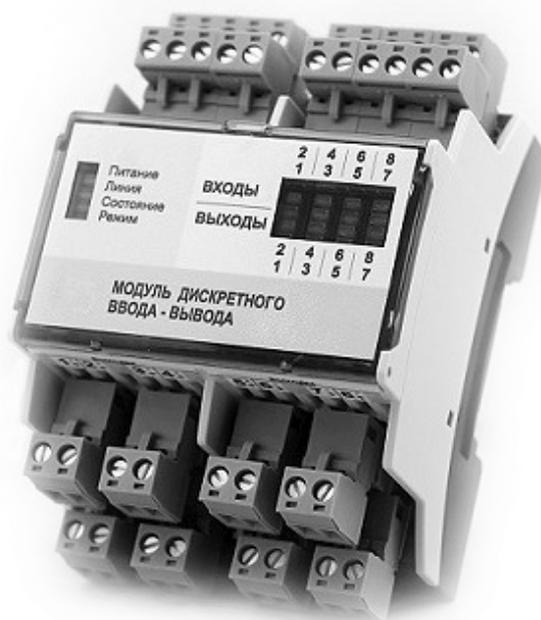


А. О. Ключев, П. В. Кустарев, А. Е. Платунов

АППАРАТНЫЕ СРЕДСТВА ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ



Санкт-Петербург

2015

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
УНИВЕРСИТЕТ ИТМО**

А. О. Ключев, П. В. Кустарев, А. Е. Платунов

**АППАРАТНЫЕ СРЕДСТВА
ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ
СИСТЕМ**

Учебное пособие

 **УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург

2015

Ключев А. О., Кустарев П. В., Платунов А. Е. Аппаратные средства информационно-управляющих систем. Учебное пособие. — СПб: Университет ИТМО, 2015. — 65 с.

Учебное пособие является введением в организацию информационно-управляющих (ИУС), встраиваемых и киберфизических систем. В нем рассматриваются основные понятия и характеристики таких систем, элементная база, аппаратные и программные средства, используемые для построения ИУС, особенности реализации и программирования современных контроллеров ИУС.

Для подготовки бакалавров по направлению 09.03.01 «Информатика и вычислительная техника», 09.03.04 «Программная инженерия».

Рекомендовано к печати заседанием ученого совета факультета КТиУ Санкт-Петербургского государственного научно-исследовательского университета, протокол № 3 от 10.03.2015.



Университет ИТМО — ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО — участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО — становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2015

© Ключев А. О., Кустарев П. В., Платунов А. Е., 2015

Содержание

Введение	5
1 Информационно-управляющие системы	7
1.1 Управление реальными объектами	7
1.2 Реальный масштаб времени	7
1.3 Надежность	8
2 Классификация ИУС	9
2.1 Интеграция с объектом управления	9
2.2 Степень распределенности системы управления	10
2.3 Назначение системы управления	10
2.4 Степень участия человека в процессе управления	11
2.5 Другие способы классификации	12
3 Функциональные элементы ИУС	13
3.1 Требования к процессорам, применяемым в ИУС	13
3.2 Микроконтроллеры	14
3.3 Программируемые логические интегральные схемы	24
4 Функциональные блоки микроконтроллеров и СнК	25
4.1 Аппаратные прерывания	25
4.2 Таймер	27
4.3 Часы реального времени	28
4.4 Сторожевой таймер	28
4.5 Система контроля питания	30
4.6 Тактовый генератор	31
4.7 Средства понижения энергопотребления	32
4.8 FLASH память	32
4.9 Энергонезависимая конфигурационная память	33
4.10 Контроллер прямого доступа к памяти	33
4.11 Устройство захвата-сравнения	33
4.12 Порт ввода-вывода	34
4.13 Аппаратные интерфейсы ИУС	35
4.14 Отладочные интерфейсы ИУС	44

4.15 Встроенный загрузчик	46
5 Конструктивные особенности ИУС	47
5.1 Классы защиты по IP	47
5.2 Температурный диапазон компонентов	47
5.3 Защита от влажности и агрессивных сред	49
5.4 Защита от механических воздействий	50
5.5 Защита от электромагнитных помех	52
5.6 Срок службы компонентов	56
Литература	60

Введение

Первые компьютерные системы, ориентированные на управление, появились в конце пятидесятих годов двадцатого века [1]. Специфика работы таких систем была связана в первую очередь с несовершенством (малой надежностью и габаритными размерами) элементной базы. Для работы электронных вычислительных машин того времени требовались большие помещения, много электроэнергии, большое количество обслуживающего персонала. Компьютер располагался в специально оборудованном вычислительном центре, удаленном от объекта управления. От управляющего компьютера до объекта управления необходимо было прокладывать кабели для передачи сигналов управления и информации от датчиков.

По мере развития электроники и вычислительной техники началась миниатюризация вычислительных машин. В один прекрасный момент оказалось, что в большинстве случаев дешевле разместить рядом с нужным местом сравнительно компактный компьютер и передавать данные от него через компьютерную сеть, чем тянуть много дорогостоящего кабеля. Так началась эра распределенных информационно-управляющих систем (РИУС).

Примерно в начале 90-х годов миниатюризация компьютеров достигла такого уровня, что появилась возможность размещать компьютерные управляющие модули непосредственно внутри объекта управления. Вычислительные управляющие системы, помещенные внутрь объекта управления, начали называть встраиваемыми или встроенными системами [2].

В начале 21 века развитие электроники, оптики и механики позволило создать так называемые киберфизические системы (КФС) [3]. В отличие от встраиваемых систем, КФС еще теснее взаимодействуют с объектом управления. КФС позволяют создавать умные вещи, умные производства, умные медицинские имплантаты, беспилотные машины и самолеты, роботов нового поколения. В процессе эволюции управляющих компьютерных систем менялись и способы проектирования. Изначально программное обеспечение проектировалось отдельно от аппаратуры, так как аппаратура считалась чем-то постоянным и

незыблемым. В связи с падением цен на производство электронных модулей появилась возможность выпускать оборудование на заказ, по мере необходимости. Особый толчок в развитии быстро заменяемой и реконфигурируемой аппаратной базы был дан при появлении программируемых логических интегральных схем (ПЛИС). Для ускорения проектирования таких систем, в середине 90-х годов двадцатого века была предложена технология совместного проектирования (CoDesign).

В настоящее время решается задача комплексного проектирования всей системы целиком, включая программное обеспечение, аппаратуру и механические части, необходимые для тесной интеграции с объектом управления.

В 2013 году мировой рынок встраиваемых систем оценивался в 140 миллиардов долларов. Ожидается, что к 2020 году сумма достигнет 214 миллиардов долларов [4].

Такие технологии как КФС, 3D принтеры и соответствующие технологии разработки позволят революционным образом изменить нашу жизнь в ближайшем будущем, делая возможным быстрое проектирование и 3D печать необходимых вещей без участия человека и без необходимости в крупногабаритных производственных комплексах.

1 Информационно-управляющие системы

Информационно-управляющая система (ИУС) — цифровая система контроля или управления некоторым реальным объектом, называемым «объект управления».

Специфика работы управляющих компьютерных систем состоит в следующем:

- Необходимо управлять реальными объектами;
- Работа протекает в реальном масштабе времени;
- Предъявляются повышенные требования к надежности и безопасности.

1.1 Управление реальными объектами

В отличие от чисто информационных систем, информационно-управляющим системам приходится работать не с абстрактными, а с реальными объектами. Это означает, что в алгоритмах управления и в аппаратуре сопряжения необходимо учитывать все особенности объекта управления. В большинстве случаев абсолютно недопустимыми являются ошибки, зависания программ и несовместимость по электрическим или механическим параметрам. Если зависший мобильный телефон вызовет всего-навсего вполне понятное раздражение владельца (это конечно скажется на репутации фирмы-производителя), то ошибка в бортовой системе управления самолета или атомной электростанции может вызвать катастрофу с очень серьезными последствиями. Таким образом, можно сказать, что проектирование информационно-управляющих систем является весьма сложной и ответственной областью человеческой деятельности.

1.2 Реальный масштаб времени

Система реального времени — вычислительная система с гарантированным временем реакции на события. Система реального времени (СРВ) — вычислительная система, в которой время формирования вы-

ходного воздействия является существенным. Существует два основных типа систем реального времени:

- Система мягкого реального времени. Невыполнение условий мягкого реального времени не приводит к катастрофическим последствиям для целевой функции системы.
- Система жесткого реального времени. Невыполнение требований жесткого реального времени приводит к катастрофическим последствиям для целевой функции системы [2].

1.3 Надежность

Надежность (dependability) — свойство готовности и влияющие на него свойства безотказности и ремонтпригодности, возможность технического обслуживания [5]. Надежность можно рассматривать как характеристику работоспособности изделия, связанную с определенным моментом времени. Надежность характеризует, какое время потребуется на восстановление работоспособности изделия в случае его выхода из строя.

К основным аспектам надежности относятся:

1. *Готовность* — это способность изделия выполнить требуемую функцию при данных условиях в предположении, что необходимые внешние ресурсы обеспечены [5]. Свойство готовности определяет частоту нахождения системы в работоспособном или не работоспособном состоянии.
2. *Безотказность* — это способность изделия выполнить требуемую функцию в заданном интервале времени при данных условиях [5]. Свойство надежности определяет вероятность эксплуатации системы в данный интервал времени без отказов.
3. *Ремонтпригодность* — это способность изделия при данных условиях использования и технического обслуживания к поддержанию или восстановлению состояния, в котором оно может выполнить требуемую функцию [5]. Свойство ремонтпригодности определяет как быстро и насколько просто можно выполнить техническое обслуживание или ремонт системы [6].

2 Классификация ИУС

Современные ИУС можно классифицировать по следующим основным признакам:

1. По степени интеграции с объектом управления;
2. По степени распределенности;
3. По назначению;
4. По степени участия человека.

Абсолютно все управляющие системы объединяет одно свойство — способность этих систем к управлению. На вход управляющей системы поступает информация с датчиков, а к выходам системы подключены различные исполнительные устройства [7].

2.1 Интеграция с объектом управления

По степени интеграции с объектом управления различают три типа управляющих систем:

1. Информационно-управляющая система;
2. Встроенная система;
3. Киберфизическая система.

Термин ИУС можно трактовать двояко. С одной стороны, это достаточно старый термин, обозначающий все классы управляющих вычислительных систем. С другой стороны, под ИУС можно понимать систему, которая находится вне объекта управления.

Встроенные вычислительные системы (ВВС) (embedded system) — специализированные (заказные) вычислительные системы (ВС), непосредственно взаимодействующие с объектом контроля или управления и объединенные с ним единой конструкцией.

Киберфизическая система (КФС) (Cyber Physical System) (CPS) — специализированная вычислительная система, имеющая физические средства взаимодействия (электрические, химические, оптические, механические, биологические и т. п.) с объектом контроля и управления. Киберфизические системы появились сравнительно недавно и в настоящий момент являются наиболее сложными, как с точки зрения внутреннего устройства, так и с точки зрения их проектирования.

2.2 Степень распределенности системы управления

По степени распределённости различают распределенные и сосредоточенные ИУС. Степень распределённости ИУС тесно связана с силой связи между компонентами вычислительной системы. По силе связи различают слабо связанные и сильно связанные системы:

1. *Слабо связанная система* — система, в которой интенсивность обмена данными в рамках одного вычислительного процесса значительно выше интенсивности обмена данными между разными вычислительными процессами. Оборудование слабо связанных систем может быть разнесено по большой площади или объёму. Для связи между блоками используются различные сетевые интерфейсы, обеспечивающие надёжную передачу данных при заданных расстояниях и уровне помех.
2. *Сильно связанная система* — система, в которой интенсивность обмена данными в рамках одного вычислительного процесса соизмерима или меньше интенсивности обмена данными между разными вычислительными процессами. Оборудование сильно связанных систем расположено в непосредственной близости друг от друга, а для передачи данных между блоками таких систем используются быстрые интерфейсы.

При построении вычислительной системы используют принцип декомпозиции, при котором отдельные подсистемы группируют по такому критерию как сила их связи. Сильно связанные подсистемы размещают вместе, а слабо связанные компоненты выносят в отдельные подсистемы.

2.3 Назначение системы управления

Информационно-управляющие системы можно разделить на следующие группы по назначению:

1. Системы автоматического управления (САУ). САУ предназначены для управления промышленным производством, горнодобывающим оборудованием, нефтяными вышками и т. д.
2. Измерительные системы. Различные приборы: генераторы сигналов, осциллографы, логические анализаторы, измерители свойств материалов, научные приборы.
3. Системы сбора информации с датчиков. Охранные системы, противопожарные системы, системы экологического мониторинга и т. д.
4. Медицинские системы. Диагностическое оборудование, системы жизне-обеспечения.
5. Системы передачи данных (коммуникационные системы). Коммутаторы, маршрутизаторы, точки доступа, сотовые станции, сотовые телефоны и т. д.
6. Системы управления подвижными объектами. Бортовые системы управления автомобилями, поездами, самолетами, кораблями, космическими аппаратами.
7. Подсистемы вычислительных систем общего назначения. Контроллеры жестких дисков, контроллеры USB, контроллеры клавиатуры, контроллеры сенсорной панели и т. д.
8. Робототехника. Системы управления робототехническими комплексами и беспилотными летательными аппаратами (БПЛА).
9. Умный дом. Системы управления светом, энергопотреблением, охраной жилого дома.
10. Умные вещи. Контроллеры бытового оборудования, позволяющие сделать привычные вещи интеллектуальными и обеспечить их интеграцию в системы типа «Умный дом». Умные часы, фитнес браслеты, погодные станции, микроволновые печи, холодильники, пылесосы, телевизоры, кондиционеры, стиральные машины и т. д.

2.4 Степень участия человека в процессе управления

По степени участия человека в процессе управления различают три вида систем:

1. Система с ручным управлением. ИУС не задействована вообще.
2. Автоматизированная система. Часть работы выполняет ИУС, а часть человек, или всю работу выполняет ИУС, но под контролем человека и человек может отменить действие системы в любой момент.
3. Автоматическая система. Человек не принимает никакого участия в управлении.

Во многих ИУС присутствует все три режима работы. В основном система работает в полностью автоматическом режиме, но при необходимости может быть переведена в режим полуавтомата (то есть может работать в автоматизированном режиме). В случае каких либо нештатных ситуаций система может быть переведена в режим ручного управления. Примеры ИУС работающих в трех режимах:

1. Система управления самолетом (автопилот);
2. Система управления судном (авторулевой);
3. Система управления фотоаппаратом (автомат, «творческий режим» P/A/S, режим ручного управления M).

2.5 Другие способы классификации

ИУС также можно классифицировать следующим образом:

1. По соотношению управляющих, коммуникационных и вычислительных функций. Блоки, в которых преобладают управляющие функции, обычно называют контроллерами (от control, управление). Блоки, предназначенные для передачи данных, называют сетевыми или коммуникационными блоками. Блоки, основная нагрузка которых состоит в различных вычислениях, называют вычислителями.
2. По способу распараллеливания. Можно выделить два способа: крупногранулярный и мелкогранулярный. При крупногранулярном распараллеливании выделяются целые задачи. При мелкогранулярном выделяются отдельные функции. Например, в первом случае мы управляем цехом № 8, а во втором заслонкой № 16.

