красный светодиод *SD2*, а если флаг сброшен, — выключает оба светодиода.

8. Далее формируется временная задержка и цикл повторяется вновь (см. п. 1).

Таким образом, в отсутствие объекта перед *ФД* индикатор работы (установленный в заднем торце корпуса) — светодиод не включен, при появлении объекта на 400 мкс включается зеленый светодиод, а затем во все время нахождения этого же объекта горит красный светодиод.

Программа, соответствующая описанному алгоритму, будет приведена далее.

В заключение этого раздела отметим назначение некоторых элементов принципиальной схемы (см. рис.).

Потребляемый *ФД* ток не превышает 25 мА.

Фотодатчик размещен в дюралевои цилиндре диаметром 18 мм и длиной 110 мм. Цилиндр имеет две торцевые крышки, крепящиеся к корпусу винтами *M2*. В одной торцевой крышке установлены светодиод *SD1* и фотодиод *FD1*. В другом торце установлен двоянный индикаторный светодиод *SD2* и выходит соединительный кабель длиной 1,5 м, снабженный штыревым разъемом *DB9V*. В отдельных случаях предусматривается регулировка чувствительности *ФД*, выполненная в виде вкручивающегося в окно фотодиода винта или специальной шторки. В зависимости от условий эксплуатации *ФД* может иметь открытые отверстия, в которые установлены светодиод *SD1* и фотодиод *FD1*, либо герметично закрытые материалом, прозрачным для красного излучения. Фотодатчик собран на двухсторонней печатной плате с двухсторонним расположением элементов.

Как уже указывалось выше, описанная основная принципиальная схема реализует только главную функцию *ФД* — определение наличия объекта и выдачу импульса. Однако достаточно часто нужно объединить несколько *ФД*, например, для подсчета количества изделий сразу на нескольких транспортерах. В этом случае *ФД* дополняется еще одной микросхемой драйвера сети *RS485*, как показано на рис. 4.17. Из нескольких *ФД* организуется командно-информационная сеть.

К свободным выводам микроконтроллера *D2* подключаются подтягивающие резисторы *R11* и переключки *JP1*. С помощью этих элементов устанавливается индивидуальный номер *ФД*. Микросхема *D4* преобразует последовательный интерфейс микроконтроллера в интерфейс *RS485*. Диоды *VD3*, *VD4* и резисторы *R12*, *R13* обеспечивают защиту микросхемы *D4*. Показанное дополнение позво-

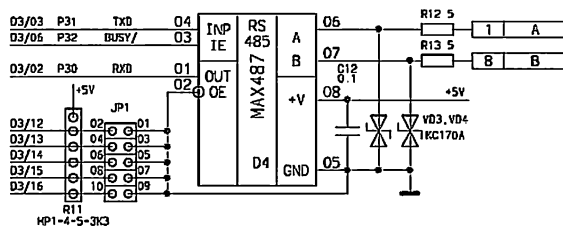


Рис. 4.17. Дополнительные элементы первого расширенного варианта ФД

ляет объединять в сеть до 32 фотодатчиков. В каждом из ФД организуется дополнительный программный счетчик, который считает количество прошедших мимо ФД объектов. По запросу от персонального компьютера содержимое счетчика передается в компьютер, и счетчик обнуляется. Таким образом, компьютер может вести учет изделий, проходящих как по каждому из транспортеров в отдельности, так и по всем транспортерам вместе.

Конструктивное оформление ФД с дополнительными возможностями такое же, как и в предыдущем случае, за исключением того, что его длина составляет 130 мм.

Рассмотрим вариант рабочей программы ФД без сетевой поддержки. Программа написана на языке «С» для микроконтроллеров семейства x51 (IAR).

```

/*****/
/* Photo Controller ver.14 */
/*****/
#include "..\include\io51.h"

/* Секция общих объявлений */
#define byte unsigned char

/* Определения используемых битов */
#define VER 14 /* Версия программы */

#define TESTP P3.2 /* Тестовый выход порога POROG */
#define TEST P3.3 /* Тестовый выход счетчика PTR */
#define SD_OUT P3.4 /* Выход запуска светодиода SD1*/
#define IN_PHOTO P3.5 /* Вход ответа с A1 */
#define WD P3.7 /* Выход перезапуска WDT */
#define OUTPUT P1.3 /* Выходной импульс ФД */
#define PLUS P1.4 /* Кнопка увеличения величины порога */
#define MINUS P1.5 /* Кнопка уменьшения величины порога */
#define RED P1.6 /* Выход управления Red SD2 */
#define GREEN P1.7 /* Выход управления Green SD2 */

#pragma memory = data
byte PTR; /* Счетчик количества импульсов */
byte i; /* Универсальный счетчик */
byte FLAG; /* Флаг наличия объекта FLAG */
byte POROG; /* Переменный уровень */
int StepOut; /* Шаг выходного импульса */
int StepKey; /* Шаг опроса кнопок */
#pragma memory = default
    
```

```

/*****/
void WDI (void)      {WD=1; WD=0;}      /* Импульс перезапуска WDT */
void Time (int Delay) {while (Delay--) {WDI ();};} /* Задержка */
void Red (void)      {RED=1; GREEN=0;} /* Индикатор красный */
void Green (void)    {GREEN=1; RED=0;} /* Индикатор зеленый */
void Black (void)    {RED=0; GREEN=0;} /* Индикатор выключен */
void Pause (void)    {Time (32);}      /* Пауза между опросами */

void Delay (int Del) /* Большая задержка */
{
byte j;

for (j=0; j<Del; j++)
{Time (2000); WDI ();}
}

/*****/
void StartInit (void) /* Процедура начальной инициализации */
{
TEST=0; /* Обнулить тестовый выход количества импульсов */
TESTP=0; /* Обнулить тестовый выход порога (уровня) */
OUTPUT=0; /* Обнулить выходной сигнал */
FLAG=0; /* Установить Flag =0 */
POROG=5; /* Установить начальное значение порога */
Green (); /* Индикатор зеленый */
Delay (59); /* Задержка примерно 1S */
Red (); /* Индикатор красный */
Delay (59); /* Задержка примерно 1S */
Green (); /* Индикатор зеленый */
Delay (59); /* Задержка примерно 1S */
Black (); /* Индикатор выключить */
IN_PHOTO=1; /* Назначить вход фотоимпульса для чтения */
StepOut=0; /* Обнулить счетчик выходного импульса */
StepKey=0; /* Обнулить счетчик опроса кнопок */
}

void main (void)
{
StartInit (); /* Начальная инициализация */
while (1)
{
PTR=0; /* Обнулить счетчик */

/* ФАЗА ОПРОСА ФОТОДАТЧИКА */
SD_OUT=0; /* Включить светодиод */
for (i=0; i<20; i++) /* Сделать 20 циклов */
if (IN_PHOTO!=0) /* Если на выходе фотоприемника 1 */
PTR++; /* Увеличить счетчик */
SD_OUT=1; /* Выключить светодиод */
WDI ();

/* ФАЗА АНАЛИЗА НАЧАЛА ОБЪЕКТА */
if((PTR>POROG)&(FLAG==0)) /* Если PTR>POROG => новый объект */
{
OUTPUT=1; /* Начать выходной импульс */
FLAG=1; /* Установить флаг нового объекта */
}
}
}

```

```

        StepOut=55;      /* Установить количество шагов выхода */
    }

                                /* ФАЗА ОПРЕДЕЛЕНИЯ КОНЦА ОБЪЕКТА*/
    if (!StepOut)          /* Если количество шагов выхода равно 0 */
    {
        FLAG=0;           /* Сбросить флаг объекта */
        OUTPUT=0;        /* Завершить выходной импульс */
    }

                                /* ФАЗА УСТАНОВКИ ИНДИКАЦИИ */
    if (FLAG==1)
        Red ();           /* Если объект - индикатор красный */
    else
    {
        if (PTR>3)        /* Если отклик больше 3 */
            Green ();     /* Есть фон - индикатор зеленый */
        else
            Black ();     /* Нет фона - индикатор выключен */
    }

                                /* ФАЗА ОПРОСА КНОПОК */
    if (StepKey==0)       /* Если шаг нулевой - опрос разрешен */
    {
        if (PLUS==0)     /* Если нажата кнопка увеличения */
        {
            POROG++;     /* Увеличить порог */
            StepKey=2000; /* Установить интервал опроса */
        }
        if (MINUS==0)    /* Если нажата кнопка уменьшения */
        {
            POROG--;     /* Уменьшить порог */
            StepKey=2000; /* Установить интервал опроса */
        }
        POROG=(POROG>20)?20:POROG; /* Проверить порог на превышение */
        POROG=(POROG<=1)?1:POROG; /* проверить порог на минимум */
    }

                                /* ФАЗА ВЫЧИСЛЕНИЯ ШАГОВ */
    StepOut--;            /* Уменьшить шаг выходного импульса */
    if (StepOut<0)       /* Проверить на минимум */
        StepOut=0;
    StepKey--;           /* Уменьшить шаг опроса кнопок */
    if (StepKey<0)      /* Проверить на минимум */
        StepKey=0;

                                /* ФАЗА ТЕСТОВЫХ СИГНАЛОВ*/
    for (i=0;i<=PTR;i++) /* Вывести количество импульсов опроса */
    { TEST=1; TEST=0; }
    for (i=0;i<=POROG;i++) /* Вывести количество импульсов порога */
    { TESTP=1; TESTP=0; }
}
Pause ();              /* Пауза для формирования требуемой частоты опроса */
}
/* КОНЕЦ ПРОГРАММЫ */

```

Приведенная рабочая программа основной части (без поддержки RS485) первого варианта фотодатчика иллюстрирует основные приемы программирования. Очевидно, что возможны и другие варианты написания программы. Например, в [130] приведена другая схема фотодатчика и соответствующая программа.

Описанные варианты реализации фотодатчика удовлетворяют всем предъявляемым требованиям. Они экономичны, просты в настройке и надежны в эксплуатации. Они позволяют считать до ста объектов в секунду. При этом возможное «дрожание» момента выдачи импульса относительно момента начала объекта ничтожно мало (менее 1 мм).

Приведенная программа занимает всего 587 байт кода (из 2048 возможных). Это позволяет, при необходимости, легко дополнить программу для поддержки сети, а также производить любые другие модификации. Повысить точность определения начала объекта можно, если повысить тактовую частоту микроконтроллера и заменить его, например, на AVR-микроконтроллер AT90S2313. Естественно, необходима будет соответствующая коррекция программы, не изменяющая общий алгоритм.

Второй вариант ФД приведем для иллюстрации использования более современной элементной базы — сверхминиатюрного (3×3 мм) микроконтроллера фирмы *Cygnal*. Принципиальная схема второго варианта фотодатчика показана на рис. 4.18.

Принципиальные схемы довольно похожи, поэтому отметим только основные отличия.

1. В качестве входного усилителя применен современный интегральный быстродействующий фотоприемник TSOP7000 (A1) фирмы Vishay [131], работающий при частотах до 255 кГц. Фотоприемник по структуре аналогичен описанной выше микросхеме *TBA2800*, но не имеет выходов промежуточных каскадов и не нуждается в дополнительных внешних элементах.

2. Микроконтроллер C8051F300 работает от внутреннего прецизионного генератора на частоте 24 МГц, кроме того, за счет конвейерной организации ядра, выполняет большинство инструкций программы за 25 нс. Это позволяет значительно увеличить точность подсчета длительности фотоимпульса и увеличить

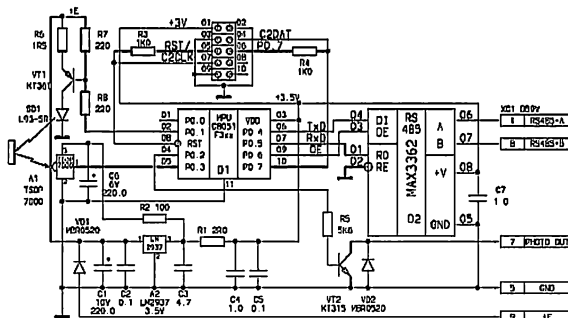


Рис. 4.18. Второй вариант фотодатчика на микроконтроллере фирмы *Cygnal*

частоту импульсов опроса примерно до 240 кГц, что, в свою очередь, увеличивает точность распознавания объекта.

3. Микроконтроллер имеет режим внутрисистемного программирования интерфейса С2, что значительно упрощает настройку и отладку изделия.

4. Индивидуальный адрес контроллера хранится в Flash-памяти и задается при программировании.

5. Все микросхемы работают от напряжения питания 3,5 В, поэтому в схему добавлена микросхема маломощного стабилизатора А2.

6. Потребление всех микросхем при выключенном светодиоде SD1 составляет всего 10 мА.

7. Печатная плата фотодатчика выполнена по технологии поверхностного монтажа и имеет значительно меньшие размеры: 10 × 35 мм.

Последний вариант фотодатчика иллюстрирует основные достоинства современной микроконтроллерной элементной базы.

4.7. Базовая программа системы малой автоматизации

В этом разделе мы ознакомимся с «базовой программой» универсального технологического контроллера, ориентированного на работу в составе командно-информационной сети (т. е. контроллера системы малой автоматизации). Термин «базовая программа» означает, что рассматриваемая программа поддерживает функционирование основных узлов современного технологического контроллера. К основным узлам относятся: последовательный интерфейс (*RS485* или *RS232C*), ориентированный на работу в командно-информационной сети, символьный жидкокристаллический индикатор *LCD* и таймер реального времени *RTC*. Перечисленные узлы управляются микроконтроллером C8051F005 фирмы *Cygnal*. Принципиальная схема рассматриваемых узлов универсального технологического контроллера приведена на рис. 4.19.

Принципиальная схема содержит: микроконтроллер D1 (C8051F005), интерфейс RS232C на микросхеме D2 (MAX232), интерфейс RS485 на микросхеме D3 (MAX487), таймер реального времени D4 (DS12997). Аналоговые входы/выходы микроконтроллера подключены к штыревому разъему J1 (PLD-16). Линии сигналов программирования интерфейса JTAG выведены на разъем J2 (PLD-10). Микросхемы последовательных интерфейсов RS485 (D3) или RS232C (D2) подключаются к микроконтроллеру через перемычки, устанавливаемые на штыревом разъеме J3 (PLD-10). Перемычка J4 служит для подключения согласующего резистора R10 в случае работы контроллера на конце сети RS485. Сигнал передачи TxD микроконтроллера выводится на линию порта P0.4. Сигнал приема RxD микроконтроллера выводится на линию порта P0.5. Для управления передатчиком интерфейса RS485 используется линия порта P1.5. К линии порта P1.4 подключен BUZZER (малогабаритный автогенерирующий звуковой излучатель). Интерфейс LCD выведен на разъем J5 (PLD-34). Перемычка J5 предназначена для выбора полярности напряжения контрастности LCD, причем в качестве отрицательного напряжения используется напряжение от встроенного пре-

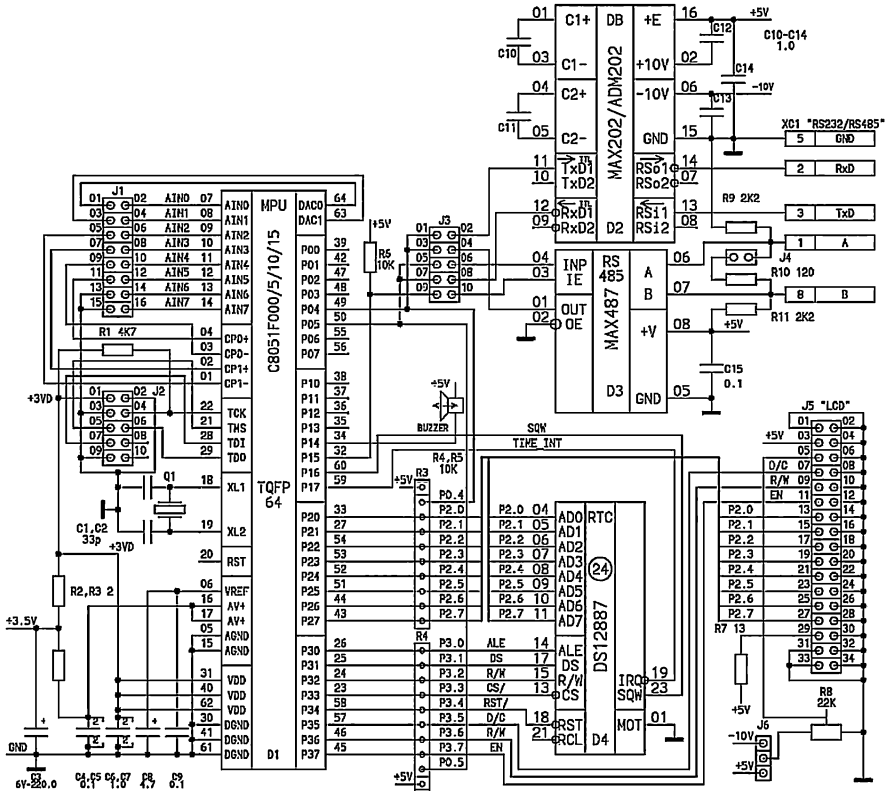


Рис. 4.19. Принципиальная схема рассматриваемых узлов универсального технологического контроллера

образователя микросхемы MAX232 (D2/06). Резистор R8 предназначен для регулировки контрастности LCD. В качестве шины данных для работы RTC и LCD используется порт P2 микроконтроллера. Линии порта P3 используются для формирования сигналов управления RTC и LCD. Поскольку RTC, LCD и интерфейсные микросхемы RS232 и RS485 используют напряжение питания +5 В, а микроконтроллер работает от напряжения +3,5 В, для согласования выходных уровней микроконтроллера используются подтягивающие резисторы R4—R6.

Базовая программа написана на языке C (Keil) и содержит 4 модуля:

MAIN.C — главный модуль;

LCD.C — модуль управления жидкокристаллическим индикатором LCD;

RTC.C — модуль управления таймером реального времени RTC;

UART.C — модуль управления интерфейсами последовательного порта.

Кроме того, каждый из этих модулей имеет файл заголовков: MAIN.H, LCD.H, RTC.H и UART.H.

Рассмотрим каждый из этих программных модулей.

4.7.1. Модуль управления жидкокристаллическим индикатором LCD

Прежде чем приступить к рассмотрению собственно программного модуля, приведем некоторые сведения об программном интерфейсе LCD. В рассматриваемом примере мы использовали LCD-индикатор SC2004ASLB-SH-H-QS, содержащий четыре строки по 20 символов с английским и русским шрифтами и кодовой таблицей ЕН. Модуль имеет габаритные размеры 98 × 60 × 8,8 мм. Модуль имеет стандартизованный разъем, соответствующий варианту L2462 (J5) и стандартный интерфейс управления для встроенного в LCD микроконтроллера HD44780 и его аналогов. Стандартный набор инструкций для управления LCD с указанным микроконтроллером приведен в табл. 4.22.

Таблица 4.22

Инструкция	Коды										Описание	Время выполнения
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Очистка дисплея и установка курсора на адрес 0	1,64 мс
Cursor At Home	0	0	0	0	0	0	0	0	1	X	Установка курсора на адрес 0	1,64 мс
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Установка направления движения курсора и включение сдвига дисплея	40 мкс
Display On/Off Control	0	0	0	0	0	0	1	D	C	B	Вкл./выключение дисплея (D), курсора (C) и режима мигания (B)	40 мкс
Cursor/Display Shift	0	0	0	0	0	1	S/C	R/L	X	X	Сдвиг курсора и дисплея	40 мкс
Function Set	0	0	0	0	1	DL	N	F	X	X	Установка разрядности интерфейса, длины строки и шрифта	40 мкс
CGRAM Address Sel	0	0	0	1	ACG					Установки адреса CGRAM	40 мкс	
DDRAM Address Sel	0	0	1	ADD					Установки адреса DDRAM	40 мкс		
Busy Flag / Address Read	0	1	BF	AC					Чтение флага занятости BF и адреса	40 мкс		
CGRAM/DDRAM Data Write	1	0	WRITE DATA					Запись данных	40 мкс			
CGRAM/DDRAM Data Read	1	1	READ DATA					Чтение данных	40 мкс			

В приведенной таблице использованы следующие сокращения и обозначения:

DDRAM — Display Data RAM — оперативная память данных дисплея;

CGRAM — Character Generator RAM — оперативная память знакогенератора;

ADD — DDRAM Address — адрес оперативной памяти данных дисплея;

ACG — CGRAM Address — адрес оперативной памяти данных знакогенератора;

AC — Address Counter — счетчик адреса;

X — значение бита игнорируется;

I/D — Increment/Decrement — увеличение (1)/уменьшение (0) адреса курсора;

S — Shift — сдвиг дисплея включен (1)/выключен (0);

S/C — (0) — сдвиг курсора/(1) сдвиг дисплея;

R/L — Right/Left — сдвиг (1) вправо/(0) влево;

DL — (1) — интерфейс восьмибитный/(0) четырехбитный;

N — (1) 2 строки/(0) 1 строка;

F — Font — Шрифт (1) 5 × 10 точек/(0) 5 × 7 точек;

BF — Busy Flag — контроллер LCD (1) занят/(0) свободен.

Приступим к рассмотрению программного модуля LCD.H. Он содержит объявления всех функций, которые реализованы в модуле LCD.C. Модули LCD содержат только самые необходимые подпрограммы управления LCD-дисплеем.

```

/*-----
LCD.H
LCD Prototypes
-----*/
#ifndef __LCD__ // Условие для исключения повторной компиляции модуля
#define __LCD__

#ifndef __BYTE__
#define __BYTE__
typedef unsigned char byte; // Определение типа переменной BYTE
#endif

#pragma SAVE
#pragma REGPARMS

void LCD_Init (void); // Инициализация LCD
void LCD_Put (byte DB); // Запись данных в LCD
void LCD_SetCom (byte DB); // Запись команды в LCD
void LCD_Wait (void); // Ожидание готовности LCD
void LCD_Reset (void); // Сброс CLD
void LCD_Clear (void); // Очистка дисплея и установка курсора в адрес 0
void LCD_Home (void); // Установка курсора в адрес 0
void LCD_ON (void); // Дисплей включить
void LCD_OFF (void); // Дисплей выключить
void LCD_AddrStr (byte X, byte Y); // Установить адрес
byte LCD_ReCode (char CH); // Подпрограмма перекодировки в стандарт MS-DOS
void LCD_WriteData (byte DAT); // Запись символа, минуя перекодировку
void LCD_WriteCData (byte DAT);
byte LCD_ReadData (void); // Запись символа с перекодировкой
byte LCD_AddrGet (void); // Получить памяти символов
void LCD_String (char *s); // Вывести строку по ранее установленному адресу
void LCD_XYString (byte STR, byte NUM, char *s); // Вывести строку по адресу
// номер строки (0-3), номер символа от 0

#pragma RESTORE
#endif // LCD

```

В модуле LCD.C содержатся собственно подпрограммы, объявленные в модуле LCD.H.

```

/*-----
LCD.C
LCD Functions
-----*/
#include <Main.h> // Включение главного модуля необходимо, т. к. в нем
                // определяются используемые аппаратные ресурсы
                // Таблица перекодировки в режим MS-DOS
code byte  LCDCODE[125]=
{0xdb,0xd1,0xdc,0xd9,0x41,0xA0,0x42,0xa1,
    0xe0,0x45,0xA3,0xA4,0xa5,0xa6,0x4b,0xa7,
    0x4d,0x48,0x4f,0xa8,0x50,0x43,0x54,0xa9,
    0xaa,0x58,0xe1,0xab,0xac,0xe2,0xad,0xae,
    0x62,0xaf,0xb0,0xb1,0xb2,0xb3,0xb4,
    0xe3,0x65,0xb6,0xb7,0xb8,0xb9,0xba,0xbb,
    0xbc,0xbd,0xbf,0xbe,0x70,0x63,0xbf,0x79,
    0xe4,0x78,0xe5,0xc0,0xc1,0xe6,0xc2,0xc3,
    0xc4,0xc5,0xc6,0xc7};

void LCD_Init (void)
{
    LCD_EN=0;        // Установить начальные значения сигналов EN, DC, RW
    LCD_DC=0;
    LCD_RW=0;
    Time(250);      // Задержка больше 40 ms
    Time(250);
    LCD_Put (0x38); // Инструкция 0x20 + 2 линии + интерфейс 8 бит + курсор
    Time(200);     // Задержка более 16 ms
    LCD_Put (0x38); // Повторить установки
    Time(100);    // Задержка более 10 ms
    LCD_Put (0x38); // Повторить установки
    Time(100);    // Задержка более 10 ms
    LCD_Reset();  // Сбросить LCD
}

void LCD_Put (byte DB) // Запись данных в LCD
{
    LCD_DC=0;
    LCD_RW=0;
    LCD_EN=1;
    Set_Data (DB);  // Установить данные на шину
    Time (40);
    LCD_EN=0;
}

void LCD_SetCom (byte DB) // Запись команды в LCD
{
    LCD_EN=1;
    LCD_DC=0;
    LCD_RW=0;
    LCD_Put (DB);  // Запись данных в LCD
    LCD_Wait ();   // Ожидать готовность LCD
}

void LCD_Wait (void) // Ожидать готовность LCD