3 Функциональные элементы ИУС

Любая ИУС построена из определённого числа функциональных элементов. Данные элементы могут быть поделены на:

- Элементы вычислительного ядра;
- Элементы системы ввода-вывода;
- Устройства ввода-вывода (УВВ). Не рассматриваются в данном учебном пособии.

Остановимся на первых двух группах.

В элементы вычислительного ядра входят следующие блоки:

- Процессорные блоки;
- Блоки памяти различного назначения (буферы, кэш-память и т. д.);
- Контроллеры прерываний, контроллеры прямого доступа к памяти и т. д.;
- Интерфейсы, объединяющие всё вышеперечисленное.

Следующие блоки входят в элементы системы ввода-вывода:

- Интерфейсы ввода-вывода;
- Процессоры ввода-вывода;
- Контроллеры ввода-вывода;
- Сервисные функциональные блоки: инструментальные блоки, тестово-диагностические блоки и т. п.

3.1 Требования к процессорам, применяемым в ИУС

Перечислим основные особенности процессоров для ИУС:

1. В ИУС чаще всего используют специализированные процессоры. В небольших системах используют однокристалльные микро-ЭВМ с интегрированной в кристалл памятью и периферийными устройствами.
2. Центральный процессор для ИУС ориентирован на управление. Такие процессоры не позволяют быстро обрабатывать большие

объемы данных. Основная задача — обработка событий и формирование сигналов управления. Связь с периферийными устройствами производится в основном через регистры специального назначения, так называемые SFR (Special Function Register). В зависимости от типа микроконтроллера эти регистры могут находиться в адресном пространстве регистров общего назначения, а могут находиться в общем адресном пространстве.

3. Как правило, процессоры для ИУС выпускаются для расширенного температурного диапазона.
4. Для надежного функционирования операционных систем реального времени (ОС РВ) во многих случаях необходим режим защиты памяти, т. е. поддержка изолированных адресных пространств для каждого процесса. Этот механизм присутствует в наиболее дорогих и сложных процессорных ядрах.
5. Для эффективного переключения контекстов необходима соответствующая поддержка в процессоре, такая как банки регистров, специальные команды для сохранения фреймов стека и т. д.
6. Для своевременной реакции на различные события в процессоре для встраиваемых применений существует развитая система прерываний. Время входа в прерывание стараются сделать минимальным. Для этого используют специальные решения, связанные к примеру с банками регистров. Экономия времени может достигаться тем, что во время вызова обработчика прерывания не происходит сохранения в стек контекста прерванной задачи, так как у каждого прерывания есть свой собственный регистр банков. При использовании операционных систем реального времени такое решение уже теряет смысл, так как на большое количество процессов регистровых банков может не хватить [2].

3.2 Микроконтроллеры

3.2.1 Классификация микроконтроллеров и микропроцессоров для ИУС

Микроконтроллером называют контроллер, построенный на основе микропроцессорной элементной базы. Микроконтроллер содержит в одном кристалле микропроцессор и набор периферийных устройств

и контроллеров: контроллер прерываний, таймеры, контроллер сети, контроллер последовательного канала, контроллер памяти, контроллер ПДП и т. д. Существует несколько способов, с помощью которых можно производить классификацию микроконтроллеров.

- По разрядности различают 8, 16 и 32 разрядные микроконтроллеры.
- По возможностям в области обработки сигналов можно рассматривать обычные микроконтроллеры и DSP-микроконтроллеры.
- По области применения различают следующие микроконтроллеры: автомобильные, промышленные, для контроллерных сетей, управления двигателями, управления беспроводными сетями.

По объему вычислительных ресурсов условно можно выделить четыре характерные группы микроконтроллеров:

- Периферийные процессоры — Microchip PIC 10, PIC12, PIC16, PIC18, PIC24, Atmel AT90xxxx и т. п.
- Универсальные 8-ми и 16-ти разрядные ОМЭВМ — Intel MCS51, Siemens SAB 5xx, Atmel Mega10x и т. п.
- Универсальные 16-ти и 32 разрядные ОМЭВМ — MSP430, Cortex-M3, Coldfire V2 и т. п.
- Универсальные однокристальные 32-х разрядные микроконтроллеры и процессоры — ARM Cortex A8, ARM Cortex A9, ARM Cortex A15, и т. п.

Для первой категории процессоров характерны следующие особенности:

- небольшой объем памяти данных (десятки-сотни байт);
- небольшой объем памяти программ (единицы-десятки килослов);
- сравнительно высокое быстродействие;
- система команд RISC;
- низкое энергопотребление;
- малое число выводов;
- невозможность подключения внешней памяти.

Пример структуры микроконтроллера первой категории можно увидеть на рисунке 3.1.

Старшие модели микроконтроллеров могут иметь в своем составе сетевые контроллеры. Основная идея в этих контроллерах — обеспечение создания устройств с низким энергопотреблением и минимальным количеством компонентов на плате.

Для второй категории процессоров (см. рисунок 3.3) характерна возможность использования внешней (внекристальной) памяти. Отличает эту категорию низкая цена и небольшие вычислительные ресурсы. Производительность таких микроконтроллеров, как правило,

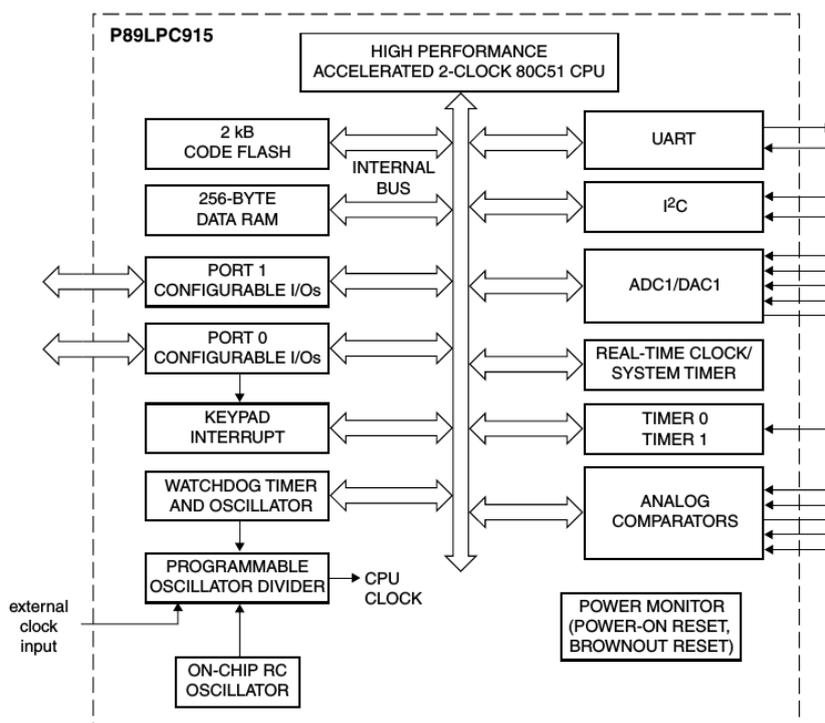


Рисунок 3.1: Структура микроконтроллера SP89LPC915 [8].

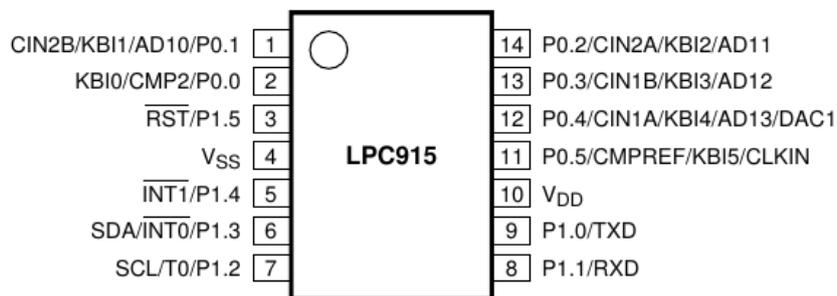


Рисунок 3.2: Микроконтроллер P89LPC915: расположение выводов.

значительно ниже, чем у первой категории. Контроллеры этого типа применяются в основном в простых и дешевых устройствах, не предъявляющих повышенных требований по производительности и энергопотреблению, но имеющих повышенные требования по объему программного кода и требуемой памяти данных.

Третья категория процессоров имеет развитые аппаратные средства для повышения производительности при обработке информации, гораздо более мощный, по сравнению с первой и второй категориями центральный вычислитель и расширенное адресное пространство. Начиная с этой категории, в состав микроконтроллеров производители начинают наиболее активно включать сетевые контроллеры. В настоящее время это наиболее распространенные микроконтроллеры.

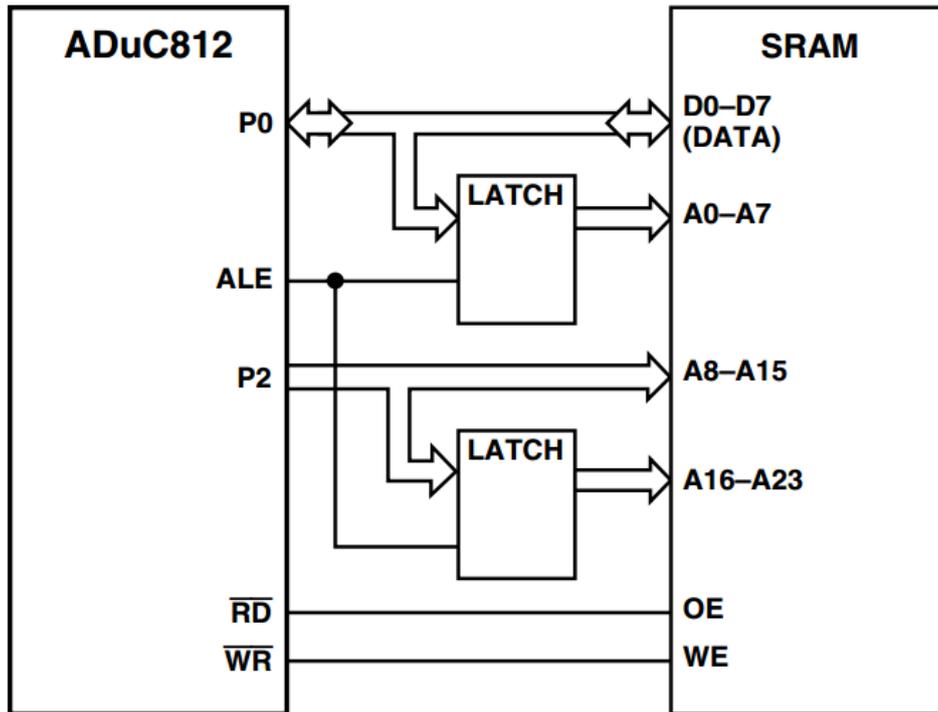


Рисунок 3.3: Микроконтроллер ADuC812: подключение внешней памяти [9].

На рисунке 3.4 можно увидеть архитектуру процессора STM32F4XX с ядром ARM Cortex-M4

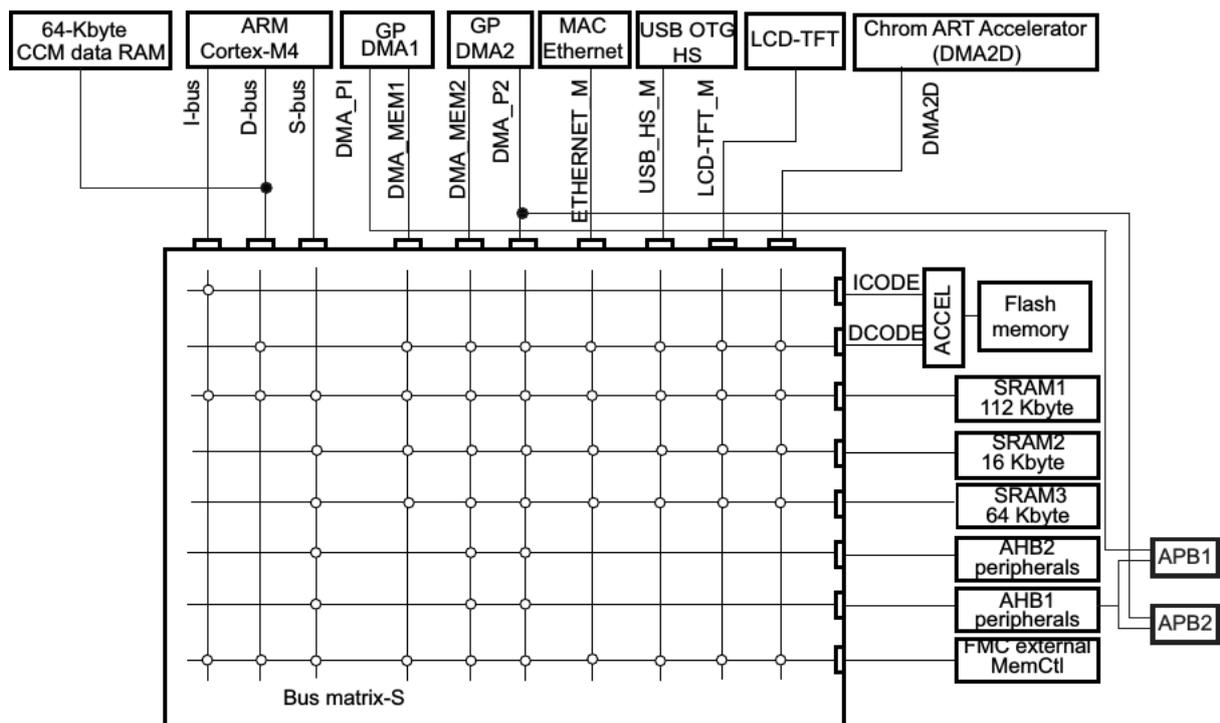


Рисунок 3.4: Архитектура микроконтроллера STM32F4XXX [10].

Для четвертой категории процессоров характерно применение механизмов защиты памяти и большое адресное пространство, что позволяет без особых проблем применять операционные системы реального времени. От третьей категории их также отличает более высокая производительность [2].

В качестве примера процессора четвертой категории для ИУС мы рассмотрим i.MX535 фирмы Freescale (см. рисунке 3.5). Процессор сделан на базе ядра ARM Cortex-A8 и предназначен для использования в мультимедийных системах, системах цифровой обработки сигналов, IP-телефонах, различных мобильных устройствах, таких как планшетные компьютеры. Процессор оптимизирован для работы с операционными системами Android, Windows Embedded Compact 7 и Linux. В список поддерживаемых операционных систем реального времени входят INTEGRITY, QNX NEUTRINO RTOS, Nucleus RTOS, emboss.