```

```

{
byte CH;

LCD_DC    =0;
LCD_EN    =1;
LCD_RW    =1;
while (1)
{
    CH = Get_Data(); // Опрос готовности
    CH &= 0x80;
    if (CH==0) break;
    WDT ();          // Перезапустить WDT
}
LCD_EN    =0;
Time (40);
LCD_RW    =0;
}

void LCD_Reset (void)
{
    LCD_SetCom (0x38); // Повторить начальные установки
    LCD_SetCom (0x0E); // Инструкция 0x08 + дисплей включить 0x04
                       // + курсор включить 0x02
    LCD_SetCom (0x06); // Инструкция 0x04 + сдвиг вправо 0x02
}

void LCD_Clear (void) {LCD_SetCom(0x01);} // Дисплей очистить
void LCD_Home (void)  {LCD_SetCom(0x02);} // Курсор в адрес 0
void LCD_ON (void)    {LCD_SetCom(0x0C);} // Дисплей включить
void LCD_OFF (void)   {LCD_SetCom(0x08);} // Дисплей выключить

void LCD_AddrStr (byte X, byte Y) // Установить адрес символа
                                   // X - номер столбца начиная от 0
                                   // Y - номер строки начиная от 0
{
byte Offset[4] = {0x00,0x40,0x14,0x54}; // Смещение первых символов строк 0-3

    LCD_SetCom (0x80|Offset[Y]|(X & 0x3F));
}

byte LCD_ReCode (char CH) // Подпрограмма перекодировки в таблицу MS-DOS
{
    if ((CH<0x20)&(CH!=0x0d)) return 0x20;
    else if (CH==0x0d) return 0x7e;
    else if ((CH<0xA7)&(CH<123)) return CH;
    else if ((CH>=123)&(CH<176)) return LCDCODE[CH-124];
    else if ((CH>=224)&(CH<240)) return LCDCODE[CH-172];
    else if (CH>240) return 0x20;
}

void LCD_WriteData (byte DAT) // Запись символа без перекодировки
                              // по текущему адресу
{
    LCD_EN=1;
    LCD_RW=0;
    LCD_DC=1;
}

```

```
    Set_Data (DAT);
    Time (40);
    LCD_EN=0;
    LCD_Wait ();
    LCD_DC=0;
}

void LCD_WriteCData (byte DAT) // Запись символа с перекодировкой
                               // по текущему адресу
{
    LCD_EN=1;
    LCD_RW=0;
    LCD_DC=1;
    Set_Data (LCD_ReCode(DAT));
    Time (40);
    LCD_EN=0;
    LCD_Wait ();
    LCD_DC=0;
}

byte LCD_ReadData (void)      // Чтение данных по предварительно заданному адресу
{
    byte DAT;

    LCD_EN=1;
    LCD_RW=1;
    LCD_DC=1;
    DAT = Get_Data ();
    Time (40);
    LCD_EN=0;
    LCD_Wait ();
    LCD_RW=0;
    LCD_DC=0;
    return DAT;
}

byte LCD_AddrGet (void)      // Чтение текущего адреса
{
    byte DAT;

    LCD_EN    =1;
    LCD_DC    =0;
    LCD_RW    =1;
    DAT=Get_Data ();
    LCD_EN    =0;
    LCD_RW    =0;
    return    DAT&0x7F;
}

void LCD_String (char *s)    // Запись строки по предварительно заданному адресу
{
    int      i;

    for (i=0; i<strlen(s); i++)
        LCD_WriteCData (s[i]);
}
```

```

void LCD_XYString (byte STR, byte NUM, char *s) // Вывести строку по адресу
                                                // номер строки (0-3), номер символа от 0
{
    LCD_AddrStr (NUM,STR); // Установить адрес первого символа
    LCD_String (s);       // Вывести строку
}
// End of LCD.C

```

Следует отметить, что приведенный модуль использует ряд функций из других программных модулей: Set_Data (byte), Get_Data (), Time (byte), WDT (). Естественно, что если в проекте не используется русский шрифт, то подпрограммы, связанные с перекодировкой, и перекодировочную таблицу можно сократить. Также очевидно, что величины задержек зависят от тактовой частоты микроконтроллера (в нашем случае — 11,0592 МГц) и производительности его ядра (в нашем случае одна короткая команда выполняется примерно за 90 нс).

4.7.2. Модуль управления таймером реального времени RTC

В описываемом контроллере применен таймер реального времени DS12887, для программиста доступный в виде массива памяти с объемом 127 байт. Первые четырнадцать байт заняты под специальные регистры, а остальные байты могут быть использованы программистом в качестве энергонезависимой памяти данных. Среди 14 специальных регистров первые десять доступны и для чтения, и для записи. В них автоматически обновляются предварительно установленные значения времени и календаря. Назначение и форма представления информации в первых десяти регистрах представлены в табл. 4.23.

Таблица 4.23

Адрес	Назначение	Десятичный диапазон значений	Диапазон значений данных	
			Двоичный	Двоично-десятичный
0x00	Секунды	00–59	00–3В	00–59
0x01	Секунды будильника	00–59	00–3В	00–59
0x02	Минуты	00–59	00–3В	00–59
0x03	Минуты будильника	00–59	00–3В	00–59
0x04	Часы, режим 12 часов	01–12	01–0С AM, 81–8С PM	01–12 AM, 81–92 PM
	Часы, режим 24 часа	00–23	00–17	00–23
0x05	Часы будильника, 12 часов	01–12	01–0С AM, 81–8С PM	01–12 AM, 81–92 PM
	Часы будильника, 24 часа	00–23	00–17	00–23
0x06	День недели (1 – воскресенье)	01–07	01–07	01–07
0x07	День месяца	01–31	01–1F	01–31
0x08	Месяц	01–12	01–0С	01–12
0x09	Года	00–99	00–63	00–99

Кроме приведенных десяти регистров, имеются еще четыре управляющих регистра: А, В, С и D.

Регистр А имеет адрес 0x0A и содержит следующие биты:

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0

UIP — Update In Progress — постоянно автоматически обновляемый бит завершенности. Если бит равен 1, идет обновление данных. Если бит равен 0, то обновления данных не будет еще как минимум 244 мс, разрешен доступ к регистрам 0x00—0x09. Бит доступен только для чтения. Не зависит от сигнала сброса. Запись 1 в бит SET регистра В запрещает любые изменения состояния бита UIP.

DV2:DV0 — биты управления включением или отключением RTC. Установка битов в состояние 010 запускает RTC. Установка состояния 11X затормаживает счет. Первое обновление значений регистров 0x00—0x09 произойдет не ранее 500 мс после установки комбинации 010.

RS3:RS0 — биты устанавливают временные интервалы периодических прерываний и частоту выходных импульсов на выходе SQW (см. табл. 4.24).

Таблица 4.24

Биты регистра А				Период периодических прерываний PIE	Частота выхода SQW
RS3	RS2	RS1	RS0		
0	0	0	0	Выключено	Выключено
0	0	0	1	3,90625 мс	256 Гц
0	0	1	0	7,8125 мс	128 Гц
0	0	1	1	122,070 мкс	8,192 кГц
0	1	0	0	244,141 мкс	4,096 кГц
0	1	0	1	488,281 мкс	2,048 кГц
0	1	1	0	976,5625 мкс	1,024 кГц
0	1	1	1	1,953125 мс	512 Гц
1	0	0	0	3,90625 мс	256 Гц
1	0	0	1	7,8125 мс	128 Гц
1	0	1	0	15,625 мс	64 Гц
1	0	1	1	31,25 мс	32 Гц
1	1	0	0	62,5 мс	16 Гц
1	1	0	1	125 мс	8 Гц
1	1	1	0	250 мс	4 Гц
1	1	1	1	500 мс	2 Гц

Регистр В имеет адрес 0x0B и содержит следующие биты:

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
SET	PIE	AIE	UIE	SQWE	DM	24/12	DSE

SET — бит установки режима обновления. Если бит установлен в 0, процесс обновления всех величин времени и календаря происходит раз в секунду. Если бит установлен в 1, процесс обновления запрещен и значения можно устанавливать программно. Значение бита не зависит от сброса или от внутренних процессов RTC.

PIE — Periodic-Interrupt Enable (PIE) — бит разрешения периодического прерывания, установки соответствующего флага PF (регистр C) и «отрицательного» импульса на выходе IRQ/. Установка в 0 запрещает периодические прерывания, но флаг PF продолжает периодически устанавливаться. При установке в 1, периодические прерывания генерируются с интервалом, заданным битами RS3—RS0 регистра A (см. табл. 4.24). Значение бита PIE не модифицируется любыми внутренними процессами в RTC, но устанавливается в 0 после сброса.

AIE — Alarm Interrupt Enable — бит разрешения прерывания от будильника. При установке в 1 прерывания от будильника разрешены, а также устанавливается соответствующий флаг AF (регистр C) и вырабатывается «отрицательный» импульс на выходе IRQ/. Прерывание от будильника вырабатывается каждую секунду, если все три байта будильника (0x01, 0x03, 0x05) равны трем соответствующим байтам времени (0x00, 0x02, 0x04) включая и коды «игнорирования» (11xxxxxb). Если бит установлен в 0, взводится бит AF, но прерывание не вырабатывается. Сброс обнуляет бит AIE, но никакие внутренние процессы на него не влияют.

UIE — Update Ended Interrupt — бит разрешения прерывания завершения обновления. Сброс обнуляет бит, а установка бита SET устанавливает бит UIE в 1.

SQWE — Square-Wave Enable — бит разрешения формирования частоты на соответствующем выходе SQWE в соответствии со значениями, установленными битами RS3—RS0 регистра A (см. табл. 4.24). Если бит установлен в 0, состояние на соответствующем выходе также устанавливается в 0. Бит сбрасывается после сброса.

DM — Data Mode — бит установки режима представления данных. Если DM = 0, все величины формируются в двоично-десятичном виде (BCD), если DM = 1, все величины формируются в двоичном виде (BIN). Значение бита не изменяется после сброса и не зависит от внутренних процессов в RTC.

24/12 — 24/12 Control Bit — бит управления режимом счета часов: установка в 1 переключает в 24-часовой режим, а обнуление — в 12-часовой режим. Бит не изменяется от сброса или внутренних процессов RTC.

DSE — Daylight Saving Enable — бит разрешения (1) режима перевода часов на летнее/зимнее время. Переходы производятся в первое воскресенье апреля с 01:59:59AM к 03:00:00AM и в последнее воскресенье октября с 01:59:59AM к 01:00:00AM. Бит не изменяется от сброса или внутренних процессов RTC.

Регистр C имеет адрес 0x0C и содержит следующие биты:

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
IRQF	PF	AF	UF	0	0	0	0

IRQF — Interrupt Request Flag — суммарный флаг прерывания, который устанавливается в соответствии с выражением:

$$\text{IRQF} = \text{PF} \times \text{PIE} + \text{AF} \times \text{AIE} + \text{UF} \times \text{UIE}$$

Все время, пока этот бит находится в состоянии 1, выход IRQ/ установлен в состояние 0. *Важно помнить, что все флаги прерываний сбрасываются после чтения регистра С или сброса!*

PF — Periodic Interrupt Flag — флаг периодического прерывания только для чтения. Устанавливается в 1 независимо от состояния PIE бита. Если биты PIE & PF = 1, устанавливается IRQF-бит и вырабатывается прерывание. Бит сбрасывается в 0 после сброса или чтения регистра С.

AF — Alarm Interrupt Flag — флаг прерывания будильника только для чтения. Устанавливается в 1 независимо от состояния бита AIE. Если биты AIE & AF = 1, устанавливается IRQF-бит и вырабатывается прерывание. Бит сбрасывается в 0 после сброса или чтения регистра С.

UF — Update Ended Interrupt Flag — флаг прерывания после завершения обновления только для чтения. Устанавливается в 1 независимо от состояния бита UIE. Если биты UIE & UF = 1, устанавливается IRQF-бит и вырабатывается прерывание. Бит сбрасывается в 0 после сброса или чтения регистра С.

Биты 3—0 не используются, читаются как 0000b, биты не могут быть записаны программно.

Регистр D имеет адрес 0x0D и содержит следующие биты:

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
VRT	0	0	0	0	0	0	0

VRT — Valid RAM & Time Bit — бит состояния только для чтения. Он всегда должен быть равен 1. Если он равен 0, то встроенная литиевая батарея выработала свой ресурс и, соответственно, данные времени и памяти не соответствуют действительности.