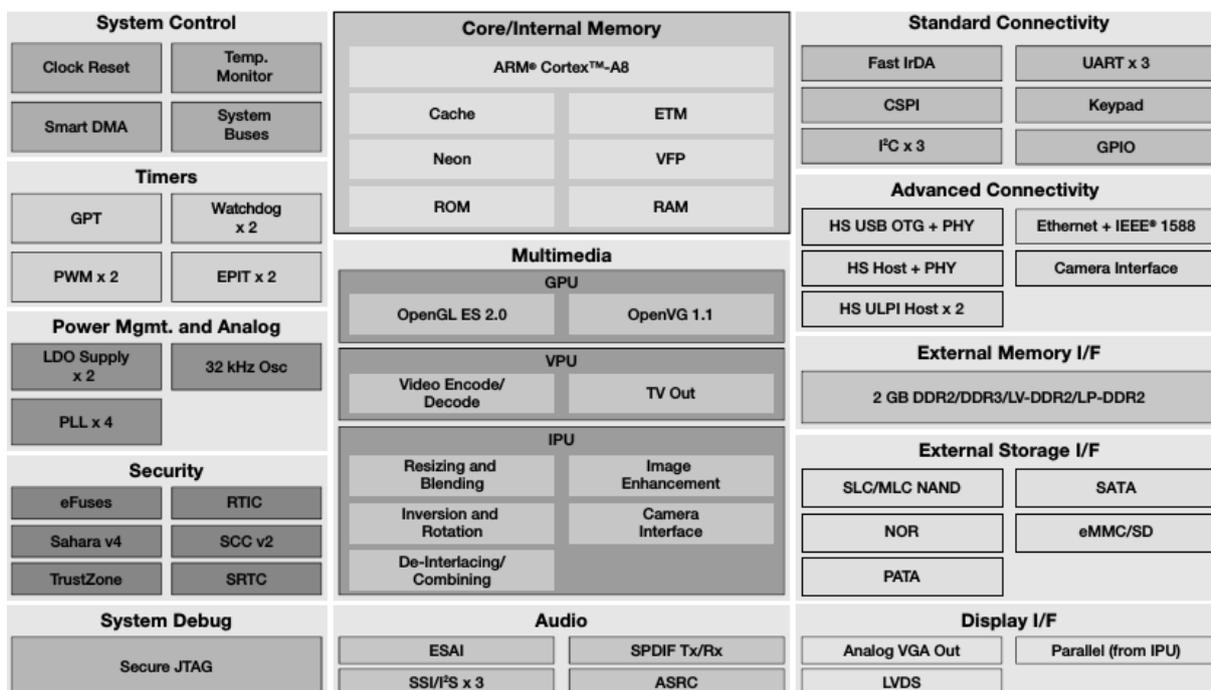


Рисунок 3.5: Структура процессора Freescale iMX535 на базе ARM Cortex-A8 [11].

- Процессор имеет в своем составе следующие блоки:
- Температурный монитор, ПДП, контроллеры шин;
 - Таймеры;
 - Управление энергопотреблением и PLL;
 - Блок для работы с JTAG;
 - Ядро ARM Cortex-A8;
 - Мультимедийный блок;
 - Блок интерфейсов (IrDA, UART, I2C, GPIO, Ethernet, USB и т.п.);
 - Шину для подключения внешней памяти DDR2/DDR3;
 - Интерфейсы для подключения внешней памяти: FLASH, SATA, SD и т. п.;
 - Интерфейсы для подключения дисплеев (HDMI, LVDS).

Статистика использования микроконтроллеров

В настоящее время существует тенденция постепенного вытеснения с рынка маломощных микроконтроллеров (см. рисунок 3.6) [12].

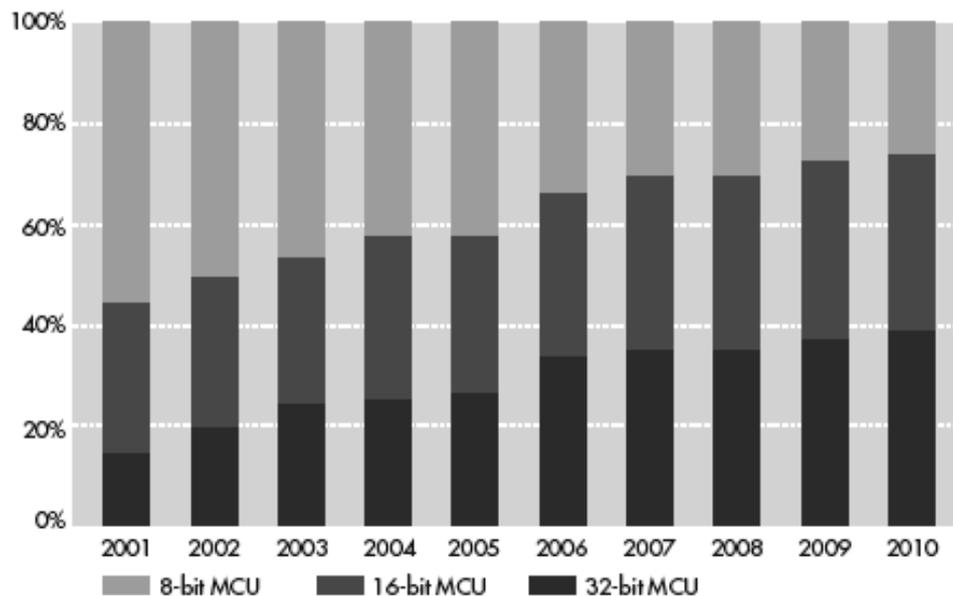


Рисунок 3.6: Относительное количество представленных на рынке 8,16 и 32-битных микроконтроллеров.

Несмотря на большое количество продаваемых на рынке маломощных микроконтроллеров, основная доля разработок приходится на 32-разрядные процессоры и микроконтроллеры (см. рисунок 3.7) [14].

Самые часто используемые тактовые частоты для микроконтроллеров — находятся в диапазонах 10...99 МГц (см. рисунок 3.8).

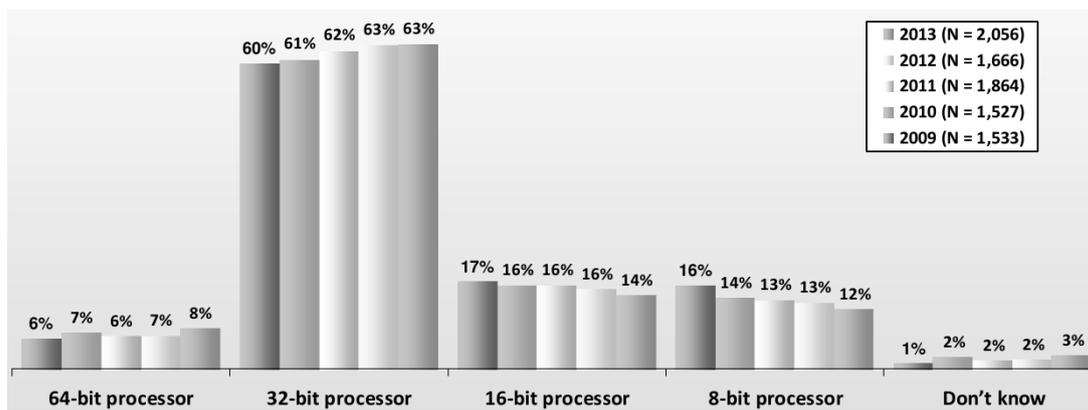


Рисунок 3.7: Используемые разработчиками процессоры [13].

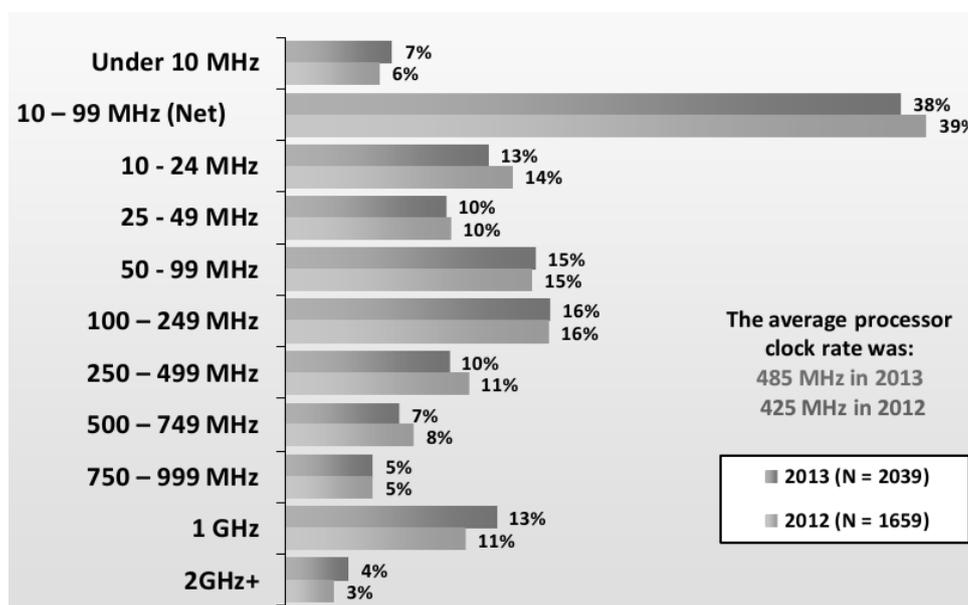


Рисунок 3.8: Используемые тактовые частоты [13].

При выборе микроконтроллера разработчики руководствуются в первую очередь доступностью инструментальных средств, производительностью и стоимостью микроконтроллера (см. рисунок 3.9).

3.2.2 Системы-на-кристалле

Система-на-кристалле (System-on-Chip, SoC) — система, в которой интегрированы процессор (или несколько процессоров, в том числе специализированные), некоторый объем памяти, ряд периферийных устройств и интерфейсов, то есть максимум того, что необходимо для решения конкретного списка задач, поставленных перед системой. Выражение «система на кристалле» не является, строго говоря, термином. Это понятие отражает общую тенденцию к повышению уровня интеграции за счет интеграции функций (см. рисунок 3.10).

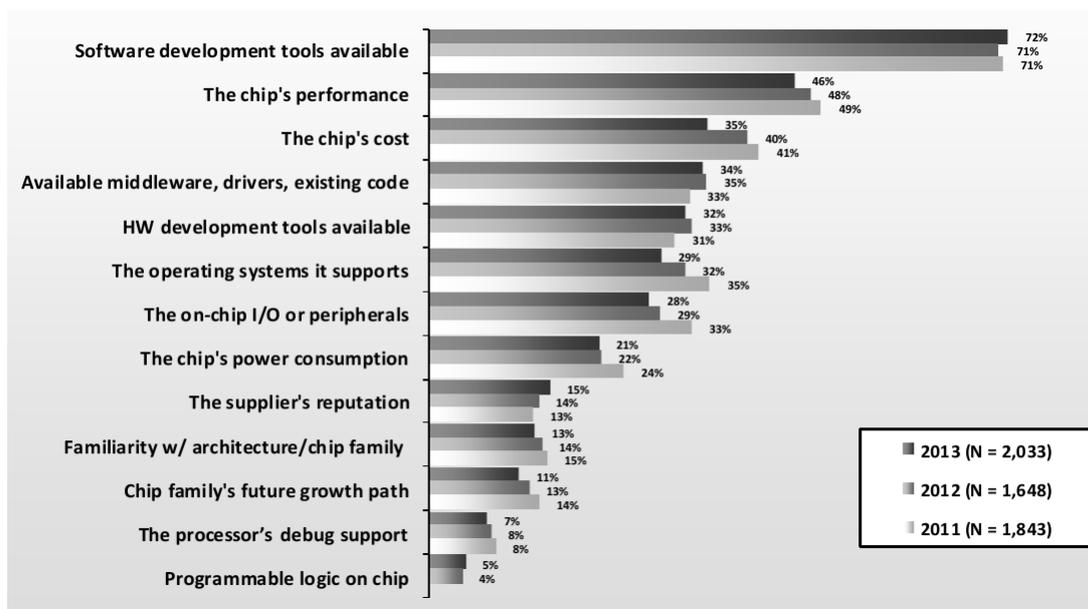


Рисунок 3.9: Факторы, влияющие на выбор микроконтроллера разработчиком [13].

Производительность приборов класса «система-на-кристалле» в значительной мере зависит от эффективности взаимодействия всех встроенных компонентов и от эффективности их взаимодействия с внешним, относительно прибора, миром. В первую очередь это связано с различием в быстродействии встроенных компонентов.

Системы на кристалле обычно состоят из трех основных цифровых системных блоков: процессор, память и логика. Процессорное ядро реализует поток управления, когда каждой управляющей программой однозначно устанавливаются последовательности выполнения операций обработки данных, что позволяет задавать один из возможных алгоритмов работы всей интегральной схемы. Память используется по ее прямому назначению — хранение кода программы процессорного ядра и данных. Наконец, логика используется для реализации специализированных аппаратных устройств обработки и прохождения данных, состав и назначение которых определяются конечным приложением.

Реальная система на кристалле содержит как минимум три перечисленных блока, что исключает применение многочисленных отдельных интегральных схем и реализацию интерфейсов связи между ними (см. рисунок 3.11).

Однокристалльное конфигурируемое или программируемое решение допускает оперативное изменение своей внутренней аппаратной структуры и конечного предназначения как на этапе производства, так и в полевых условиях, непосредственно в проекте. Такие интегральные схемы были отнесены к группе изделий системного уровня интегра-

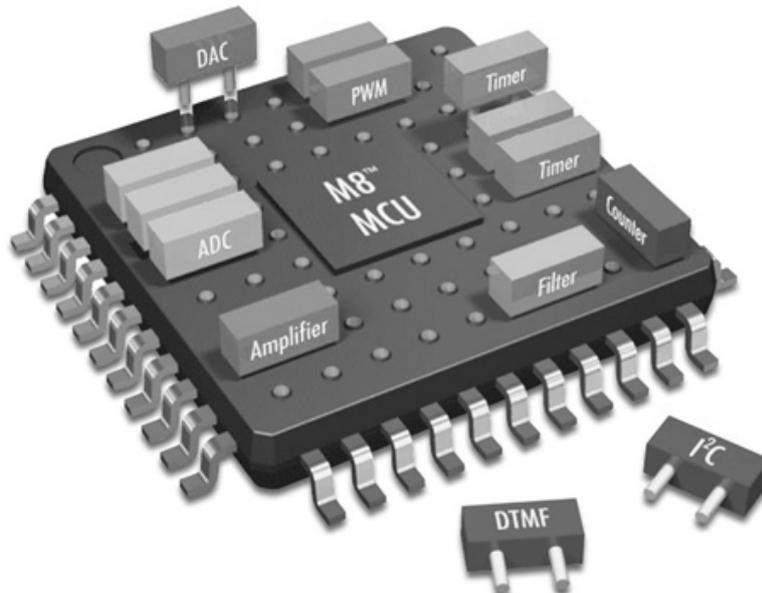


Рисунок 3.10: Интеграция функций в СнК.