Биты 6—0 не используются, читаются как 0000000b, биты не могут быть записаны программно.

Кроме описания служебных регистров при написании подпрограмм поддержки функций RTC следует помнить следующие моменты.

1. Данные о текущем времени и календаре могут быть получены при чтении соответствующих регистров.

2. Данные о текущем времени и календаре могут быть изменены при записи соответствующих регистров.

3. Данные могут представляться в двоичном или двоично-десятичном виде.

4. Перед записью данных о текущем времени, будильнике и календаре необходимо установить бит SET = 1. Это поможет избежать искажений данных при записи за счет блокирования текущего обновления.

5. Режим представления данных предполагает обязательную переустановку всех данных времени, будильника и календаря.

6. Следует помнить, что если установлен 12-часовой режим, старший бит часов индицирует РМ.

7. При чтении в момент обновления значения часов, минут и секунд могут быть искажены.

8. Если установлены часы, минуты и секунды будильника, прерывание будет вырабатываться в момент совпадения заданного времени каждый день.

9. Если в байт данных (часов, минут или секунд) будильника записан код, содержащий две старшие единицы 11xxxxxb, то такой код называется «незначим».

10. Если «незначимый» код записан в регистр часов будильника, прерывание будильника будет генерироваться каждый час.

11. Если «незначимый» код записан в регистры и часов, и минут будильника, прерывание будильника будет генерироваться каждую минуту.

12. Если «незначимый» код записан в регистры и часов, и минут, и секунд будильника, прерывание будильника будет генерироваться каждую секунду.

Приступим к рассмотрению программного модуля RTC.H. Модуль содержит объявления всех функций, которые реализованы в модуле RTC.C. Модули RTC содержат только самые необходимые подпрограммы управления таймером реального времени — RTC.

```

/*-----
RTC.H
RTC Prototypes
-----*/
#ifndef __RTC__ // Условие для исключения повторной компиляции модуля
#define __RTC__

#ifndef __BYTE__
#define __BYTE__
typedef unsigned char byte; // Определение типа переменной BYTE
#endif

// Определения адресов специальных регистров
#define RTC_SEC 0x00 // Регистр секунд
#define RTC_ASEC 0x01 // Регистр секунд будильника
#define RTC_MIN 0x02 // Регистр минут
#define RTC_AMIN 0x03 // Регистр минут будильника
#define RTC_HAUSE 0x04 // Регистр часов
#define RTC_AHAUSE 0x05 // Регистр часов будильника
#define RTC_WDAY 0x06 // Регистр дня недели
#define RTC_MDAY 0x07 // Регистр дня месяца
#define RTC_MOUNTH 0x08 // Регистр номера месяца
#define RTC_YEAR 0x09 // Регистр номера года
#define RTC_RGA 0x0A // Регистр A
#define RTC_RGB 0x0B // Регистр B
#define RTC_RGC 0x0C // Регистр C
#define RTC_RGD 0x0D // Регистр D

// Проверка валидности литиевой батареи
#define RTC_Valid() RTC_Read(RTC_RGD)&0x80

// Чтение регистра C наличия прерывания
#define RTC_Int() RTC_Read(RTC_RGC)
// Чтение регистра B

```

```
#define RTC_Mode()      RTC_Read(RTC_RGB)
                        // Разрешение обновлений
#define RTC_Set1()     RTC_Write(RTC_RGB, RTC_Mode())|0x80
                        // Запрещение обновлений
#define RTC_Set0()     RTC_Write(RTC_RGB, RTC_Mode())&0x7F
                        // Разрешение периодических прерываний
#define RTC_PIE1()    RTC_Write(RTC_RGB, RTC_Mode())|0x40
                        // Запрещение периодических прерываний
#define RTC_PIE0()    RTC_Write(RTC_RGB, RTC_Mode())&0xBF
                        // Разрешение прерываний будильника
#define RTC_AIE1()    RTC_Write(RTC_RGB, RTC_Mode())|0x20
                        // Запрещение прерываний будильника
#define RTC_AIE0()    RTC_Write(RTC_RGB, RTC_Mode())&0xDF
                        // Разрешение прерываний обновления
#define RTC_UIE1()    RTC_Write(RTC_RGB, RTC_Mode())|0x10
                        // Запрещение прерываний обновления
#define RTC_UIE0()    RTC_Write(RTC_RGB, RTC_Mode())&0xEF
                        // Разрешение генерации импульсов
#define RTC_SQWE1()   RTC_Write(RTC_RGB, RTC_Mode())|0x08
                        // Запрещение генерации импульсов
#define RTC_SQWE0()   RTC_Write(RTC_RGB, RTC_Mode())&0xF7
                        // Установка бинарного режима
#define RTC_DM1()     RTC_Write(RTC_RGB, RTC_Mode())|0x04
                        // Установка двоично-десятичного режима
#define RTC_DM0()     RTC_Write(RTC_RGB, RTC_Mode())&0xFB
                        // Установка 24-часового режима
#define RTC_24()      RTC_Write(RTC_RGB, RTC_Mode())|0x02
                        // Установка 12-часового режима
#define RTC_12()      RTC_Write(RTC_RGB, RTC_Mode())&0xFD
                        // Разрешение режима коррекции
#define RTC_DSE0()    RTC_Write(RTC_RGB, RTC_Mode())|0x01
                        // Запрещение режима коррекции
#define RTC_DSE1()    RTC_Write(RTC_RGB, RTC_Mode())&0xFE
                        // Чтение регистра A статуса
#define RTC_Stat()    RTC_Read(RTC_RGA)
                        // Чтение готовности RTC
#define RTC_Ready()   RTC_Stat()&0x80
                        // Включение RTC
#define RTC_ON()      RTC_Write(RTC_RGA, RTC_Stat())&0x8F|0x20
                        // Выключение RTC
#define RTC_OFF()     RTC_Write (RTC_RGA, RTC_Stat())|0x70
                        // Установка периодичности прерываний и частоты импульсов
#define RTC_Rate(C)   RTC_Write(RTC_RGA, RTC_Stat())&0xF0|C&0xF0
                        // Чтение секунд
#define RTC_GetSec()  RTC_Read(RTC_SEC)
                        // Запись секунд
#define RTC_SetSec(DATA) RTC_Write(RTC_SEC, DATA)
                        // Чтение минут
#define RTC_GetMin()  RTC_Read(RTC_MIN)
                        // Запись минут
#define RTC_SetMin(DATA) RTC_Write(RTC_MIN, DATA)
                        // Чтение часов
```

```

#define RTC_GetHause() RTC_Read(RTC_HAUSE)
    // Запись часов
#define RTC_SetHause(DATA) RTC_Write(RTC_HAUSE, DATA)
    // Чтение дня недели
#define RTC_GetWDay() RTC_Read(RTC_WDAY)
    // Запись дня недели
#define RTC_SetWDay(DATA) RTC_Write(RTC_WDAY, DATA)
    // Чтение дня месяца
#define RTC_GetMDay() RTC_Read(RTC_MDAY)
    // Запись дня месяца
#define RTC_SetMDay(DATA) RTC_Write(RTC_MDAY, DATA)
    // Чтение месяца
#define RTC_GetMounth() RTC_Read(RTC_MOUNTH)
    // Запись месяца
#define RTC_SetMounth(DATA) RTC_Write(RTC_MOUNTH, DATA)
    // Чтение года
#define RTC_GetYear() RTC_Read(RTC_YEAR)
    // Запись года
#define RTC_SetYear(DATA) RTC_Write(RTC_YEAR, DATA)
    // Чтение секунд будильника
#define RTC_GetASec() RTC_Read(RTC_ASEC)
    // Запись секунд будильника
#define RTC_SetASec(DATA) RTC_Write(RTC_ASEC, DATA)
    // Чтение минут будильника
#define RTC_GetAMin() RTC_Read(RTC_AMIN)
    // Запись минут будильника
#define RTC_SetAMin(DATA) RTC_Write(RTC_AMIN, DATA)
    // Чтение часов будильника
#define RTC_GetAHause() RTC_Read(RTC_AHAUSE)
    // Запись часов будильника
#define RTC_SetAHause(DATA) RTC_Write(RTC_AHAUSE, DATA)

    // Начальная инициализация RTC
void RTC_Init
    (void);
    // Чтение данных из RTC
byte RTC_Read
    (byte ADDR);
    // Запись данных в RTC
void RTC_Write
    (byte ADDR, byte DATA);
    // Чтение структуры календаря
void Get_Date
    (void);
    // Чтение структуры времени
void Get_Time
    (void);
    // Формирование сигнала ALE для RTC
void RTC_Ale
    (void);
#endif // RTC

```

В модуле RTC.C содержатся собственно подпрограммы, объявленные в модуле RTC.H.

```

/*-----
// RTC.C
// RTC Functions
-----*/

```

```

#include <Main.h>           // Включение главного модуля необходимо, т. к. в нем
                           // определяются используемые аппаратные ресурсы

                           // Структура данных календаря объявляется в
                           // дополнительной оперативной памяти микропроцессора

xdata struct date
{
    byte year;             // Текущий год
    byte mounth;          // Месяц
    byte day;              // День месяца
    byte wday;            // День недели
} da;

                           // Структура данных времени объявляется в
                           // дополнительной оперативной памяти микропроцессора

xdata struct time
{
    byte hause;           // Час
    byte min;             // Минуты
    byte sec;             // Секунды
} ti;

                           // Структура данных будильника объявляется в
                           // дополнительной оперативной памяти микропроцессора

xdata struct alarm
{
    byte ahause;          // Час будильника
    byte amin;           // Минуты будильника
    byte asec;           // Секунды будильника
} al;

void RTC_Init (void)      // Инициализация RTC
{
    RTC_ALE=0;           // Сбросить ALE
    RTC_CS=1;            // Установить CS/
    RTC_DS=1;            // Установить RD/
    RTC_RW=1;           // Установить WR/
    RTC_RST=0;          // Установить RST
    Time (10);           // Задержка на длительность сигнала RST
    RTC_RST=1;          // Сбросить RST
}

static void RTC_Ale (void) // Генерация импульса ALE
{
    Time (2);            // Задержка
    RTC_ALE=1;          // Установить ALE
    Time (12);           // Задержка на длительность сигнала ALE
    RTC_ALE=0;          // Сбросить ALE
    Time (5);            // Задержка
}

byte RTC_Read (byte ADDR) // Чтение данных из RTC
{
byte CH;

    EA=0;                // Прерывания запретить
    RTC_CS=0;            // Активизировать CS/

```

```

Set_Data (ADDR); // Задать адрес
RTC_Ale (); // Сгенерировать ALE
RTC_DS=0; // Активизировать RD/
Time (15); // Задержка
CH=Get_Data (); // Получить данные
Time (15); // Задержка
RTC_DS=1; // Снять RD/
RTC_CS=1; // Снять CS/
EA=1; // Разрешить прерывания
return CH; // Вернуть прочитанные данные
}

void RTC_Write (byte ADDR, byte DATA) // Запись данных в RTC
{
    EA=0; // Запретить прерывания
    RTC_CS=0; // Активизировать CS/
    Set_Data (ADDR); // Установить адрес
    RTC_Ale (); // Сгенерировать ALE
    Set_Data (DATA); // Установить данные на шину
    RTC_RW=0; // Активизировать WR/
    Time (15); // Задержка
    RTC_RW=1; // Снять WR/

    RTC_CS=1; // Снять CS/
    EA=1; // Разрешить прерывания
}

void Get_Date (void) // Обновить структуру календаря
{
    while (RTC_Ready ()); // Ожидать готовности
    da.year=RTC_GetYear(); // Читать год
    da.mounth=RTC_GetMounth(); // Читать месяц
    da.day=RTC_GetMDay(); // Читать день месяца
    da.wday=RTC_GetWDay(); // Читать день недели
}

void Get_Time (void) // Обновить структуру времени
{
    while (RTC_Ready ()); // Ожидать готовности
    ti.hause=RTC_GetHause(); // Читать часы
    ti.min=RTC_GetMin(); // Читать минуты
    ti.sec=RTC_GetSec(); // Читать секунды
}

// End of RTC.C

```

4.7.3. Модуль обслуживания последовательных портов UART

Приведенный ниже модуль обслуживания последовательных портов предназначен для работы в составе командно-информационной сети. Модуль ориентирован на работу с интерфейсом RS485, однако он работает и с интерфейсом RS232C (при этом один сигнал — управления передатчиком) не используется.