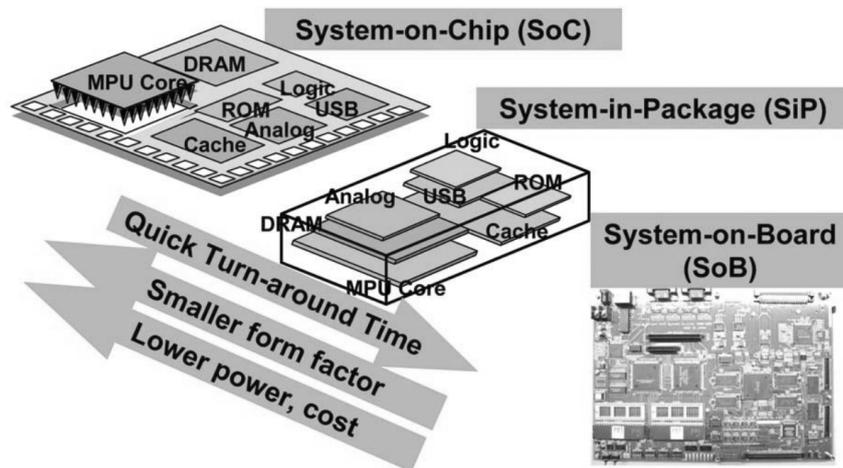


Рисунок 3.11: SoC, SiP, SoB [11].

ции, но получили другое название — Configurable System on a Chip или CSoC. Поскольку термин CSoC не стандартизирован, то существуют и другие названия изделий этого класса — System on Programmable Chip (SoPC), Programmable System on a Chip (PSoC) или просто SoC, что определяется вкусом и желаниями конкретного производителя микросхем [7].

3.3 Программируемые логические интегральные схемы

Программируемая логическая интегральная схема (PLD, Programmable Logic Device) — электронный компонент, состоящий из логических ячеек и конфигурируемой схемы соединений. Основное назначение — построение реконфигурируемых цифровых схем. В отличие от обычных интегральных схем, функциональность не определяется на этапе изготовления раз и навсегда и может создаваться и изменяться конечным пользователем (инженером), исходя из своих нужд.

ПЛИС представляет собой множество однотипных логических элементов, соединяемых с помощью специальных коммутационных матриц. Соединение и инициализация элементов осуществляется посредством бинарного образа, загружаемого в конфигурационную память ПЛИС. Файлы конфигурации (бинарные образы) генерируются с помощью САПР производителя конкретной ПЛИС и являются его интеллектуальной собственностью. На аппаратной базе ПЛИС можно реализовывать системы на кристалле. Описания проектов возможно делать как на языках структурно-функционального описания аппаратуры (Verilog, VHDL), так и при помощи более высокоуровневых языков, таких как SystemC [7].

4 Функциональные блоки микроконтроллеров и СнК

В данном разделе будет рассмотрены основные механизмы современных процессоров, микроконтроллеров и СнК, позволяющие поддерживать реальный масштаб времени. Механизмом реального времени мы будем называть набор аппаратно-программных средств, обеспечивающих возможность соблюдения требований реального масштаба времени.

4.1 Аппаратные прерывания

Система прерываний является важной частью подавляющего большинства современных микропроцессоров, микроконтроллеров и систем на кристалле. В процессорах выполненных в рамках модели фон-Неймана, прерыванием называют прекращение выполнения текущей команды или последовательности команд для обработки некоторого события обработчиком прерывания, с последующим возвратом к выполнению прерванной программы.

Система прерываний обеспечивает:

1. Обработку событий;
2. Организацию приоритетной обработки событий в рамках заданного способа планирования (например, короткие события вперед или срочные события вперед). Наличие прерываний не гарантирует своевременной обработки события в том случае, когда несколько событий происходит в один момент времени.
3. Организацию условий для реализации модели вычислений, отличной от модели фон-Неймана (например, сети процессов Кана [15]).

Система прерываний является совокупностью аппаратных и программных средств. Аппаратные средства системы прерываний пред-

ставлены контроллером прерываний. Как правило, контроллер прерываний интегрирован в процессор. Контроллер прерываний обеспечивает следующие функции:

1. Принимает запрос прерывания, возникший в результате какого-либо события;
2. Определяет приоритет прерывания;
3. Сообщает процессору о необходимости обработки наиболее приоритетного прерывания.

С программной точки зрения система прерываний представлена как совокупность регистров, через которые можно настроить контроллер прерываний и множество векторов прерываний. Таблица векторов прерываний представляет собой обычно область памяти, каждая ячейка которой отвечает за определенный вектор прерываний. В зависимости от типа процессора, в такие ячейки могут записываться адреса обработчиков прерываний (Interrupt Handler) или команды безусловного перехода.

Система прерываний работает следующим образом:

1. При возникновении события вырабатывается запрос на прерывание.
2. Контроллер прерываний обрабатывает запрос.
3. Из всех запросов на прерывание контроллер прерываний выбирает самый приоритетный и передает номер прерывания процессору.
4. Процессор приостанавливает выполнение текущей программы, если прерывания разрешены.
5. Из таблицы векторов прерываний по номеру прерывания извлекается адрес программы-обработчика.
6. Программе обработчику передается управление.
7. Программа обработчик стартует и первым делом сохраняет в стеке все регистры процессора, которые она может затронуть в процессе исполнения.
8. Выполняется код обработчика прерываний.

9. Из стека восстанавливаются все необходимые регистры.
10. Производится возврат из прерывания, управление передается прерванной программе.

4.2 Таймер

Таймер является средством отсчета интервалов времени и источником прерывания, сообщающего о прекращении заданного временного интервала. Работа таймера основана на двоичном счетчике, в который можно записывать значения. Двоичный счетчик может быть инкрементирующий или декрементирующий. Когда при очередной операции инкремента или декремента в счетчике оказывается ноль, таймер выставляет запрос на прерывание.

Способы применения таймера следующие:

1. Подсчет временных интервалов;
2. Организация периодической обработки событий;
3. Организация систем разделения времени (переключателей задач).

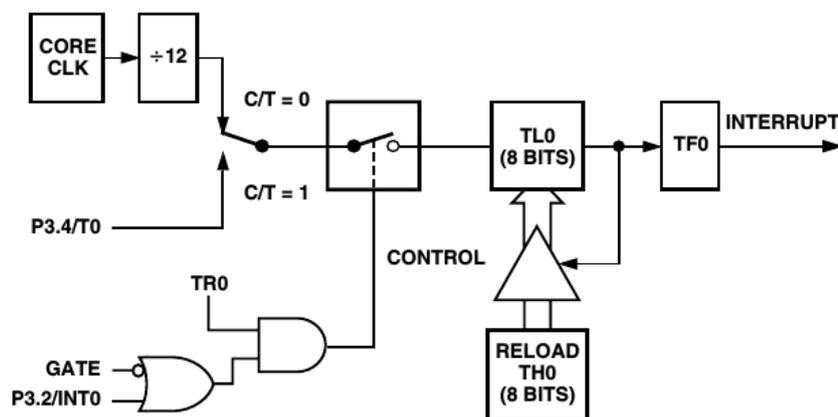


Рисунок 4.1: Таймер микроконтроллера ADuC812 (Timer/Counter 0, Mode 2) [9].

Рассмотрим работу таймера на примере таймера микроконтроллера ADuC812 (см. рисунок 4.1). В данном примере рассматривается таймер 0, работающий в режиме 8-разрядного таймера с перезагрузкой.

Тактовый сигнал CORE CLK попадает в делитель на 12. К примеру, если на входе у нас тактовая частота 12 МГц, то на выходе делителя

мы получим 1 МГц. Следующий делитель — 8-разрядный регистр TLO. Когда регистр TLO переполняется, происходит установка бита TFO и автоматическая перезагрузка таймера из регистра TNO. Если прерывания от таймера разрешены, то каждое переполнение таймера будет вызывать прерывание. Максимально возможный период при частоте 12 МГц будет равен 256 микросекундам.

4.3 Часы реального времени

Часы реального времени (RTC) являются механизмом предоставляющим информацию о точном астрономическом времени. Как правило, часы реального времени имеют следующие свойства:

1. Возможность установки и чтения текущего времени и даты. В часах реального времени есть регистры для записи и чтения данных.
2. Возможность сохранения хода при выключенном питании. Как правило, у всех часов реального времени есть возможность подключения дополнительной батареи, а сами часы сделаны таким образом, чтобы потреблять минимум энергии.



Рисунок 4.2: а) часы реального времени PCF8583 с интерфейсом I²C, б) часы реального времени MCP795 с интерфейсом SPI.

Часы реального времени могут быть встроены в микроконтроллер или быть отдельной микросхемой (см. рисунок 4.2).

Для работы часов реального времени нужен кварцевый резонатор с частотой 32768 Гц (см. рисунок 4.3).

4.4 Сторожевой таймер

Сторожевой таймер является средством предотвращения зависаний и закликиваний программ в ИУС. Как правило, сторожевой таймер реализуется в виде счетчика, на выход которого заведен сигнал

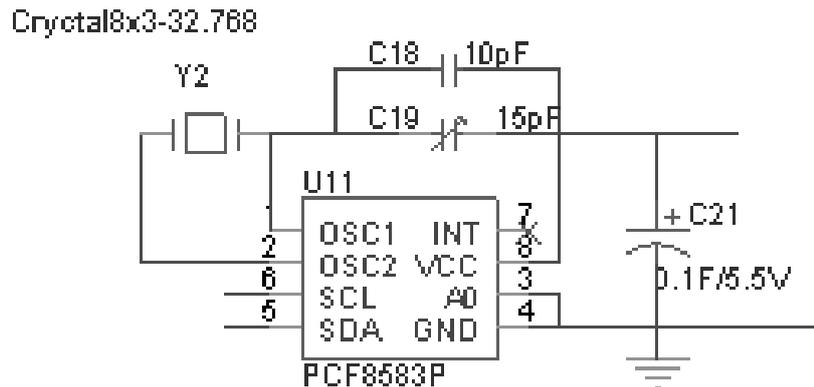


Рисунок 4.3: Схема подключения PCF8583

RESET. У сторожевого таймера есть один вход, активный сигнал на котором приводит к перезаписи в счетчик нового значения. В более сложных сторожевых таймерах можно выбирать период работы (чем больше записываемая константа, тем дольше сторожевой таймер не перезапустится).

Принцип работ сторожевого таймера следующий:

1. В регистр сторожевого таймера заносится число (или производится его сброс через специальный вход).
2. Один раз в заданный период времени сторожевой таймер декрементирует число.
3. Если сторожевой таймер досчитает до нуля, то вырабатывается сигнал RESET и система перезапускается.
4. Для того чтобы перезапуска не было, необходимо эпизодически перезаписывать значение в сторожевом таймере.

Рассмотрим работу сторожевого таймера на примере микросхемы MAX6369 (см. рисунок 4.4). Входы SET0..SET2 позволяют выставить задержку срабатывания сторожевого таймера в диапазоне от 1 мс (на всех входах «0») до 60 сек (на всех входах «1»).

Вход WDI предназначен для перезагрузки внутреннего счетчика сторожевого таймера. Если сформировать импульс на этом входе, сторожевой таймер начнет отсчет нового периода. Если импульс на входе WDI задержать, сторожевой таймер выдаст на выход WDO импульс отрицательной полярности, который можно использовать как сигнал RESET. После изменения состояния сигналов на входах SET0..SET2 сторожевой таймер некоторое время ($t_{SETUP} + t_{DELAY}$) не активен (см. рисунок 4.5).

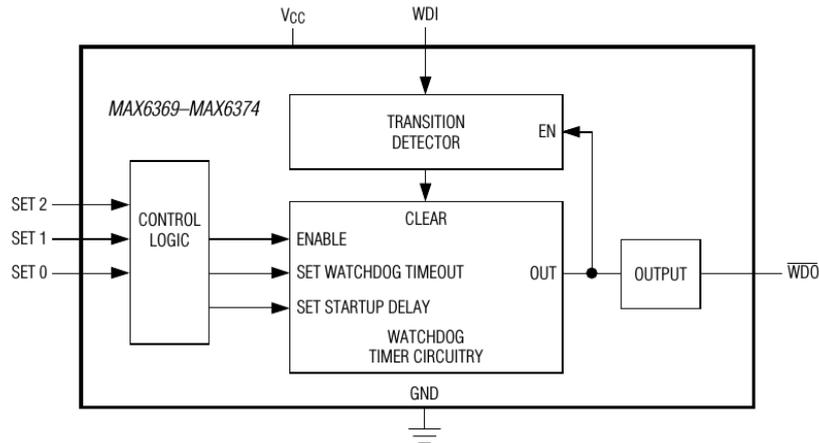


Рисунок 4.4: Сторожевой таймер MAX6369 [16].

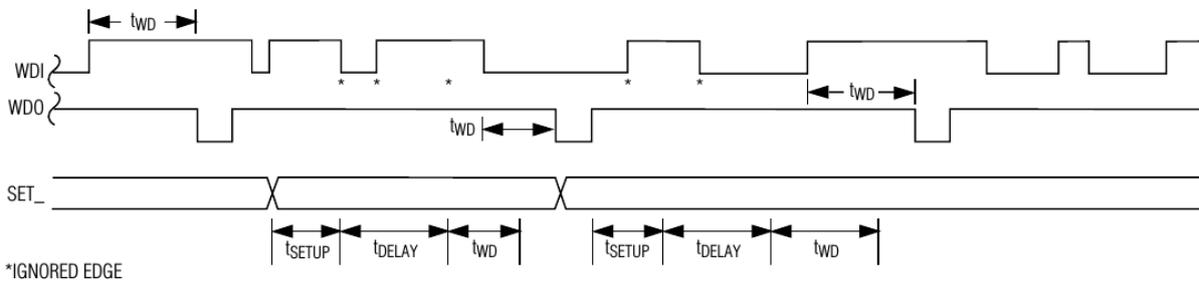


Рисунок 4.5: Временная диаграмма работы сторожевого таймера [16].

4.5 Система контроля питания

Система питания является одним из важнейших элементов системы реального времени. К сожалению, во встроенных системах не всегда удастся добиться качественного вторичного электропитания при использовании автономных источников энергии (аккумуляторов, химических элементов питания и т. п.), бортовой сети при наличии большого количества помех. Система контроля питания предназначена для обеспечения надежного функционирования микроконтроллера в условиях нестабильного питающего напряжения [2].

При включении питания напряжение питания некоторое небольшое время скачкообразно меняется (см. рисунок 4.6). Для того чтобы процессор и другие аппаратные средства не перезапускались на время прохождения переходных процессов, система контроля питания вырабатывает сигнал RESET. Как известно, во время активного уровня данного сигнала все выводы микросхем переходят в высокоимпедансное (Z) состояние, то есть состояние, в котором выводы всех микро-

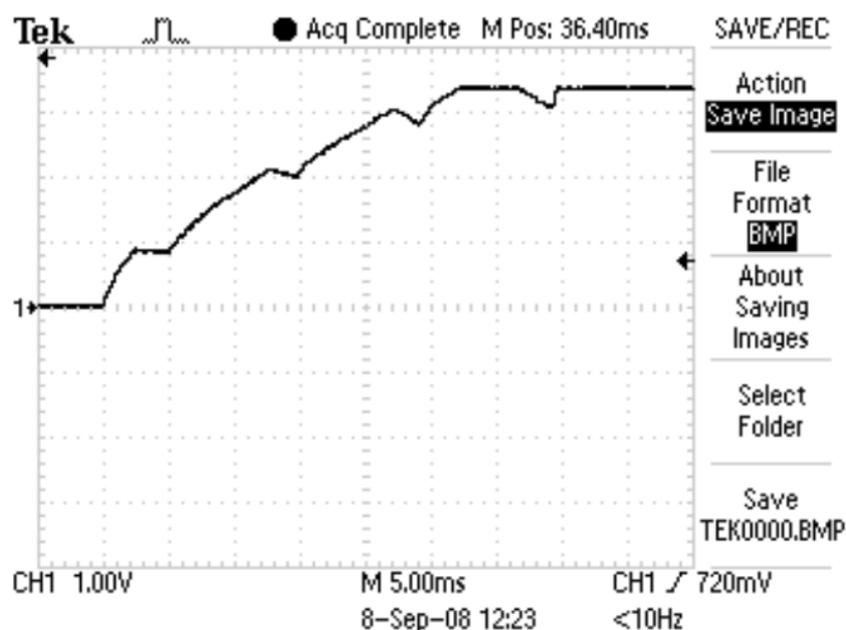


Рисунок 4.6: Осциллограмма шины питания при включении контроллера [2].

схем имеющих вход RESET имеют высокое сопротивление и практически изолированы от схемы.

В процессе работы система контроля питания постоянно проверяет уровень напряжения в цепях питания. Если уровень напряжения отклоняется от заданной величины, система контроля питания выдаст сигнал, который можно интерпретировать как прерывание. Обработчик прерывания может корректно завершить работу встраиваемой системы, например, при внезапном пропадании питания. В обработчике прерывания нужно попытаться сохранить в энергонезависимой или обычной памяти контроллера текущее состояние прикладной программы, чтобы после возобновления подачи электроэнергии продолжить работу с прерванного места.

4.6 Тактовый генератор

Схемотехника, лежащая в основе современных микроконтроллеров, позволяет им работать в широком диапазоне частот, вырабатываемых тактовым генератором. Для обеспечения гибкого управления тактовой частотой используются делители и умножители частоты. Для управления тактовой частоты используют фазовую автоподстройку частоты (ФАПЧ, Phase-locked loop, PLL).

Фазовая автоподстройка частоты (ФАПЧ) — система автоматического регулирования, подстраивающая частоту управляемого генера-

тора так, чтобы она была равна частоте опорного сигнала. Регулировка осуществляется благодаря наличию отрицательной обратной связи. Выходной сигнал управляемого генератора сравнивается на фазовом детекторе с опорным сигналом, результат сравнения используется для подстройки управляемого генератора [2]. Блоки управления тактовой частотой являются конфигурируемыми и позволяют управлять не только тактовой частотой центрального процессора, но и отдельно регулировать тактовые частоты внутренних периферийных устройств и внешней шины.

4.7 Средства понижения энергопотребления

Практически все современные микроконтроллеры имеют встроенные средства понижения энергопотребления, позволяющие отключать не используемые в данный момент блоки, понижать тактовую частоту процессора и переходить в различные режимы сна.

4.8 FLASH память

Встроенная FLASH память в основном предназначена для хранения программ и сравнительно больших объемов данных. Объем памяти, в зависимости от типа микроконтроллера, может достигать от единиц до сотен килобайт. FLASH память не позволяет перезаписывать отдельные байты. Адресное пространство памяти разбито на участки по аналогии с жесткими дисками, называемыми секторами. Для записи новой информации необходимо с помощью специальной команды стереть целый сектор или все сектора сразу.

Один из секторов называется загрузочным. Обычно он располагается в той части адресного пространства микроконтроллера, с которого стартует процессор. Адрес старта — очень важный параметр, его обязательно нужно знать для всех используемых микроконтроллеров. Необходимо помнить, что количество циклов записи и стирания ограничено и достигает десятков тысяч – миллионов раз, и, если вы собираетесь использовать FLASH память для хранения данных, то это необходимо иметь в виду.

Для обеспечения операции стирания и записи используются команды, отправляемые по специальным адресам адресного пространства FLASH памяти [2].

4.9 Энергонезависимая конфигурационная память

Встроенная энергонезависимая память (non-volatile memory) создается по разным технологиям. EEPROM обычно имеет ограничение на количество циклов записи в десятки тысяч – миллионы раз. Если энергонезависимая память является сегнетоэлектрической (FRAM), то количество циклов записи может быть гораздо большим (миллиарды циклов записи). Встроенная энергонезависимая память предназначена для хранения сравнительно небольшого (десятки – сотни байт) количества информации (например, конфигурационной). Этот блок обычно присутствует в маломощных микроконтроллерах для уменьшения необходимого количества микросхем и соединений на плате, с целью удешевления и снижения энергопотребления.

Доступ к ячейкам памяти байтовый. Для чтения и записи обычно используются регистры специального назначения [2].

4.10 Контроллер прямого доступа к памяти

Контроллер прямого доступа к памяти (ПДП, Direct Memory Access, DMA) — блок, позволяющий разгрузить центральный процессор во время пересылок между устройством ввода-вывода и памятью. Обычно блоками ПДП снабжаются сравнительно мощные модели микроконтроллеров. В основном ПДП используется для взаимодействия памяти с устройствами ввода-вывода, которые могут создать большой поток данных: сетевыми контроллерами, UART, ЦАП и АЦП. Как правило, блок ПДП программисты встроенных систем, особенно начинающие, стараются обходить стороной, считая его излишне сложным. На самом деле, использование ПДП не сложнее использования системы прерываний. Эффект от применения ПДП состоит в снятии нагрузки с центрального процессора и значительного увеличения суммарной производительности контроллера [2].

4.11 Устройство захвата-сравнения

Устройство захвата-сравнения предназначено для измерения периодов и длительностей импульсов, скважности, а также для генерации импульсов. С помощью этого модуля можно генерировать импульсы с широтно-импульсной (ШИМ) или фазовой модуляцией (ФМ).

4.12 Порт ввода-вывода

Порт ввода-вывода — это точка подключения микроконтроллера к внешним устройствам ввода-вывода. Порты бывают дискретные и аналоговые. Дискретный порт ввода-вывода позволяет формировать и принимать сигналы, которые кодируют логические значения «0» и «1». Логическому нулю соответствует напряжение, близкое к нулю, а логической единице — напряжение, близкое к напряжению питания. Напряжение, примерно соответствующее средней части диапазона «питание — корпус», при считывании дает непредсказуемый результат.

Каждый дискретный порт можно запрограммировать на вход или на выход. Кроме этого, во многих микроконтроллерах можно указать, будет ли использоваться порт по схеме с открытым или закрытым коллектором.

Аналоговый порт ввода-вывода предназначен для работы с аналоговым (непрерывным) сигналом. В отличие от дискретного сигнала, принимающего всего два значения, напряжение аналогового сигнала может иметь любое значение (в определенных пределах) и меняться во времени. Пример аналогового сигнала — синусоида. Такую форму, например, имеет электрическое напряжение в сети 220В 50Гц.

Для оцифровки аналогового сигнала используют аналого-цифровой преобразователь (АЦП). Для обратного преобразования цифрового кода в аналоговый сигнал — цифро-аналоговый преобразователь (ЦАП). Часто в микроконтроллерах вместо ЦАП используют модули захвата-сравнения, с помощью которых реализуют широтно-импульсный модулятор (ШИМ). Аналоговый сигнал получается на выходе после обработки сигнала, полученного с выхода ШИМ с помощью интегрирующей схемы.

В большинстве микроконтроллеров каждый вывод может выполнять две или три различные функции, например: дискретный порт ввода-вывода, аналоговый выход и последовательный канал. С помощью специального конфигурационного регистра можно выбрать нужную функцию. С точки зрения безопасности аппаратуры, лучше всего производить настройку портов ввода-вывода сразу после старта микроконтроллера [2].

Для примера рассмотрим устройство порта ввода-вывода микроконтроллера STM32F4XX (см. рисунок 4.7).

Как видно на рисунке 4.7, каждый порт может работать на ввод, вывод, применяться в качестве порта общего назначения или реализовывать одну из альтернативных функций.

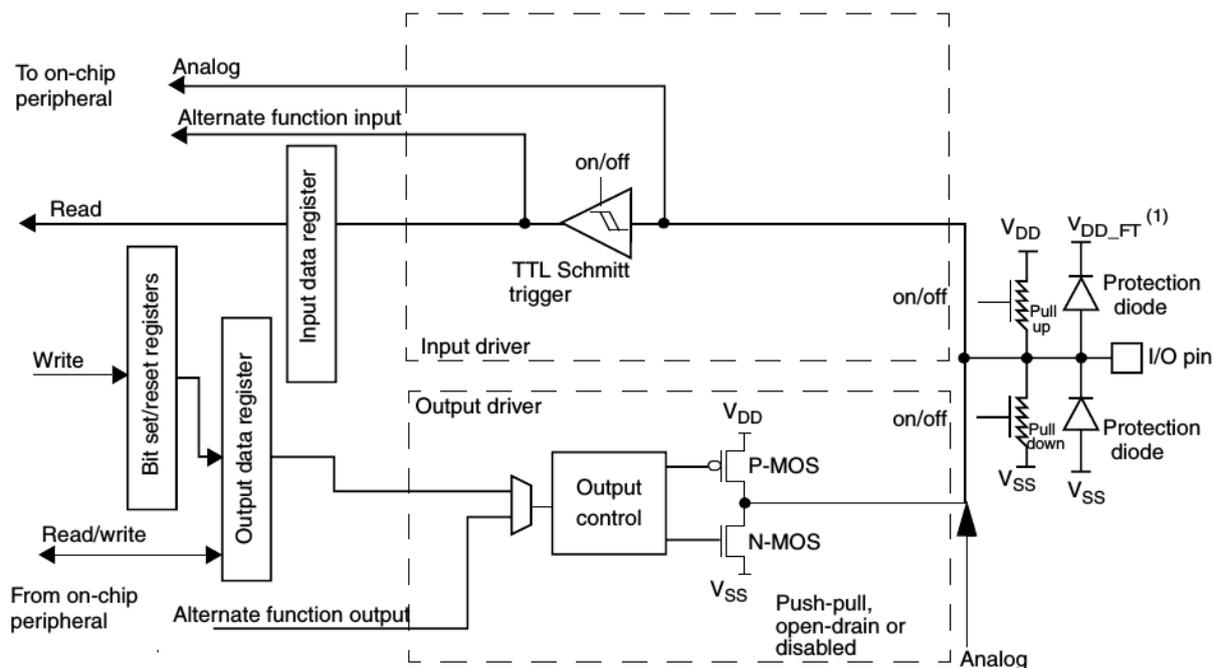


Рисунок 4.7: Порт ввода-вывода микроконтроллера STM43F4XX [10].

4.13 Аппаратные интерфейсы ИУС

4.13.1 Интерфейс I²C

Интерфейс I²C разработан фирмой Philips в 1980 году для связи микросхем друг с другом [17]. I²C расшифровывается, как Inter-Integrated Circuit и предназначен для соединения микроконтроллера с различными устройствами, расположенными в непосредственной близости друг от друга: ЖКИ, расширителями ввода-вывода, часами реального времени, энергонезависимой памятью, различными датчиками и т. д.

К достоинствам I²C можно отнести следующее:

- Гибкость при проектировании различных устройств, возможность замены устаревших микросхем на новые без переработки печатной платы.
- Существенная экономия места на печатных платах и упрощение трассировки проводников, так как I²C требует для передачи данных всего три провода, включая землю.
- Возможность подключения к линии нескольких главных устройств.
- Встроенные в контроллеры интерфейса средства подавления помех.

- Возможность подключения сравнительно большого количества ведомых устройств (емкость шины не должна превышать 400 пФ).

Таблица 4.1: Версии интерфейса I²C [17].

Наименование режима	Частота
Standard-mode	100 кГц
Fast-mode (Fm+)	400 кГц
Fast-mode Plus (Fm+)	1 МГц
High-speed mode (Hs)	3.4 МГц
Ultra Fast-mode (UFm)	5 МГц

Интерфейс I²C является последовательным, двунаправленным интерфейсом с шинной топологией. Для передачи данных используются порты с открытым стоком (коллектором). Уровню логического «0» соответствует напряжение около 0 В, а уровню логической «1» соответствует напряжение, близкое к напряжению питания контроллера I²C. Каждое устройство имеет уникальный адрес, задаваемый 7-битовым числом. У большинства контроллеров с интерфейсом I²C адрес задается перемычками на выводах (см. рисунок 4.8).



Рисунок 4.8: а) Часы реального времени PCF8583 [18] и EEPROM AT24C02 [19], б) Входы A0 и A0, A1,A2 предназначены для задания адреса устройства I²C.

Максимальная частота линии синхронизации SCL зависит от возможностей аппаратного обеспечения и версии используемого интерфейса (см. таблицу 4.1).

На шине I²C может присутствовать несколько главных и подчиненных устройств (см. рисунок 4.9). Передача данных осуществляется в

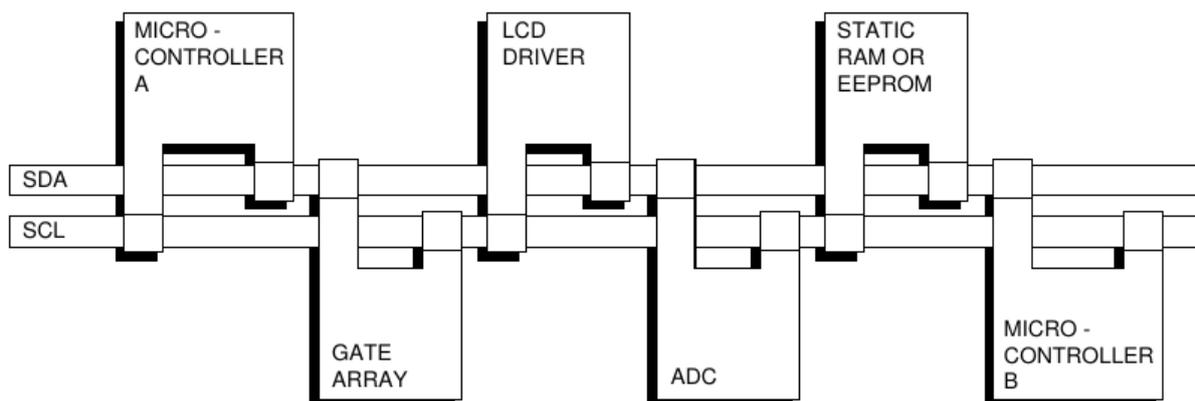


Рисунок 4.9: Шина I²C с несколькими главными и несколькими подчиненными устройствами [17].