Приступим к рассмотрению программного модуля UART.H. Модуль содержит объявления всех функций, которые реализованы в модуле UART.C. Модули UART содержат только самые необходимые подпрограммы управления сетевым интерфейсом CI-LAN.

```

/*-----
UART.H
UART Prototypes
-----*/
#include <Main.h> // Включение главного модуля необходимо, т. к. в нем
// определяются используемые аппаратные ресурсы

#ifndef _UART_ // Условие для исключения повторной компиляции модуля
#define _UART_

#ifndef _BYTE_
#define _BYTE_
typedef unsigned char byte; // Определение типа переменной BYTE
#endif

#define SYSCLK 11059200 // Частота кварцевого резонатора
#define BAUDRATE 115200 // Скорость передачи данных

void SendB (byte SendByte); // Переслать байт
byte GetB (void); // Принять байт
void Get_Frame (void); // Принять фрейм
void Send_Frame (byte CODE, byte DATA1, byte DATA2); // Передать фрейм ответа
byte Input_CS (byte NUM); // Посчитать контр. сумму входного фрейма
byte Output_CS (byte NUM); // Посчитать контр. сумму выходного фрейма

#endif // _UART_

```

В модуле UART.C содержатся собственно подпрограммы, объявленные в модуле UART.H.

```

/*-----
// UART.C
// CI-LAN "MISNET"
-----*/
#include <Main.h> // Включение главного модуля необходимо, т. к. в нем
// определяются используемые аппаратные ресурсы

// Внешние переменные
extern code byte ADDR; // Адрес контроллера
extern data byte IFrame[7]; // Массив входного фрейма
extern data byte OFrame[7]; // Массив выходного фрейма
extern data byte PTR; // Универсальный указатель
extern data byte Byte; // Входной байт
extern data byte FRAME_PTR; // Указатель фрейма
extern data byte FRAME_RDY; // Флаг готовности фрейма

// Структура входного командного фрейма содержит 6 байт
// Байт 0 (IFrame[0]) - преамбула, должен быть равен 0xAA
// Байт 1 (IFrame[1]) - адрес
// Байт 2 (IFrame[2]) - команда
// Байт 3 (IFrame[3]) - первый байт данных
// Байт 4 (IFrame[4]) - второй байт данных

```



```

// Байт 5 (IFrame[5]) - контрольная сумма
// Структура выходного ответного фрейма содержит 6 байт
// Байт 0 (OFrame[0]) - преамбула, должен быть равен 0xAA
// Байт 1 (OFrame[1]) - адрес
// Байт 2 (OFrame[2]) - ошибка или статус
// Байт 3 (OFrame[3]) - первый байт данных
// Байт 4 (OFrame[4]) - второй байт данных
// Байт 5 (OFrame[5]) - контрольная сумма

/*****/
void SendB (byte SendByte) // Переслать байт
{
#иf WORK
    while (!TI) WDT(); // Ожидать отправки предыдущего байта
#endif
    TI=0; // Очистить флаг передачи
    Time (1); // Задержка
    SBUF=SendByte; // Записать байт в буфер передачи
}

/*****/
byte GetB (void) // Получить байт
{
byte InputByte;
    while (!RI) {WDT();} // Ожидать завершения приема
    InputByte = SBUF; // Взять байт из буфера приема
    RI = 0; // Сбросить флаг приема
    return InputByte; // Вернуть принятый байт
}

/*****/
void Get_Frame (void) // Получить фрейм
{
Byte=GetB(); // Получить байт
иf (!FRAME_PTR) // Если PTR==0 & Byte!=0xAA, ожидать следующий байт
{
    иf (Byte==0xAA) goto Next;
    else return;
}
Next:
IFrame[FRAME_PTR++]=Byte; // Записать байт во входной буфер фрейма
// и увеличить указатель фрейма
иf (FRAME_PTR>=6) // Если конечная длина фрейма достигнута
{
    ES=0; // Запретить прерывания последовательного порта
    FRAME_RDY=1; // Установить флаг завершения приема фрейма
    TI=1; // Установить флаг передачи
}
}

/*****/
void UARTO_INT (void) interrupt 4 // Прерывание от последовательного порта
{
    Get_Frame (); // Получить фрейм
}

```

```

}

/*****/
byte Input_CS (byte NUM)    // Подсчитать контрольную сумму входного фрейма
{
byte CS=0;
  for (PTR=0; PTR<NUM; PTR++)
    CS=(CS+IFrame[PTR])&0x0FF;
  return CS;
}

/*****/
byte Output_CS (byte NUM)  // Подсчитать контрольную сумму выходного фрейма
{
byte CS=0;
  for (PTR=0; PTR<NUM; PTR++)
    CS=(CS+OFrame[PTR])&0x0FF;
  return CS;
}

/*****/
void Send_Frame (byte CODE, byte DATA1, byte DATA2) // Послать фрейм
{
  OFrame[0]=0xAA;          // Записать в буфер байт преамбулы
  OFrame[1]=ADDR;         // Записать в буфер байт адреса станции
  OFrame[2]=CODE;         // Записать в буфер байт ошибки или статус
  OFrame[3]=DATA1;        // Записать в буфер первый байт данных
  OFrame[4]=DATA2;        // Записать в буфер второй байт данных
  OFrame[5]=Output_CS(5); // Записать в буфер байт контрольной суммы
  Time (250);              // Задержка
  ON485=1;                 // Открыть передатчик (только для RS485)
  Time (20);               // Задержка

  for (PTR=0; PTR<6; PTR++) // Отправить 6 байт фрейма
  {
    Time (250);            // Задержка
    SendB(OFrame[PTR]);    // Отправить очередной байт
  }

#if WORK
  while (!TI) WDT();      // Ожидать завершение отправки последнего байта
#endif
  Time (20);              // Задержка
  ON485=0;                // Закрыть передатчик (только для RS485)
  TI=0;                   // Очистить флаг передачи
}

// End of UART.C

```

4.7.4. Главный программный модуль MAIN

Главный программный модуль содержит собственно программу. Как уже упоминалось, контроллер содержит четырехстрочный LCD-дисплей. На первую строку выводится сперва заголовок, а затем текущие сведения календаря. На вторую строку выводится счетчик циклов и текущее время. На третью строку

выводится статус или сообщения об ошибках. На четвертую строку выводится температура кристалла от встроенного датчика температуры.

```

/*-----
// Main.H
// Test program for Kit005 (UART+LCD+TEMP+ХМЕМ)
-----*/

#ifndef __MAIN__
#define __MAIN__

#include <C8051F000.h> // Файл определения регистров микропроцессора
#include <stdio.h>
#include <string.h>

#include <LCD.H> // Файл описаний модуля LCD
#include <UART.H> // Файл описаний модуля UART
#include <RTC.H> // Файл описаний модуля RTC

#ifndef __BYTE__
#define __BYTE__
typedef unsigned char byte; // Определение типа переменной BYTE
#endif

#define FALSE 0
#define TRUE 1
#define WORK 1
#define VER 08

#define SYSCLK 11059200 // Частота кварцевого резонатора
#define BAUDRATE 115200 // Скорость передачи данных по
// последовательному порту

#define SAMPLE_RATE 50000 // Частота опроса в Hz
#define INT_DEC 256
#define LCD_LEN 19 // Длина строки индикатора

/*****
// Аппаратно зависимое определение битов и байтов, зависящее от типа
// микроконтроллера и настройка коммутатора ресурсов Crossbar
*****/

#define DATABUS P2 // Порт шины данных
#define DATABUS_CF PRT2CF // Режим порта шины данных

sbit RTC_ALE = P3^0; // Линия P3.0 сигнала ALE для RTC
sbit RTC_DS = P3^1; // Линия P3.1 сигнала DS для RTC
sbit RTC_RW = P3^2; // Линия P3.2 сигнала RW для RTC
sbit RTC_CS = P3^3; // Линия P3.3 сигнала CS для RTC
sbit RTC_RST = P3^4; // Линия P3.4 сигнала RST для RTC
sbit LCD_DC = P3^5; // Линия P3.5 сигнала DC для LCD
sbit LCD_RW = P3^6; // Линия P3.6 сигнала RW для LCD
sbit LCD_EN = P3^7; // Линия P3.7 сигнала EN для LCD

sbit ON485 = P1^5; // Линия P1.5 сигнала ON для RS485
sbit BUZZER = P1^4; // Линия P1.4 подключения буззера

/*****
// Определение шестнадцатитбитных регистров
*****/

```

```

sfr16 DP = 0x82; // Указатель данных
sfr16 TMR3RLL = 0x92; // Величина перезагрузки таймера 3
sfr16 TMR3 = 0x94; // Счетчик таймера 3
sfr16 ADC0 = 0xbe; // Данные ADC0
sfr16 ADC0GT = 0xc4; // Верхний порог ADC0
sfr16 ADC0LT = 0xc6; // Нижний порог ADC0
sfr16 RCAP2 = 0xca; // Величина перезагрузки таймера 2
sfr16 T2 = 0xcc; // Счетчик таймера 2
sfr16 DAC0 = 0xd2; // Данные DAC0
sfr16 DAC1 = 0xd5; // Данные DAC1

/*****/
// Прототипы функций
/*****/

byte Get_Data (void); // Получить данные с шины
void Set_Data (byte CH); // Установить данные на шину
void Time (unsigned Delay); // Малая задержка
void Delay (char Step); // Большая задержка
void WDT (void); // Перезапуск WDT
void SYSCLK_Init (void); // Инициализация тактового генератора
void PORT_Init (void); // Инициализация портов ввода/вывода
void UART0_Init (void); // Инициализация порта UART
void ADC0_Init (void); // Инициализация ADC0
void Timer3_Init (int counts); // Инициализация таймера 3
void ADC0_ISR (void); // Прерывания ADC0
void Get_Temperature (void); // Вычислить температуру

void XMEM_CLEAR (void); // Очистить массив доп. Памяти XRAM
void XMEM_WRITE (unsigned int PTR, byte DATA); // Записать в XRAM
byte XMEM_READ (unsigned int PTR); // Прочитать XRAM

#endif // Main.h

```

И наконец рассмотрим главный программный модуль MAIN.C.

```

/*-----
// Main.C
// Test program for KIT005 (UART+LCD+TEMP+XMEM)
-----*/
#include <Main.h> // Включение главного модуля необходимо, т. к. в нем
// определяются используемые аппаратные ресурсы

code byte ADDR = 0x01; // Индивидуальный адрес станции

// Определение адреса встроенной дополнительной памяти
unsigned char xdata xdata_buffer [256] _at_ 0x8000;

// Массив строк дней недели
byte RTCD[7][4] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};
// Массив строк названий месяца
byte RTCM[12][4] = {"Jan", "Feb", "Mar", "Apr", "May", "Jun",
"Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};

// Определения переменных
data long REZ; // Десятичный результат ADC0
data byte IFrame[7]; // Входной командный фрейм

```

```

data byte  OFrame[7]; // Выходной фрейм ответа
data byte  Buffer[22]; // Буфер строки
data byte  PTR;       // Универсальный указатель
data byte  Byte;      // Входной байт
data byte  FRAME_PTR; // Указатель фрейма
data byte  FRAME_RDY; // Флаг готовности фрейма
data unsigned long COUNT; // Счетчик циклов
data int   TEMP_INT;  // Целая часть значения температуры
data int   TEMP_FRAC; // Дробная часть значения температуры

data bit   FLAG;     // Флаг разрешения опроса температуры
data long  TEMPER;   // Значение температуры в сотых градуса

extern xdata struct date // Структура календаря
{
    byte year;
    byte month;
    byte day;
    byte wday;
} da;

extern xdata struct time // Структура времени
{
    byte hause;
    byte min;
    byte sec;
} ti;

extern xdata struct alarm // Структура будильника
{
    byte ahause;
    byte amin;
    byte asec;
} al;

/*****/
// Функции
/*****/

void Time (unsigned Delay) // Маленькая задержка
{
    while (Delay--)
        WDT();
}

/*****/
void WDT (void) // Перезапуск WDT
{
    WDTCN = 0xA5;
}

/*****/
void Beep (unsigned Delay) // Регулируемая длительность BUZZER
{
    while (Delay--)
    {
        BUZZER=0;
    }
}

```

```

        Time (50);
        BUZZER=1;
    }
}

/*****/
byte Get_Data (void)      // Получение данных с программной шины (P2)
{
    DATABUS_CF=0;        // Данные на ввод
    DATABUS=0xFF;
    return DATABUS;      // Данные получить
}

/*****/
void Set_Data (byte CH)   // Установка данных на программную шину (P2)
{
    DATABUS_CF=0x00;     // Шину в режим с "открытым истоком" для
                        // согласования с 5-вольтовым питанием
    DATABUS=CH;          // Установить данные
}

/*****/
void XMEM_CLEAR (void)   // Заполнить массив XRAM данными /0
{
    memset (xdata_buffer, '\0', sizeof (xdata_buffer));
}

/*****/
void XMEM_WRITE (unsigned int PTR, byte DATA) // Записать данные DATA
                                                // по адресу PRT
{
    xdata_buffer[PTR] = DATA;
}

/*****/
byte XMEM_READ (unsigned int PTR) // Прочитать данные по адресу PRT
{
    return xdata_buffer[PTR];
}

/*****/
void Get_Temperature (void) // Получить температуру
{
    EA = 0;                // Запретить прерывания
    TEMPER = REZ;          // Записать результат
    EA = 1;                // Разрешить прерывания

                        // Рассчитать температуру TEMPER
                        // В сотых долях градуса
    TEMPER = TEMPER - 41857;
    TEMPER = (TEMPER * 100L) / 154;
    TEMP_INT = TEMPER / 100;
    TEMP_FRAC = TEMPER - (TEMP_INT * 100);
}

```

```

/*****/
void SYSClk_Init (void)          // Инициализировать кварцевый генератор
{
    OSCXCN = 0x67;              // Запустить внешний генератор 11.0592 МГц
    Time (250);                 // Задержка на время установления
    while (!OSCXCN&0x80)        // Опросить готовность
        WDT();
    OSCICN = 0x08;              // Переключиться на внешний генератор
}

/*****/
void PORT_Init (void)           // Инициализировать порты
{
    XBR0 = 0x16;                // Разрешить периферию SPI, UART0, PCA0-1
    XBR1 = 0x00;
    XBR2 = 0x40;                // Разрешить работу Crossbar
    PRT0CF = 0x55;              // Назначить входы/выходы порта 0
                                // P0.7 <- CEX1 интерфейс PCA
                                // P0.6 <- CEX0 интерфейс PCA
                                // P0.5 <- RxD интерфейс UART
                                // P0.4 <- TxD интерфейс UART
                                // P0.3 <- NSS интерфейс SPI
                                // P0.2 <- MOSI интерфейс SPI
                                // P0.1 <- MISO интерфейс SPI
                                // P0.0 <- SCK интерфейс SPI
    PRT1CF |= 0x20;             // Назначить входы/выходы порта 1
                                // P1.5 <- 4850N
                                // P1.4 <- BUZZER
    PRT2CF = 0x00;              // Назначить входы/выходы порта 2 шины данных
                                // Выходы с "открытым истоком"
    PRT3CF = 0x00;              // Назначить входы/выходы порта 3
                                // Выходы с "открытым истоком"
                                // P3.7 <- LCD_EN
                                // P3.6 <- LCD_RW
                                // P3.5 <- LCD_DC
}

/*****/
void UART0_Init (void)          // Инициализировать порт UART
{
    SCON = 0x52;                // SCON: режим 1, 8-bit UART, разрешить RX
    TMOD = 0x20;                // TMOD: таймер 1, режим 2, 8-bit reload
    TH1 = -(SYSClk/BAUDRATE/16); // Установить величину перезагрузки таймера 1
    TR1 = 1;                    // Запустить таймер 1
    CKCON |= 0x10;              // Таймер 1 пересчитывает SYSClk
    PCON |= 0x80;               // SMOD = 1
    TI = 1;                     // Показать готовность TX (передатчика)
}

/*****/
void ADC0_Init (void)           // Инициализировать ADC0 на работу с автозапуском от
                                // таймера 3, генерацией прерывания по завершению
                                // преобразования
{

```

```

ADC0CN = 0x05;           // Запретить ADC0;
                        // Режим слежения
                        // Запуск по переполнению таймера 3
                        // Выравнивание влево
REF0CN = 0x07;         // Выбрать температурный датчик
                        // Выбрать встроенное опорное напряжение
AMXOSL = 0x0f;         // Переключить вход мультиметра
                        // на температурный датчик
ADCOCF = 0x80;         // Частота преобразования ADC = SYSCLK/16
ADCOCF |= 0x01;       // Коэффициент усиления PGA = 2
EIE2 |= 0x02;         // Разрешить прерывания от ADC
}

/*****/
void Timer3_Init (int counts) // Инициализировать таймер 3 в режим
                             // перезагрузки с генерацией прерываний
{
    TMR3CN = 0x02;         // Остановить таймер 3
                        // Стереть флаг TF3
                        // Использовать для счета SYSCLK
    TMR3RL = -counts;     // Инициализировать перезагрузку
    TMR3 = 0xffff;       // Произвести перезагрузку немедленно
    EIE2 &= ~0x01;       // Запретить прерывания таймера 3
    TMR3CN |= 0x04;       // Запустить таймер 3
}

/*****/
void ADC0_ISR (void) interrupt 15 using 1 // Прерывание завершения преобразования
{
    static unsigned int_dec=INT_DEC; // Счетчик
    static long accumulator=0L;     // Целое число результата

    ADCINT = 0; // Индикатор завершения преобразования = 0
    accumulator += ADC0; // Прочитать ADC и добавить в накопитель
    WDT ();
    int_dec--; // Обновить значение счетчика
    if (int_dec == 0) // Если 0 - результат готов
    {
        int_dec = INT_DEC; // Переустановить счетчик
        REZ = accumulator>>8; // Результат
        accumulator = 0L; // Сбросит накопитель
    }
}

/*****/
void Start_Init (void) // Начальная инициализация
{
    FLAG=0; // Запретить опрос температуры
    PORT_Init (); // Инициализировать порты
    SYSCLK_Init (); // Запустить кварцевый генератор
    LCD_Init (); // Инициализировать LCD
    UART0_Init (); // Инициализировать UART
    Timer3_Init(SYSCLK/SAMPLE_RATE); // Инициализировать таймер 3
    ADC0_Init (); // Инициализировать ADC0
}

```



```

ADCEN = 1; // Разрешить работу ADC0
RTC_Init (); // Инициализировать RTC

ON485=0; // Закрыть передатчик RS485
WDT();
RI=0; // Сбросить флаги приема и передачи
TI=0;
EA=1; // Разрешить общие прерывания
COUNT=0; // Сбросить счетчик циклов
LCD_Reset (); // Сбросить LCD
LCD_ON (); // Включить LCD
LCD_Clear (); // Очистить LCD
// Вывести стартовый заголовок на 1-й строке
LCD_XYString (0,0,"Start CI_LAN MISNET");
// Вывести версию на 2-й строке
LCD_XYString (1,0,"Version 4.0");
Beep (250); // Звуковой сигнал

if (RTC_Valid ()) // Проверить годность батарейки RTC
{
    RTC_ON (); // Включить RTC
    RTC_DM1(); // Установить двоичный режим
    RTC_24(); // Установить режим 24 часа
    // Настроить будильник на сигнал
    // каждую минуту
    RTC_SetAHause (0xFF); // Записать в часы будильника переполнение
    RTC_SetAMin (0xFF); // Записать в минуты будильника переполнение
    RTC_AIE1 (); // Разрешить прерывание будильника

    Get_Date (); // Прочитать структуру календаря
    // Вывести в буфер строки последовательно
    // номер дня месяца, имя месяца, текущий год,
    // имя дня недели
    sprintf (Buffer,"%02d %s %04d, %s",
(int)da.day, RTCM[(int)da.mounth-1], (int)da.year+2000,
RTCD[(int)da.wday-1]);
    LCD_XYString (0,0,Buffer); // Вывести буфер на 1 строку LCD
}
FLAG=1; // Разрешить опрос температуры
EIE2 |= 0x20; // Разрешить внешнее прерывание
// от RTC EI7(P1.7)
}

/*****
void EX7_ISR (void) interrupt 19 using 2 // Внешнее прерывание от RTC
{ // генерируемое каждую минуту
unsigned int i;
byte CH;

BUZZER=0; // Запустить Buzzer
i=20000; // Запустить счетчик
while (i--)
    WDTCN = 0xA5;
BUZZER=1; // Выключить Buzzer
EA=0; // Запретить прерывания

```

```

CH = RTC_Int (); // Прочитать причину прерывания RTC
// для дальнейшего использования
PRT1IF &= ~0x80; // Сбросить флаг внешнего прерывания
EA=1; // Прерывания разрешить
}

/*****/
void General_Routines (void) // Подпрограммы общего назначения
{
    if (!FLAG) return; // Если запрет, завершить подпрограмму
    EIE2 |= 0x02; // Разрешить прерывание ADC0
    Get_Temperature(); // Получить температуру
    memset (Buffer, '\0', sizeof(Buffer)); // Очистить буфер строки
    sprintf(Buffer, "T = %02d.%02d ", // Записать температуру в буфер строки
        TEMP_Int. TEMP_FRAC);
    LCD_XYString (3,0,Buffer); // Вывести буфер на 4 строку LCD
    EIE2 &= ~0x02; // Запретить прерывание ADC0
    memset (Buffer, '\0', sizeof(Buffer)); // Очистить буфер строки
    sprintf(Buffer, "%010ld", COUNT++); // записать в буфер значение счетчика
    // циклов
    LCD_XYString (1,0,Buffer); // Вывести буфер на 2 строку LCD
    Get_Time (); // Получить текущее время
    sprintf (Buffer, "%02d:%02d:%02d", // Записать в буфер значение времени
        (int)ti.hause, (int)ti.min, (int)ti.sec);
    LCD_XYString (1,12,Buffer); // Вывести буфер на 2 строку LCD
    LCD_XYString (2,0,"Waiting Command"); // Вывести на 3 строку LCD сообщение
    // об ожидании сетевой команды
}

/*****/
void Monitor (void) // Подпрограмма распознавания команд
// фрейма и их исполнения
{
    byte CH;

    if (IFrame[1]!=ADDR) // Если есть ошибка адреса, вывести
    {LCD_XYString (2,0,"Address Error "); return;} // сообщение на строку 3 LCD
    if (IFrame[5]!=Input_CS(5)) // Если есть ошибка КС, вывести
    {LCD_XYString (2,0,"CRC Error "); return;} // сообщение на строку 3 LCD
    switch (IFrame[2]) // Анализ команды
    {
        // ПОСЛЕ ВЫПОЛНЕНИЯ КАЖДОЙ КОМАНДЫ ФОРМИРУЕТСЯ ФРЕЙМ ОТВЕТА,
        // СОДЕРЖАЩИЙ НОМЕР ВЫПОЛНЕННОЙ КОМАНДЫ
        case 0x01: LCD_Clear (); // Очистить LCD
            Send_Frame (1,0,0);
            break;
        case 0x02: LCD_Reset (); // Сбросить LCD
            Send_Frame (2,0,0);
            break;
        case 0x03: LCD_Home (); // Курсор LCD домой (в адрес 0)
            Send_Frame (3,0,0);
            break;
    }
}