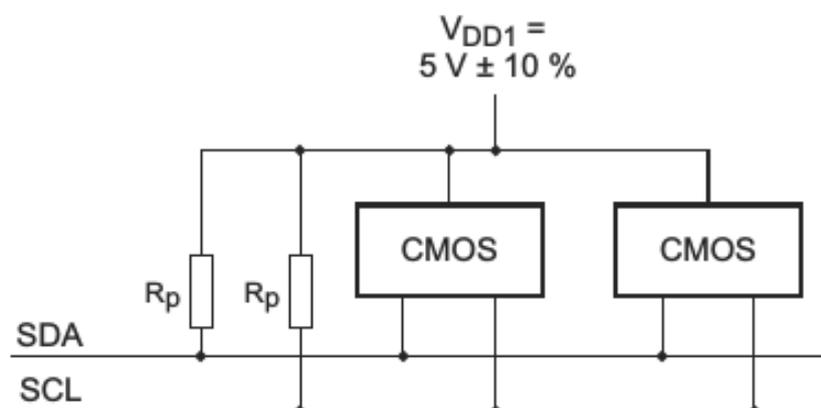


Рисунок 4.10: Подтягивающие резисторы R_p на сигналах SDA и SCL.

полудуплексном режиме. В один момент времени передачу может осуществлять только одно устройство. Передачу всегда начинает главное устройство. Подчиненное устройство может только отвечать на запрос.

Принцип работы большинства I²C устройств основан на идее, что все подчиненные устройства являются устройствами памяти, с которыми можно осуществлять операции чтения и записи данных. На самом деле, стандарт на I²C не вносит таких ограничений, и производитель устройств может интерпретировать данные передаваемые I²C устройствами разными способами, в зависимости от необходимости.

На физическом уровне интерфейса I²C используется два сигнала: SDA и SCL. Оба сигнала подключаются к шине питания через подтягивающие резисторы (см. рисунок 4.10).

Сигнал SDA отвечает за передачу данных, а сигнал SCL обеспечивает синхронизацию. Данные на линии SDA считаются корректными,

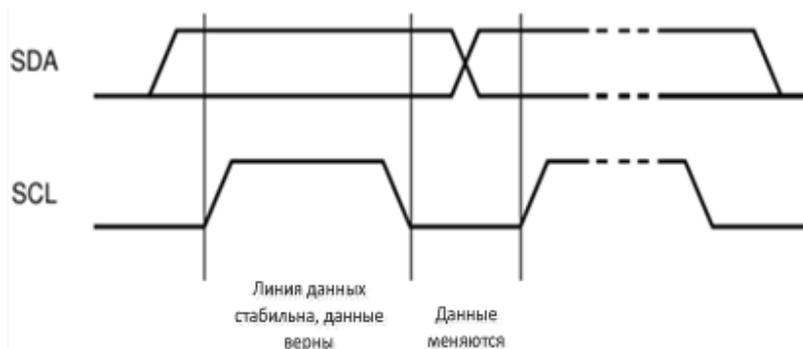


Рисунок 4.11: Принцип передачи данных в I²C.

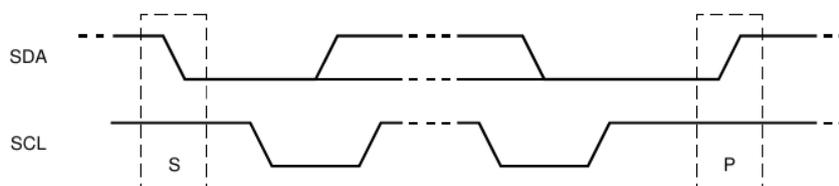


Рисунок 4.12: Состояния «Старт» и «Стоп» на шине I²C.

когда на линии SCL установлен уровень логической единицы (см. рисунок 4.11).

На шине I²C в начале посылки нужно сформировать состояние START, а в конце состояние STOP (см. рис. 4.12).

Состояния START (S) и STOP (P) генерируются главным устройством. Состояние START выглядит как перепад с логической «1» на логический «0» на линии SDA при наличии логической «1» на линии SCL. Состояние STOP выглядит как перепад с логического «0» на логической «1» на линии SDA при логической «1» на линии SCL (см. рисунок 4.12).

По шине I²C данные передаются байтами, как показано на рисунке 4.13. Байты передаются старшим разрядом вперед (MSB). Син-

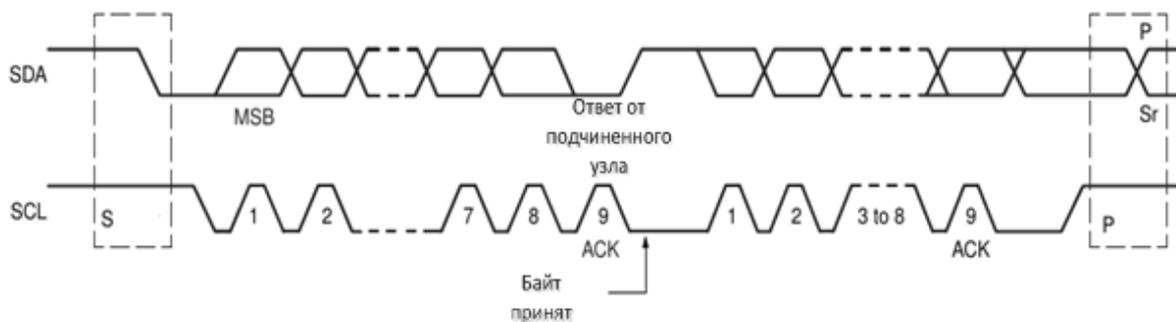


Рисунок 4.13: Передача данных по шине I²C.

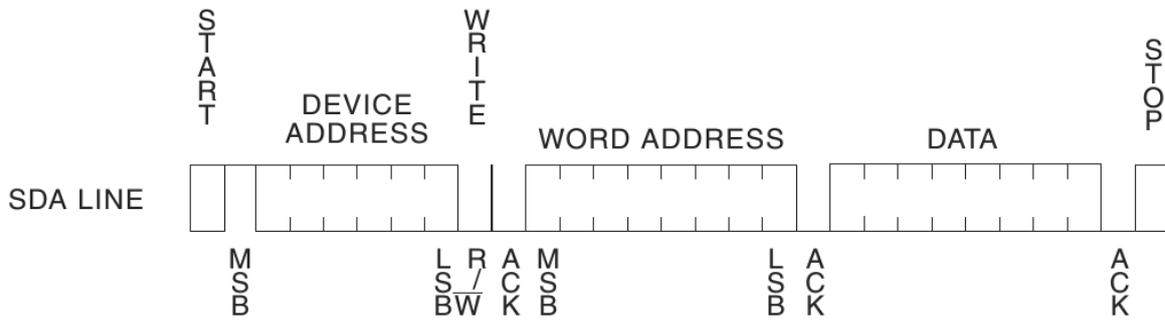


Рисунок 4.14: Запись одного байта в E²PROM.

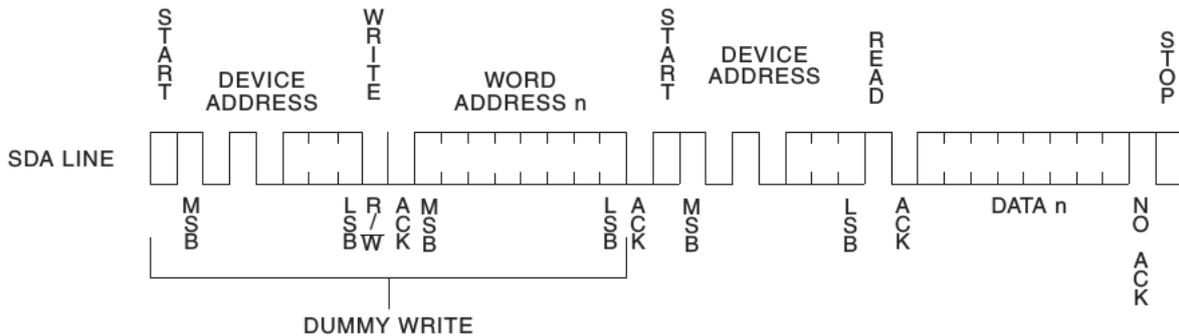


Рисунок 4.15: Чтение произвольного байта из E²PROM [8].

хронизация по линии SCL осуществляется по схеме монтажного «И». После передачи каждого байта главным узлом, подчиненный узел выставляет бит подтверждения ACK, переводя линию в состояние логического «0». После передачи байта, главное устройство должно переключить вывод SDA на вход, чтобы подчиненное устройство смогло выдать бит подтверждения. Если устройств несколько, и они имеют различную скорость работы, главное устройство продолжит передачу лишь тогда, когда самое медленное устройство уберет низкий уровень на линии SCL. Таким образом, скорость работы всего интерфейса I²C определяется скоростью работы самого медленного подчиненного устройства.



Рисунок 4.16: Первый байт послыки по I²C.

В качестве примера рассмотрим порядок чтения и записи байтов в микросхему энергонезависимой памяти (см. рисунок 4.14 и 4.15).

На рисунках 4.14 и 4.15 видно, что после стартового состояния главный контроллер выдает адрес устройства, далее устанавливает бит ЧТЕНИЕ или ЗАПИСЬ. При записи передается адрес ячейки памяти E²PROM и содержимое ячейки (см. рисунок 4.16). При чтении, передается адрес ячейки и ожидается прием байта от E²PROM.

Работа с портами ввода-вывода реализована через макросы SDA-HI, SDA-LO, SCL-HI, SCL-LO. Эти макросы позволяют устанавливать логические «1» и «0» на нужных портах микроконтроллера.

В большинстве современных контроллеров реализован контроллер последовательного канала, в котором есть режим работы с I²C, позволяющий работать как в режиме главного, так и подчиненного узла.

4.13.2 Интерфейс SPI

Интерфейс SPI (Serial Peripheral Interface) предназначен для передачи данных между компонентами на небольшие расстояния, как правило, внутри одного электронного устройства. Интерфейс предложен фирмой Motorola [20].

SPI относится к классу последовательных синхронных интерфейсов. Передача данных осуществляется в обе стороны (полный дуплекс). На физическом уровне сигнал передается в виде последовательности импульсов, логическому «0» соответствует 0 В, логической «1» соответствует напряжению, приблизительно равное напряжению питания. Для передачи данных используется две линии данных, одна линия синхронизации и одна и более линии выбора подчиненного устройства (см. рисунок 4.17).

Так же как и I²C, SPI используется для связи микроконтроллеров с различными контроллерами, ЦАП, АЦП, датчиками и устройствами памяти, начиная от EEPROM и Data FLASH и заканчивая картами памяти SD (см. рисунок 4.18).

Скорость передачи данных по интерфейсу SPI не лимитирована какими либо стандартами и может достигать единиц мегабит в секунду. Достоинствами SPI является:

- Простота реализации;
- Возможность передачи слов любой разрядности;
- Полный дуплекс;
- Высокая скорость передачи (по сравнению с I²C).

К недостаткам SPI можно отнести следующее:

- В системе может быть только одно главное устройство;

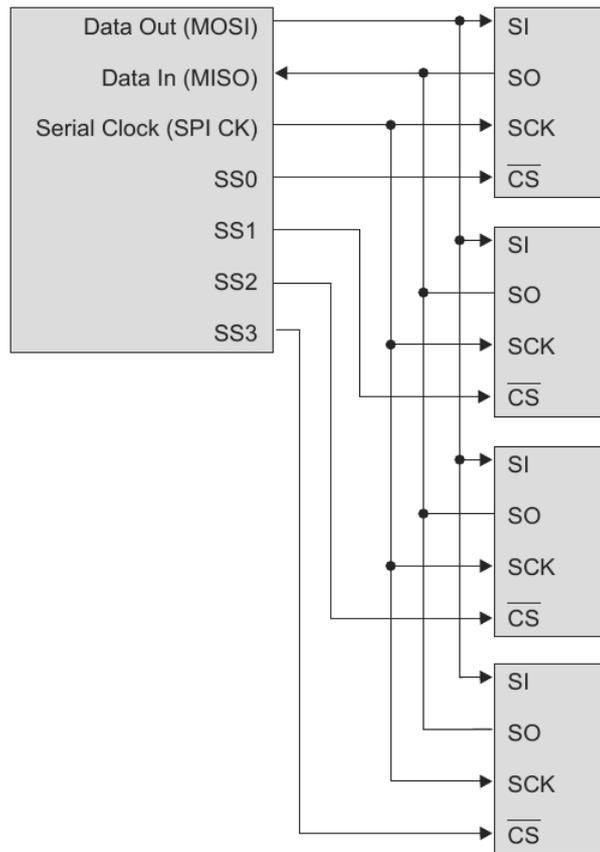


Рисунок 4.17: Интерфейс SPI [21].

- Необходимость в линиях для управления подчиненными устройствами (на каждое устройство — одна линия);
- В отличие от I²C, SPI не имеет средств аппаратного квитирования. Если по какой-либо причине подчиненное устройство перестанет успевать обрабатывать принимаемые данные, главное устройство не будет поставлено об этом в известность и данные будут потеряны. Формат передачи данных показан на рисунке 4.19.

Сигнал SS (Slave Select) имеет активный уровень «0» и предназначен для включения выбранного подчиненного устройства. SS находится в активном состоянии все время, пока идет обмен данными. Все остальные сигналы SS (если они есть), в этот момент времени должны быть состояний «1», то есть остальные подчиненные устройства не активны.

Сигнал SCK формируется главным устройством и предназначен для синхронизации. SCK может иметь разную полярность. Полярность импульсов синхронизации задается специальным битом в контроллере SPI или реализуется программно, если для реализации интерфейса используются обычные порты ввода-вывода. Разумеется, полярность им-



Рисунок 4.18: SD карта с интерфейсом SPI.

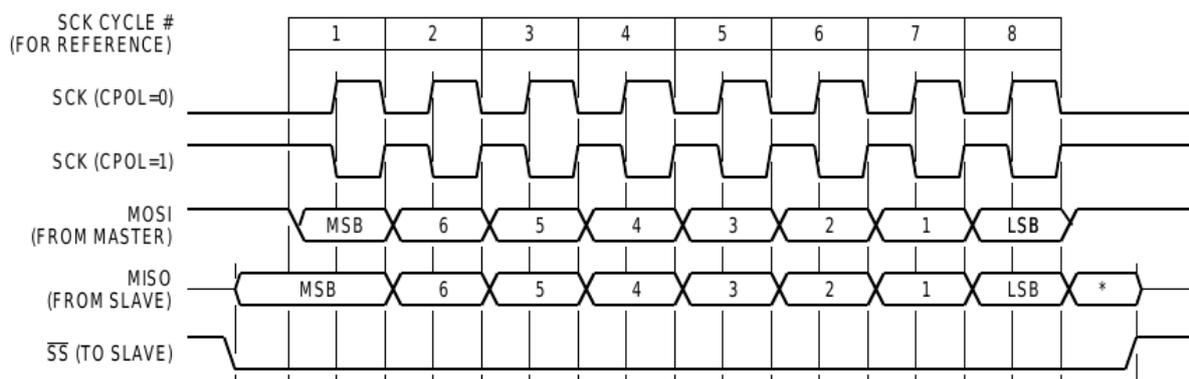


Рисунок 4.19: Формат передачи данных SPI [20].