```

```
case 0x04: LCD_ON ();          // Включить LCD
           Send_Frame (4,0,0);
           break;
case 0x05: LCD_OFF ();        // Выключить LCD
           Send_Frame (5,0,0);
           break;
case 0x06: LCD_WriteData (IFrame[3]); // Записать в LCD
           Send_Frame (6,0,0);
           break;
case 0x07: LCD_WriteCData (IFrame[3]); // Записать с перекодировкой
           Send_Frame (7,0,0);
           break;
case 0x08: LCD_AddrStr (IFrame[3],IFrame[4]); // Установить адрес
           Send_Frame (8,0,0);
           break;
case 0x09: CH=LCD_AddrGet ();   // Прочитать адрес
           Send_Frame (9,CH,0);
           break;
case 0x0A: LCD_SetCom (IFrame[3]); // Передать команду в LCD
           Send_Frame (0x0A,0,0);
           break;
case 0x0B: LCD_Put (IFrame[3]); // Передать данные в LCD
           Send_Frame (0x0B,0,0);
           break;
case 0x0C: CH=LCD_ReadData (); // Прочитать данные из LCD
           Send_Frame (0x0C,CH,0);
           break;
case 0x0D: XMEM_CLEAR ();      // Очистить XRAM
           Send_Frame (0x0D,0,0);
           break;
case 0x0E: XMEM_WRITE (IFrame[3],IFrame[4]); // Записать в XRAM
           Send_Frame (0x0E,0,0);
           break;
case 0x0F: CH=XMEM_READ (IFrame[3]); // Прочитать из XRAM
           Send_Frame (0x0F,CH,0);
           break;
case 0x10: Get_Temperature(); // Прочитать температуру
           Send_Frame (0x10,TEMP_Int.TEMP_FRAC);
           break;
case 0x11: while (RTC_Ready ()); // Ожидать готовности
           RTC_SetYear(IFrame[3]); // Установить год
           Send_Frame (0x11,0,0);
           break;
case 0x12: while (RTC_Ready ()); // Ожидать готовности
           RTC_SetMounth(IFrame[3]); // Установить месяц
           Send_Frame (0x12,0,0);
           break;
case 0x13: while (RTC_Ready ()); // Ожидать готовности
           RTC_SetMDay(IFrame[3]); // Установить день месяца
           Send_Frame (0x13,0,0);
           break;
case 0x14: while (RTC_Ready ()); // Ожидать готовности
           RTC_SetWDay(IFrame[3]); // Установить день недели
```

```

        Send_Frame (0x14,0,0);
        break;
    case 0x15: while (RTC_Ready ()); // Ожидать готовности
               RTC_SetHause(IFrame[3]); // Установить часы
               Send_Frame (0x15,0,0);
               break;
    case 0x16: while (RTC_Ready ()); // Ожидать готовности
               RTC_SetMin(IFrame[3]); // Установить минуты
               Send_Frame (0x16,0,0);
               break;
    case 0x17: while (RTC_Ready ()); // Ожидать готовности
               RTC_SetSec(IFrame[3]); // Установить секунды
               Send_Frame (0x17,0,0);
               break;
    case 0x18: while (RTC_Ready ()); // Ожидать готовности
               RTC_SetAHause(IFrame[3]); // Установить часы будильника
               Send_Frame (0x18,0,0);
               break;
    case 0x19: while (RTC_Ready ()); // Ожидать готовности
               RTC_SetAMin(IFrame[3]); // Установить минуты будильника
               Send_Frame (0x19,0,0);
               break;
    case 0x1A: while (RTC_Ready ()); // Ожидать готовности
               RTC_SetASec(IFrame[3]); // Установить секунды будильника
               Send_Frame (0x1A,0,0);
               break;

    default:                                     // Сообщение ни 3 строке LCD
                                                // о вызове отсутствующей команды
        LCD_XYString (2,0,"Command Error ");
        Send_Frame (0xEE,0,2);
        break;
}
Time(2);
}
/*****/
void main (void) // Главная подпрограмма
{
    Start_Init (); // Произвести начальную инициализацию
    ES=1; // Разрешить прерывания UART
    FRAME_RDY=0; // Обнулить флаг готовности фрейма
    FRAME_PTR=0; // Обнулить указатель фрейма

    while (1)
    {
        WDT();
        General_Routines (); // Подпрограммы общего назначения
                             // выполняются постоянно в отсутствии команд
                             // по CI-LAN

        if (FRAME_RDY==1) // Если пришел командный фрейм
        {
            Monitor (); // Распознать, выполнить команду,
                       // сформировать и отправить фрейм ответа или

```

```
        // фрейм ошибки
    ES=1;        // Разрешить прерывания UART
    FRAME_RDY=0; // Обнулить флаг готовности фрейма
    FRAME_PTR=0; // Обнулить указатель фрейма
    }
}
}
/***** END PROGRAM *****/
```

Приведенная программа занимает довольно большой объем, более 4 К и, очевидно, имеет множество лишних команд и комментариев. Все эти излишества приведены лишь с одной целью: показать различные варианты написания сходных функций. Приведенная программа может служить основой для создания читателем собственной рабочей программы или библиотеки функций для систем малой автоматизации.

Список обозначений и сокращений

μP8, Byte-wide	Parallel Byte Wide Interface	Параллельный однобайтный интерфейс (микросхем)
2-Wire		Двухпроводный интерфейс (микросхем)
3-Wire		Трехпроводный интерфейс (микросхем)
AC, CPT	Analog Comparator	Аналоговый компаратор
ADC	Analog-to-Digital Converters	Аналого-цифровой преобразователь, АЦП
Alarm		Будильник
AMUX	Analog Multiplexer	Аналоговый мультиплексор
BUZZER		Автогенерирующий излучатель звука
CAN	Controllers Area Network	Последовательный интерфейс
CCS	Control CS	Управляющая вычислительная машина
CI LAN	Command — Information LAN	Командно-информационная сеть
CPU	Central Processor Unit	Центральное процессорное устройство
CS	Computing Systems	Электронная вычислительная машина
DAC	Digital-to-Analog Converters	Цифро-аналоговый преобразователь, ЦАП
DCS	Distributed Control System	Распределенная система управления
DPC	PC Dispatcher	Диспетчер персонального компьютера
DPTR	Data Pointer Register	Указатель данных
DS	Station Dispatcher	Диспетчер станции
E,EH	Enhanced	Элемент (узел) с расширенными возможностями
EM	External Memory	Внешняя память
Ext	External	Внешний
Frame		Фрейм (кадр) — последовательность передаваемых по сети байтов
FW	Firmware	Программное обеспечение микроконтроллера

I/O	Input/Output Lines	Линии ввода/вывода (или порта)
iButton		Разновидность прибора с встроенной памятью и двухпроводным интерфейсом
ICs	Integrated Circuits	Интегральная схема
IDE	Integrated Development Environment	Интегрированная среда разработки (программная оболочка)
IDROM	Internal Data ROM	Встроенная постоянная память данных
Int	Internal	Встроенный
IROM, Flash	Internal ROM	Встроенная постоянная память
IS	Interrupt Source	Источники прерываний
ISP	In-System Programmable	Внутрисистемное программирование
JTAG		Последовательный интерфейс программирования и отладки
LAN	Local Area Network	Локальная сеть
LCD		Жидкокристаллический дисплей (индикатор)
LED		Светодиодный индикатор
MC	Microcontroller	Микроконтроллер
MGT	Station Management	Диспетчер станции
MIMD	Multiply Instruction Multiply Data	Система с множеством потоков команд и данных
MIPS	Million Instructions per Second	Миллион инструкций в секунду (ед. измерения производительности)
MMP, MMPS	Multi-Microprocessor System	Мультимикроспроцессорная (многопроцессорная) система
MP	Microprocessor	Микропроцессор
MPS	Microprocessor Systems	Микропроцессорная система
Mixed		Смешанный (интерфейс)
Optocoupler		Оптрон, элемент оптической изоляции (развязки)
OTP	One-Time Programmable	Однократно программируемая (память)
PC	Personal Computer	Персональный компьютер
PLD		Двухрядный штыревой разъем
PLS		Однорядный штыревой разъем

PROM	Programmable ROM	Программируемая постоянная память
PS	Peripheral Stations	Периферийная станция
RAM	Random Access Memory	Память со случайным доступом (оперативная)
ROM	Read Only Memory	Память только для чтения (постоянная)
ROMless		Элемент, не содержащий памяти (микрочип)
RTC	Real Time Clock	Таймер реального времени
SAN	Small Area Networks	Малая локальная сеть
SAS	Small Automation Systems	Система малой автоматизации
Ser	Serial	Последовательный (интерфейс)
SFR	Special Function Registers	Регистры специальных функций (микрочипов)
SMBus		Последовательный интерфейс, совместим с I2C
SPI	Serial Peripheral Interface	Последовательный периферийный интерфейс
SSLC	Synchronous Serial Link Controller	Синхронный последовательный интерфейс микроконтроллеров
STD	Standard Timer Data	Стандартное представление календаря (RTC)
SV	Power Supervisory	Супервизор (монитор) питания
SW	Software	Программное обеспечение верхнего уровня
T/C	Timer / Counter	Таймер/счетчик (микрочипа)
TM	Transmission Medium	Среда передачи данных, сетевая среда
TMCS	Technological MPS	Технологическая микроконтроллерная система
TV		Телевизионный (индикатор)
UART		Универсальный последовательный интерфейс
USB	Universal Serial Bus	Универсальная последовательная шина (интерфейс)
WDT	WatchDog Timer	Охранный таймер
XRAM	Extended RAM	Дополнительная оперативная память

Литература

1. Малиновский Б. Н. История вычислительной техники в лицах. Киев: Фирма «КИТ», ПТОО «А.С.К.», 1995.
2. Гутер Р. С., Полуянов Ю. Л. От Абака до компьютера. М.: 1981.
3. <http://schools.techno.ru>
4. Частиков А. П. История компьютера. М.: Информатика и образование, 1996.
5. Каган Б. М., Каневский М. М. Цифровые вычислительные машины и системы. Энергия, М.: 1974.
6. Виноградов В. И. Информационно-вычислительные системы: Распределенные модульные системы автоматизации. Энергоатомиздат, М.: 1986.
7. Липаев В. В. История развития комплексов программ реального времени для территориально-распределенных информационных систем (1958—1988 гг.) <http://www.computer-museum.ru>
8. Колпаков К. История развития авиационных бортовых цифровых вычислительных машин в России. <http://www.computer-museum.ru>
9. Липаев В. В. Из истории развития отечественной вычислительной техники для военных систем управления в реальном времени. <http://www.computer-museum.ru>
10. Хетагуров Я. А. Из истории развития специализированных бортовых вычислительных машин. <http://www.computer-museum.ru/>
11. Вопросы кибернетики. Архитектура магистрально-модульных компьютеров специального назначения: Сб. ст./ Рос. АН. Науч. совет по комплекс. проблеме «Кибернетика», НИИ систем. исслед.; Под ред. В. Б. Бетелина. М., 1997, 141 с.
12. Виноградов Б. Н., Шахнов В. А. Распределенные микропроцессорные системы и локальные вычислительные сети ЭВМ // Микропроцессорные средства и системы, 1984, № 4, с. 26—31.
13. Пискулов Е. А., Бекетов Е. О. Современные архитектуры вычислительных систем: Учеб. пособие. М.: Изд-во МАИ, 1994.
14. Дорохин В. Локальные вычислительные сети: создание и развитие <http://www.computer-museum.ru>
15. Бэрри Нанс Компьютерные сети. М.: БИНОМ, 1996, 395 с. Пер. изд.: Introduction to networking/Barry Nance. S.l., 1994.

16. Никольский О. А. Перспективные модульные системы и локальные сети. // Микропроцессорные средства и системы. 1984, № 2, с. 91—92.
17. Conference on local computer networks (17; 1992; Minneapolis, Mn). 17th conference on local computer networks, Sept.13—16, 1992, Minneapolis (Mn): Proc. Los Alamitos(Ca) et al: IEEE computer soc.press, 1992, XVI, 710 p. IEEE publications/Inst.of electrical and electronics engineers (New York).
18. Локальные вычислительные сети / Под ред. С. В.Назарова. Кн. 1—4: Принципы построения, архитектура, коммуникационные средства. М., Финансы и статистика, 1994.
19. Прангишвили И. В. Микропроцессоры и микро-ЭВМ. М.: Энергия, 1979.
20. Прангишвили И. В., Стецора Г. Г. Микропроцессорные системы. М.: Наука, 1980.
21. Прангишвили И. В., Стецора Г. Г., Подлазов В. С. Локальная сеть обработки данных с децентрализованным пространственно-временным управлением обменов информации. Тезисы докладов II Всесоюзной конференции «Вычислительные сети коммутации пакетов». Рига: Институт электроники и вычислительной техники, 1981, т. 2, с. 104—107.
22. Прангишвили И. В. Применение микропроцессоров в приборостроении / Приборы и системы управления, 1981, № 2, с. 48—49.
23. Прангишвили И. В., Виленкин С. Я., Медведев И. Л. Многопроцессорные вычислительные системы с единым потоком команд. М.: Энергоатомиздат, 1983.
24. Прангишвили И. В. Микропроцессоры и локальные сети микро-ЭВМ в распределенных системах управления. Энергоатомиздат, М.: 1985.
25. Якубайтис Э. А. Архитектура вычислительных сетей. М.: Статистика, 1980.
26. Якубайтис Э. А. Архитектура локальных вычислительных сетей. Тезисы докладов II Всесоюзной конференции «Вычислительные — сети коммутации пакета». Рига: Институт электроники и вычислительной техники, 1981, т. 2, с. 69.
27. Игнатъев В. М., Колосков В. А., Титов В. С. Проектирование самоорганизующейся дефектоустойчивой микроконтроллерной сети.: Тул. гос. техн. ун-т. Тула, 1995.
28. Однородные вычислительные структуры, среды и распределенные системы. Всероссийская науч.-техн. конф. (9; 1996). IX Всероссийская научно-техническая конференция «Однородные вычислительные структуры, среды и распределенные системы». ОВС-96: Тез. докл. М., 1996.
29. Кравец О. Я. Вычислительные сети: архитектура, оптимизация, управление: Учеб. пособие.: Воронеж. гос. техн. ун-т. Воронеж, 1996.