пульсов синхронизации у главного и подчиненных устройств должна быть одинаковой. Линия *MOSI* является линией, по которой передаются данные от главного к подчиненному устройству. Линия *MISO* является линией, по которой передаются данные от подчиненного устройства к главному.

4.13.3 Интерфейс 1-Wire

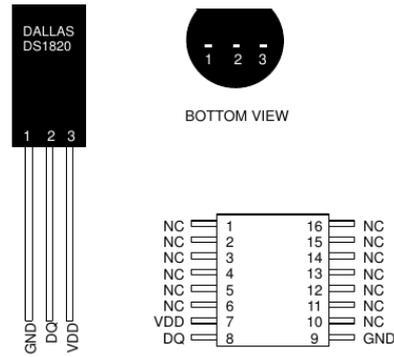
Интерфейс 1-Wire разработан фирмой Dallas Semiconductor и предназначен для подключения различных устройств, таких как датчики температуры, устройства iButton, часы реального времени и т. п., к микроконтроллеру. Сети с интерфейсом 1-Wire называют *MicroLan* [22] (см. рисунок 4.20).

Особенностью данного интерфейса является возможность питания устройств от сети *MicroLan*.

При наличии коротких линий связи для передачи данных и для питания можно использовать один провод. В случае наличия сильных помех или при необходимости создания протяженной линии связи (десятки метров) питание можно подводить отдельным проводом (см. рисунок 4.21).



а)



б)

Рисунок 4.20: Устройства с интерфейсом 1-Wire: а) контактная память (iButton), б) цифровой датчик температуры DS1820.

К достоинствам интерфейса 1-Wire можно отнести возможность работы подчиненных устройств без источников питания и малое количество сигналов. К недостаткам можно отнести низкую скорость работы интерфейса.

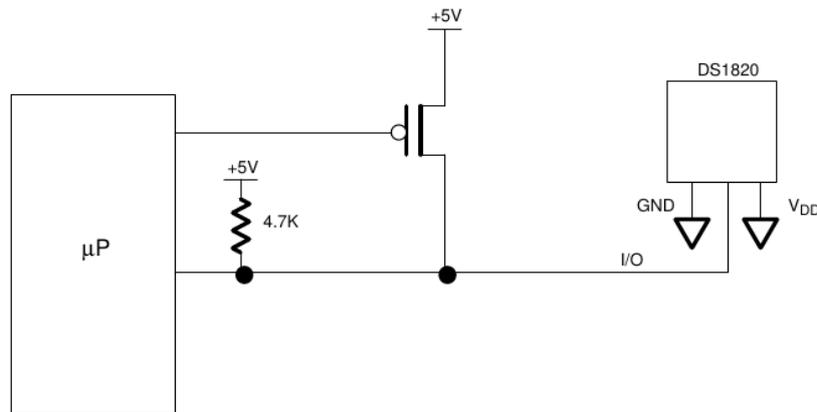


Рисунок 4.21: Схема подключения DS1820 к микроконтроллеру.

Порядок работы с устройствами 1-Wire следующий:

1. Инициализация устройства (см. рисунок 4.22). Главное устройство открывает выходной транзистор, который шунтирует линию данных, обеспечивая на ней логический «0». Через 480...960 мкс главное устройство отпускает линию (закрывает выходной транзистор), обеспечивая на линии логическую «1». Через 15...60 микросекунд подчиненное устройство переводит линию данных в логический «0» на период 60...240 мкс и освобождает линию (переводит линию в логическую «1»). Отрицательный импульс на линии является признаком того, что подчиненное устройство работает и готово отвечать на команды главного устройства.

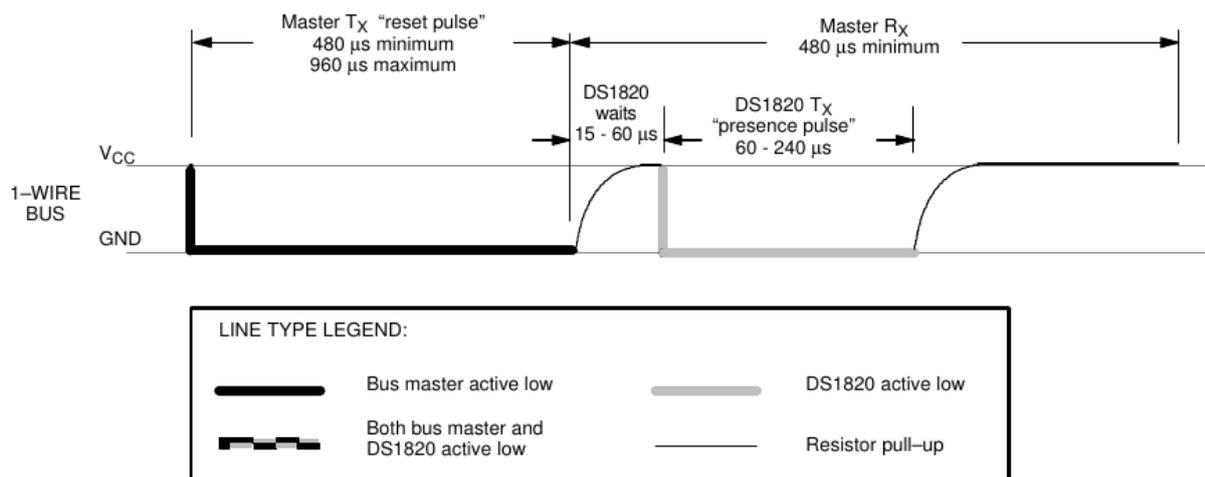


Рисунок 4.22: Порядок инициализации устройства [22].

2. Обмен данными (см. рисунок 4.23).

При передаче данных используется следующий способ кодирования:

1. Передача логического нуля. Главное устройство устанавливает на линии логический «0» на период от 60 до 120 мкс. После чего переводит линию в логическую «1». Время нахождения линии в этом состоянии может длиться от 1 мкс до бесконечности. Подчиненное устройство судит о передаваемом бите на участке, отстоящем на 15 мкс и более от начала сигнала (см. рисунок 4.23).
2. Передача логической единицы. Главное устройство временно устанавливает на линии сигнал логического «0» (от 1 до 15 мкс), после чего переводит линию в состояние логической единицы.
3. Прием логического нуля. Главное устройство временно (1...15 мкс) переводит линию в логический «0». После этого главное устройство переключает порт в режим ввода данных. Если на линии «0», то подчиненное устройство передает «0», если «1», то подчиненное устройство передает «1».

4.14 Отладочные интерфейсы ИУС

В связи с тем, что ИУС, как правило, не имеют собственных средств разработки, отладки и тестирования «на борту», создание про-

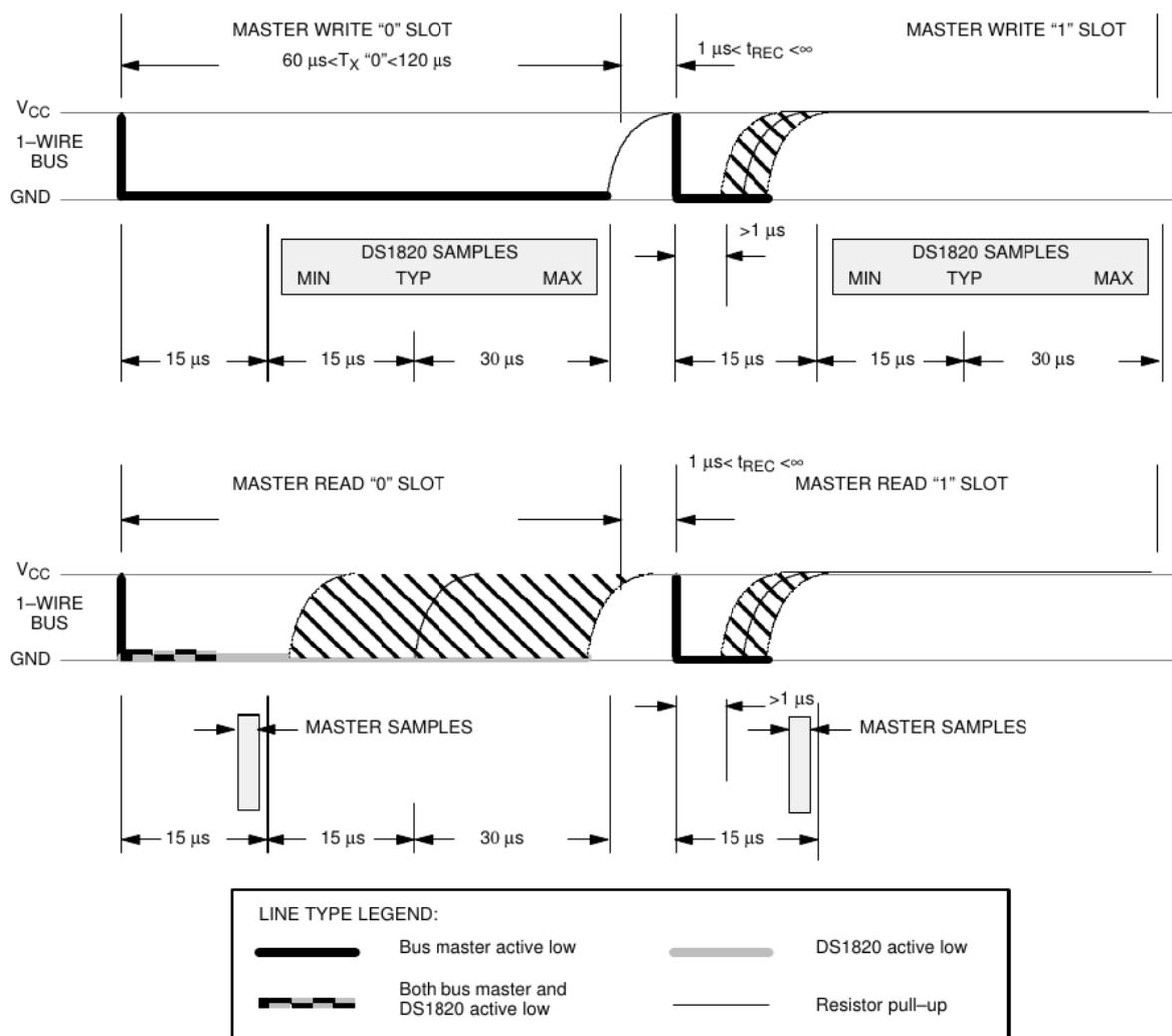


Рисунок 4.23: Порядок обмена данными по шине 1-Wire [22].

граммного обеспечения и его отладка имеют ряд особенностей. В большинстве случаев все современные микроконтроллеры и микропроцессоры снабжают специальными отладочными и загрузочными интерфейсами. На обычном персональном компьютере или рабочей станции разработчик разворачивает специальное инструментальное программное обеспечение, называемое «кросс-система». В состав этой кросс-системы обычно входит интегрированная среда разработки, компилятор, отладчик и программатор. Компилятор преобразует исходные тексты в исполняемый код целевой (ИУС), а не инструментальной машины. С помощью программатора загрузочный модуль доставляется в ИУС и запускается там. Отладочный интерфейс позволяет осуществлять в отладчике просмотр ячеек памяти и пошаговую отладку в привычном виде.

4.15 Встроенный загрузчик

Встроенный загрузчик добавляется в микроконтроллер производителем, находится там всегда (его, как правило, невозможно уничтожить) и позволяет обновлять прошивку микроконтроллера через обычный последовательный канал, без использования дополнительного программного обеспечения со стороны целевой системы и аппаратного обеспечения (программатора) со стороны инструментальной.

5 Конструктивные особенности ИУС

В отличие от систем общего назначения, информационно-управляющие системы могут эксплуатироваться в местах не защищенных от вандализма и воздействий окружающей среды: пыли, твердых предметов, влаги, воды, щелочей и кислот, радиации, низких или высоких температур.

На рисунках (см. рис. 5.1) показаны возможные места использования ИУС: на земле, в воздухе и в космосе.

5.1 Классы защиты по IP

Ниже приведена таблица демонстрирующая классы защиты по коду IP [23] для конструктивов от воздействия твердых предметов и воды. Степень защиты обозначается двумя буквами IP и двумя цифрами. Первая цифра обозначает степень защиты от проникновения твердых механических предметов, вторая цифра показывает степень защиты от воздействия жидкости [2].

5.2 Температурный диапазон компонентов

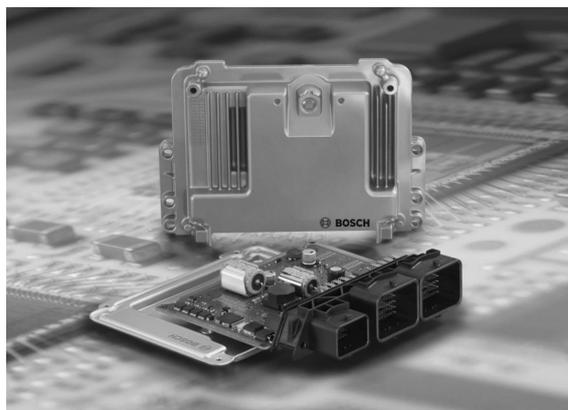
Все аппаратные компоненты ИУС имеют такую характеристику как температурный диапазон. В рамках температурного диапазона производитель компонента гарантирует нормальное функционирование. Температурные диапазоны обычно разбивают на несколько классов. В зависимости от используемых стандартов температурные диапазоны могут отличаться. Примерные диапазоны рабочих температур представлены ниже:

- Для работы в помещении ($0 \dots + 70 \text{ }^\circ\text{C}$) Для промышленного использования ($-40 \dots + 85 \text{ }^\circ\text{C}$);
- Для автомобильной техники ($-40 \dots + 125 \text{ }^\circ\text{C}$);
- Для авиатехники ($-55 \dots + 125 \text{ }^\circ\text{C}$).

Если ИУС находится в температурной зоне, в которой не может нормально функционировать хотя бы один ее компонент, необходимо



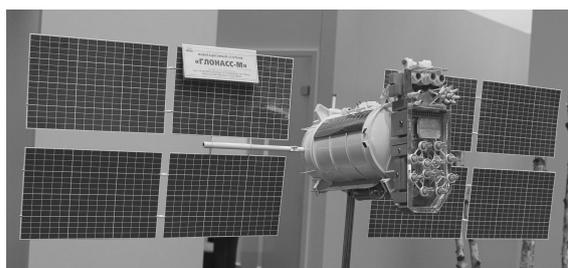
а)



б)



в)



г)

Рисунок 5.1: а) электро-шкаф с оборудованием для работы на улице, б) блок управления двигателем автомобиля, в) истребитель ПАК-ФА, с потолком полета 20 км, г) спутник ГЛОНАСС-М.