30. Локальные вычислительные сети / Под ред. С. В. Назарова. Кн. 1: Принципы построения, архитектура, коммуникационные средства, М., Финансы и статистика, 1994.
31. Кабанов В. А. Микропроцессорные системы автоматизации: Учеб. пособие по курсу «Микропроцессоры и их применение» / Под ред. А. И. Карякина. М.: Изд-во МЭИ, 1993.
32. Современные магистрально-модульные системы для автоматизации промышленности: Справ. / Сост.: Е. В. Дергачева. М.: Всерос. НИИ межотрасл. информ., 1993.
33. Мячев А. А. и др. Интерфейсы систем обработки данных: Справочник / Под ред. А. А. Мячева. М.: Радио и связь, 1989.
34. Якубайтис Э. А. Классификация вычислительных сетей / Автоматика и вычислительная техника, 1983, № 1, с. 3—6.
35. Cotton I. W. Technologies for local computer networks. Computer Networks, 1980, № 5, p. 197—208.
36. Николайчук О. И. Что такое «Малая автоматизация» или чем увлекаются сегодня западные радиолобители // Схемотехника, 2001, № 10, с. 3—5.
37. Nicolaiciuc O., Gutuleac E., Cheibas V. The main principles of small systems development for manufactarin and research automation // Computer Science Journal of Moldova, 1999, vol. 7, № 2(20), p. 237—246.
38. Nicolaiciuc O., Gutuleac E., Cheibas V. The small local computer network for automation of manufacture and scintific researches // Computer Science Journal of Moldova, 1999, vol. 7, № 2(20), p. 228—236.
39. Nicolaiciuc O., Gutuleac E., Cheibas V. The Small LAN «SISNET» // The International Symposium System Theory «SINTES-10», Xth-Edition, 25—26 May, 2000, Craiova, ROMANIA, p. 14—16.
40. Николайчук О. Командно-информационные сети — что это такое? Термины и схемотехника // Схемотехника, 2001, № 6, с. 26—30.
41. Николайчук О. Командно-информационные сети — что это такое? Алгоритмы и программы // Схемотехника, 2001, № 7, с. 37—41.
42. CAMAC, Compatible extended use of the CAMAC dataway. Final draft, ECA-ESONE, Publication, March 1983, 20 p.
43. FASTBUS. A modular high speed data acquisition system for high energy physics and other application.ESONE/FB/01. ESONE Committee, May 1983, 51 p.
44. Proposed standart specification for P896/D3.4.1. Advanced microcomputer system backplane.IEEE Computer Society Microcomputer Standarts Committee, 1981, 65 p.

45. Басиладзе С. Г. Стандарт Мультибас II и перспективы его применения в аппаратуре автоматизации эксперимента // Приборы и техника эксперимента. 1985, № 4, с. 7—27.
46. Басиладзе С. Г. Анализ предложений новых стандартов для систем автоматизации эксперимента // Приборы и техника эксперимента. 1986, № 3, с. 7—32.
47. ЕВРОБАС. Стандарт на модульные системы управления, обработки и сбора данных для промышленного применения / Пер. с англ. Е. И. Котова, №Г-10862; Под ред. В. И. Виноградова. М.: ВЦП научно-технической литературы и документации, 1981, 166 с.
48. ОСТ25 969—83 Система малых электронных вычислительных машин. Интерфейс И41. Технические требования.
49. Николайчук О. Быстродействующий диспетчер персонального компьютера RS232C — RS232C / RS485 с оптической развязкой // Схемотехника, 2002, № 5, с. 34—37.
50. Николайчук О. Интерфейс персонального компьютера и локальной сети микроконтроллеров // Приборы и техника эксперимента. 2000, № 3, с. 38—40.
51. Nicolaiciuc O. A PC and Local Microcontroller Network Interface // Instruments And Experimental Technique, Consultants Bureau, New York, Vol.43, № 3, 2000, p. 318—320.
52. Nicolaiciuc O. The Station of Command Information LAN «MISNET» // Proceeding of the 5th International Conference on «Development and Application Systems», DAS2000, 18—20 May, 2000, Suceava, ROMANIA, p. 298—300.
53. Nicolaiciuc O., E. Gutuleac. The Working Algorithms of Command Information LAN «MISNET» // Proceeding of the 5th International Conference on «Development and Application Systems», DAS-2000, 18—20 May, 2000, Suceava, ROMANIA, p. 301—306.
54. Nicolaiciuc O., E. Gutuleac, V.Cheibas. The LAN «MISNET» // The International Symposium On System Theory «SINTES10», Xth-Edition, 25—26 May, 2000, Craiova, ROMANIA, p. 17—20.
55. International symposium on computer architecture (21st; 1994; Chicago (II)). Proceedings the 21st annual international symposium on computer architecture, Apr.18—21, 1994,Chicago(II). Washington (DC) et al: IEEE computer soc. press, 1994, XII, 394 p.: ill.
56. High-performance computing and networking: Intern. conf. and exhibition, Brussels, Apr.15—19,1996; Proc./ HPCN Europe 1996; Ed. H.Liddell. Berlin et al.: Springer, 1996, XXV, 1040 p.
57. Вопросы кибернетики. Архитектура магистрально-модульных компьютеров специального назначения: Сб. ст./ Рос. АН. Науч. совет по комплекс. про-

блеме «Кибернетика», НИИ систем. исслед.; Под ред. В. Б. Бетелина. М., 1997, 141 с.

58. Трехцинский Р. Актуальное состояние разработок системы И42 (ММС MULTIBUS-II) в ПНР в рамках проблемы 2.2.2 / Международный семинар по проблеме реализации комплексной программы научно-технического прогресса стран-членов СЭВ до 2000 года. Международный семинар-показ КАМАК-оборудования «КАМАК-87» (Кишинев, 13—17 апреля 1987): докл. Варшава, 1987, 9 с.

59. Фрунзе А. Рабы своих семейств, или почему разработчики микроконтроллерной техники не очень охотно меняют базу для своих разработок.
<http://www.compitech.ru>

60. <http://www.atmel.com>

61. <http://www.atmel.ru>

62. Николайчук О. Очередные достижения фирмы ATMEL в области микроконтроллеров семейства x51 // Схемотехника, 2002, № 6, с. 42—46.

63. Николайчук О. Новая x51-совместимая микросистема сбора данных MAX7651 фирмы MAXIM // Схемотехника, 2002, № 9, с. 45—47.

64. Николайчук О. Семейства x51-микроконтроллеров фирмы CYGNAL // Компоненты и технологии, 2002, № 1, с. 86—91.

65. Николайчук О. Эксперименты с микроконтроллерами фирмы CYGNAL // Схемотехника, 2002, № 8, с. 48—50.

66. Николайчук О. И. x51-совместимые микроконтроллеры фирмы Cygnal. — М.: ООО «ИД СКИМЕН», 2002. 472 с.

67. Николайчук О. Схемотехника универсальных технологических контроллеров: Эволюционный контроллер // Схемотехника, 2002, № 7, с. 30—32.

68. Николайчук О. Схемотехника универсальных технологических контроллеров (цикл статей) // Схемотехника, с № 9, 2001 по № 8, 2002.

69. <http://www.analog.com>

70. <http://www.dalsemi.com>

71. <http://www.etchcorp.com>

72. <http://www.impweb.com>

73. <http://www.linear-tech.com>

74. <http://www.spezial.ru> и <http://www.maxim-ic.com>

75. <http://www.microchip.com>

76. <http://www.sipex.com>

77. <http://www.telcom-semi.com>

78. <http://www.ti.com>

79. <http://www.thomson.com>
80. <http://www.philips.com>
81. <http://www.epson.co.jp>
82. <http://www.motorola.com>
83. <http://www.benchmark.com>
84. Николайчук О. Современные микросхемы памяти (часть 1) // Компоненты и технологии, 2002, № 4, с. 32—36.
85. Николайчук О. Современные микросхемы памяти (часть 2) // Компоненты и технологии, 2002, № 6, с. 74—77.
86. <http://www.issi.com/61c256ah.pdf>
87. <http://www.toshiba.com/>
88. <http://www.idt.com>
89. <http://www.pericom.com>
90. <http://www.hitachi.com>
91. <http://www.itis.spb.ru>
92. [http://www.itis.spb.ru/Компоненты iButton фирмы Dallas Semiconductor.htm](http://www.itis.spb.ru/Компоненты_iButton_фирмы_Dallas_Semiconductor.htm)
93. <http://ham.kiev.ua/sprav/leds/ledh10.htm>
94. АС СССР № 1238072. Устройство адресации памяти. Николайчук О. И., МКИ4 G06F 9/36. БИ, 1986, № 22.
95. <http://www.burr-brown.com>
96. <http://www.exar.com>
97. <http://www.fairchildsemi.com>
98. <http://www.intersil.com>
99. <http://www.national.com>
100. <http://www.tokoam.com>
101. <http://www.maxim-ic.com> — max187-max189.pdf
102. <http://www.maxim-ic.com> — max1240-1241.pdf
103. <http://www.analog.com> — ad7475_95_a.pdf
104. <http://www.analog.com> — ad7896.pdf
105. <http://www.analog.com> — 1985_0.pdf
106. <http://www.maxim-ic.com> — 1514.pdf
107. <http://www.maxim-ic.com> — 1588.pdf

108. <http://www.hea.com>
109. <http://www.infineon.com>
110. <http://www.icsi.com.tw>
111. <http://www.issiusa.com>
112. <http://www.intel.com>
113. <http://www.okisemi.com>
114. <http://www-us2.semiconductors.philips.com>
115. <http://www.ssti.com>
116. <http://www.syncmos.com.tw>
117. <http://www.winbond.com.tw>
118. Николайчук О. Новая x51-совместимая микросистема сбора данных MAX7651 фирмы MAXIM // Схемотехника, 2002, № 9, с. 45—47.
119. <http://www.maxim-ic.com> — MAX7651-MAX7652.pdf
120. <http://www.cygnal.com>
121. <http://www.atos.ru>
122. <http://www.keil.com>
123. <http://www.cygnal.com/datasheets/c8051f2xx.pdf>
124. <http://www.cygnal.com/datasheets/c8051f0xx.pdf>
125. <http://www.cygnal.com/datasheets/c8051f018-9.pdf>
126. <http://www.cygnal.com/datasheets/c8051f02x.pdf>
127. <http://www.cygnal.com/datasheets/c8051f04x.pdf>
128. <http://www.cygnal.com/datasheets/c8051f12x.pdf>
129. <http://www.micronas.com>, Tba2800.pdf
130. Николайчук О. Интеллектуальный малогабаритный фотодатчик, работающий на отражение // Схемотехника, 2001, № 7, с. 60—63.
131. <http://www.vishay.com>

Содержание

Введение	3
1. Системы малой автоматизации	5
1.1. Предпосылки создания систем малой автоматизации	5
1.2. Основные понятия о распределенных системах управления.....	9
1.3. Основные понятия о локальных вычислительных сетях	13
1.4. Основные понятия о системах малой автоматизации.....	18
2. Командно-информационные сети	26
2.1. Основные понятия	26
2.2. Диспетчеры персонального компьютера.....	29
2.3. Способы управления доступом к каналу сети RS485.....	39
2.4. Диспетчеры периферийных станций	47
2.5. Формат фреймов и общий алгоритм работы <i>CI-LAN</i>	51
2.6. Тестовая программа контроллера станции <i>CI-LAN</i>	57
3. Схемотехника универсальных технологических контроллеров	62
3.1. Предпосылки создания универсальных технологических контроллеров	62
3.2. Основные понятия и тенденции развития <i>TMCS</i>	63
3.3. Технические требования к главному микроконтроллерному модулю <i>TMCS</i>	69
3.4. Обобщенная функциональная схема центрального микроконтроллерного модуля.....	70
3.5. Супервизоры питания и охранные таймеры.....	73
3.6. Таймеры реального времени	83
3.7. Устройства ввода/вывода и расширения	95
3.8. Аналого-цифровые преобразователи <i>ADC</i>	105
3.9. Цифроаналоговые преобразователи.....	119
4. Проектирование систем малой автоматизации	149
4.1. Выбор микроконтроллеров для систем малой автоматизации	149
4.2. Эволюционный контроллер.....	172
4.3. Вариант универсального контроллера.....	178
4.4. Мощный универсальный технологический контроллер на микроконтроллере <i>Cygnal</i>	183
4.5. Универсальная «слотовая» система сбора данных на микроконтроллере фирмы <i>Cygnal</i>	187
4.6. Специализированные технологические контроллеры-фотодатчики	198
4.7. Базовая программа системы малой автоматизации	207
Список обозначений и сокращений	237
Литература	240