осуществлять специальные меры по созданию благоприятных температурных условий. Для холодных условий эксплуатации применяют специальные кожухи с электрическим подогревом, а для горячих — системы охлаждения различного типа (вентиляция, водяное охлаждение, элементы Пельтье, см. рис. 5.3).

Для проверки работоспособности ИУС в различных климатических условиях используют климатические камеры. На рисунке 4 показана климатическая камера BINDER, предназначенная для испытаний на теплостойкость или морозостойкость в диапазоне температур от $-40\text{ }^{\circ}\text{C}$ до $180\text{ }^{\circ}\text{C}$ (см. рис. 5.4).

Таблица 5.1: Классы защиты по IP

Класс	Защита от твердых предметов	Класс	Защита от попадания жидкости
0	Защита отсутствует	0	Защита отсутствует
1	Защита от твердых предметов размером более 50мм	1	Защита от капель воды падающих вертикально
2	Защита от твердых предметов размером более 12мм	2	Защита от капель воды падающих под углом 15 от вертикали
3	Защита от твердых предметов размером более 2,5мм	3	Защищенность от дождя
4	Защита от твердых предметов размером более 1мм	4	Защита от водных брызг
5	Защита от пыли	5	Защита от водных брызг под давлением
6	Полная пылезащищенность	6	Защищенность от волн
		7	Защита от погружения в воду на глубину не более 1м
		8	Защита от затопления (глубина указывается дополнительно, в метрах)

5.3 Защита от влажности и агрессивных сред

Под воздействием влаги металлические части подвергаются коррозии, тогда электронные схемы начинают давать сбои. Для защиты электронных плат применяют специальные защитные лаки, которыми покрывают платы на производстве. Платы покрывают лаком методом погружения или распыления аэрозоли, для нанесения лака может использоваться специальное оборудование (см. рис. 5.5).

При небольших партиях, например при изготовлении прототипов, лак наносят кисточкой. Перед покрытием лаком платы тщательно промывают и обезжиривают. В случае необходимости (например, при за-



а)



б)

Рисунок 5.2: Корпуса фирмы BOPLA [6] с различной степенью защиты по IP: а) класс защиты — IP30, б) класс защиты — IP67.

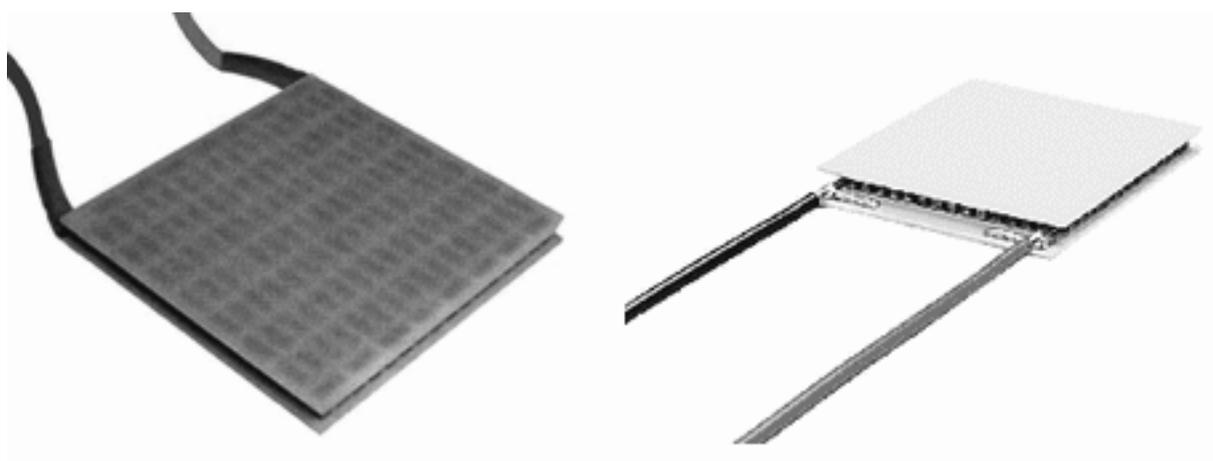


Рисунок 5.3: Элементы Пельтье

мене электронного компонента), лак снимают с помощью специальных растворителей.

Для обеспечения работы устройств в условиях высокой влажности, соляного тумана, агрессивных жидкостей необходимо провести герметизацию корпуса (см. рис. 5.6). Изоляции от внешней среды подвергается корпус, элементы пользовательского интерфейса (кнопки, индикатор и т. п.) и кабельные вводы.

5.4 Защита от механических воздействий

Механические воздействия могут привести к выходу из строя компонентов, трещинам в платах, повреждению конструктивных элементов. Электронные устройства могут подвергаться различным механическим воздействиям:

- Вибрация;
- Удары;
- Линейные ускорения.



Рисунок 5.4: Климатическая камера BINDER [24].

При продолжительном воздействии механических нагрузок внутри конструкции ИУС могут возникнуть резонансные колебания, приводящие к вибрации. Для защиты от механических воздействий применяют следующие способы:

- Резонансная отстройка. Изменяется собственная частота колебаний конструкции за счет введения ребер жесткости, дополнительных точек крепления печатных плат и т. д.
- Демпфирование. Используются материалы, гасящие колебания (например, полиуретан).
- Вибро и ударо изоляция. Для изоляции ИУС от механических воздействий применяют амортизаторы.

Для защиты от вибрации и ударов в конструкциях для жестких условий эксплуатации избегают применять разъемные соединения. Микросхемы для встраиваемых применений обычно выпускаются в корпусах под пайку. В случаях, если без разъемов обойтись не удастся, используют винтовые соединения для увеличения жесткости фиксации (см. рис. 5.7).



Рисунок 5.5: Настольная система дозирования, нанесения влагозащитного покрытия или компаунда PVA 350 [24].

5.5 Защита от электромагнитных помех

Все современные ИУС подвержены воздействию электромагнитных импульсов (ЭМИ). Электромагнитный импульс (ЭМИ) — это возмущение электромагнитного поля, оказывающее влияние на любой материальный объект, находящийся в зоне его действия. Поражающее действие электромагнитного импульса (ЭМИ) обусловлено возникновением наведённых напряжений и токов в различных проводниках. Наиболее уязвимы линии связи, сигнализации и управления. При этом может произойти пробой изоляции, повреждение трансформаторов, порча полупроводниковых компонентов и т. п.

ГОСТ Р 50397-92 определяет электромагнитную совместимость как «Способность технического средства функционировать с заданным качеством, в заданной электромагнитной обстановке и не создавать недопустимых электромагнитных помех другим техническим средствам».



Рисунок 5.6: Герметичный корпус для тяжелых условий эксплуатации.

Электромагнитная помеха (ЭМП) — нежелательное физическое явление или воздействие электрических, магнитных или электромагнитных полей, электрических токов или напряжений внешнего или внутреннего источника, которое нарушает нормальную работу технических средств, или вызывает ухудшение технических характеристик и параметров этих средств.

Методы защиты ИУС от ЭМП и ЭМИ (см. рис. 5.8):

- Экранирование чувствительных к ЭМИ компонентов (помещение всей системы в металлический корпус, экранирование линий связи и отдельных компонентов устройства, заземление экранов).
- Предохранительные цепи в устройствах сопряжения с объектом (УСО) (плавкие предохранители, восстанавливаемые предохранители, грозозащита);

Фильтры, пропускающие полезный сигнал и задерживающие помеху. Для подавления помех в цепях питания цифровых схем используют фильтрующие конденсаторы (см. рис. 5.9). Фильтрующие конденсато-

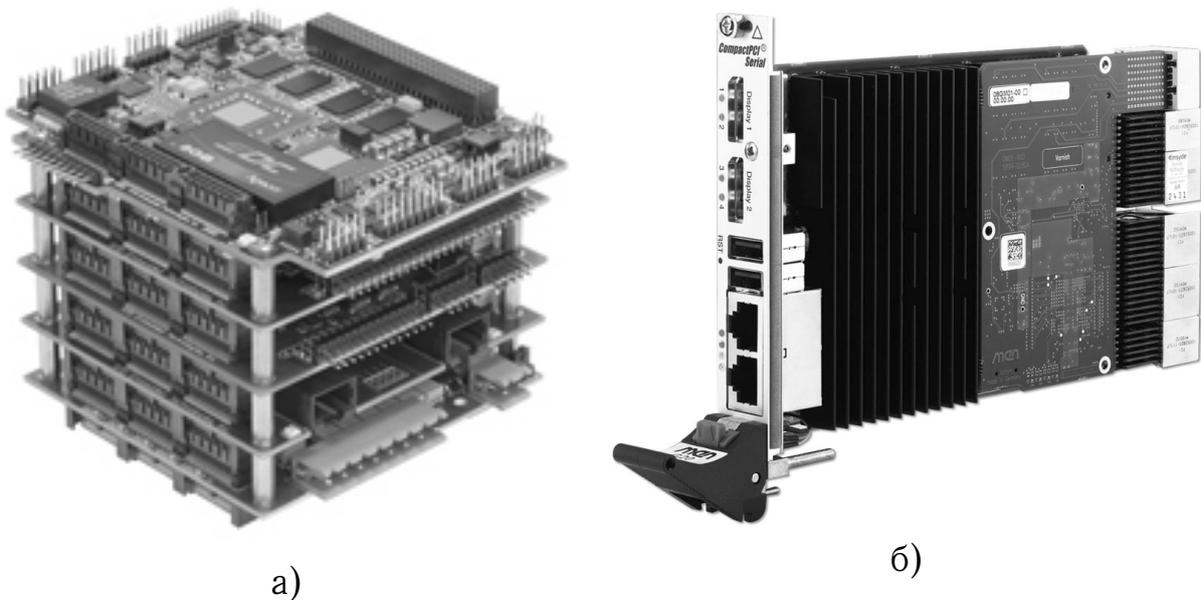


Рисунок 5.7: Примеры конструктивов для встраиваемых применений.
 а) мезонинная конструкция в формате PC-104, б) плата в формате Compact-PCI.

ры равномерно распределяют по всей поверхности печатной платы. Их необходимо размещать в точках подключения питания и земли для каждой микросхемы. Каждый конденсатор соединяет плюс питания с корпусом. Фильтрующие конденсаторы шунтируют (закорачивают) высокочастотные помехи, возникающие в цепях питания.

Шунтирование происходит из-за того, что активное сопротивление конденсатора тем меньше, чем выше частота сигнала (см. форм. 5.1).

$$X_C = \frac{1}{2\pi fC} \quad (5.1)$$

где, X_C — реактивное сопротивление конденсатора, f — частота, C — емкость.

Для постоянного напряжения сопротивление конденсатора близко к бесконечности, а для переменного напряжения высокой частоты конденсатор является резистором с низким сопротивлением.

Для защиты дискретных входов от импульсных помех часто используют катушки индуктивности или дроссели. Катушка индуктивности имеет почти нулевое сопротивление для постоянного тока и высокое сопротивление для переменного (см. форм. 5.2).

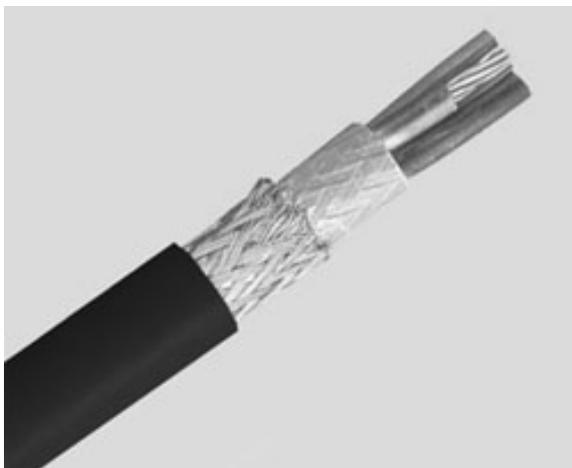
$$X_L = 2\pi fL \quad (5.2)$$



а)



б)



в)



г)

Рисунок 5.8: а) GSM модуль в защитном экране, б) устройство грозозащиты, в) экранированный кабель, г) ферритовый фильтр для кабеля

где, X_L — реактивное сопротивление индуктивности, f — частота, L — индуктивность.

Для защиты входов от высоких напряжений часто используют гальваническую изоляцию на базе оптронов. Гальваническая изоляция или гальваническая развязка — разделение электрических цепей посредством не проводящего ток материала.

Для примера рассмотрим оптрон TLP181, используемый в учебном стенде SDK-1.1 (см. рис. 5.10). Этот оптрон состоит из светодиода (выводы 1,3) и фототранзистора (выводы 6,4). Если через светодиод пустить ток, то он начинает излучать свет. Свет падает на PN переход фототранзистора и открывает его. Когда свет гаснет, фототранзистор закрывается. Гальваническая изоляция достигается за счет то-

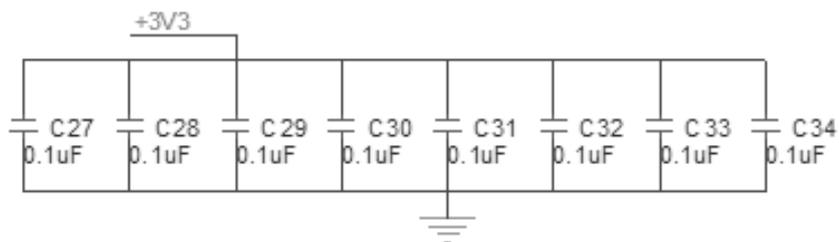
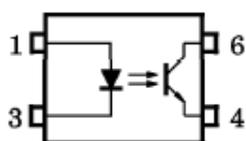


Рисунок 5.9: Фильтрующие конденсаторы на схеме электрической принципиальной.



- 1 : ANODE
- 3 : CATHODE
- 4 : EMITTER
- 6 : COLLECTOR

Рисунок 5.10: Оптрон TLP181.

го, что между двумя элементами оптрона нет никакой связи, кроме оптической. Получается, что напряжение на транзисторе (коллектор-эмиттер) физически не может превысить напряжения питания оптрона.

5.6 Срок службы компонентов

Срок службы электронных компонентов ограничен. Обычно он не превышает десяти лет. В большинстве случаев срок службы можно узнать из технической документации на конкретный электронный компонент (см. рис. 5.11 и 5.12).

Ограничение срока службы связано с необратимыми изменениями, происходящими в компонентах в течение времени. Меняется сопротивление резисторов, емкость конденсаторов и характеристики р-п переходов у полупроводников.

У электронных схем есть некоторый «запас прочности». Тем не менее, рано или поздно один из электронных компонентов может вывести всю схему из строя. Некоторые электронные компоненты, такие как электролитические конденсаторы, диоды, транзисторы или микросхемы, выходят из строя, сразу делая все устройство неработоспособным.

Срок службы компонентов зависит от многих факторов: температуры компонента, напряжения питания, частоты сигналов. Чем больше нагревается элемент, тем меньше срок его службы.

При проектировании ИУС необходимо учитывать срок службы компонентов и закладывать необходимость замены и модернизации блоков. С большой степенью вероятности по истечению срока службы компонентов производство многих компонентов будет прекращено и устройство придется переделывать на более современной элементной базе.

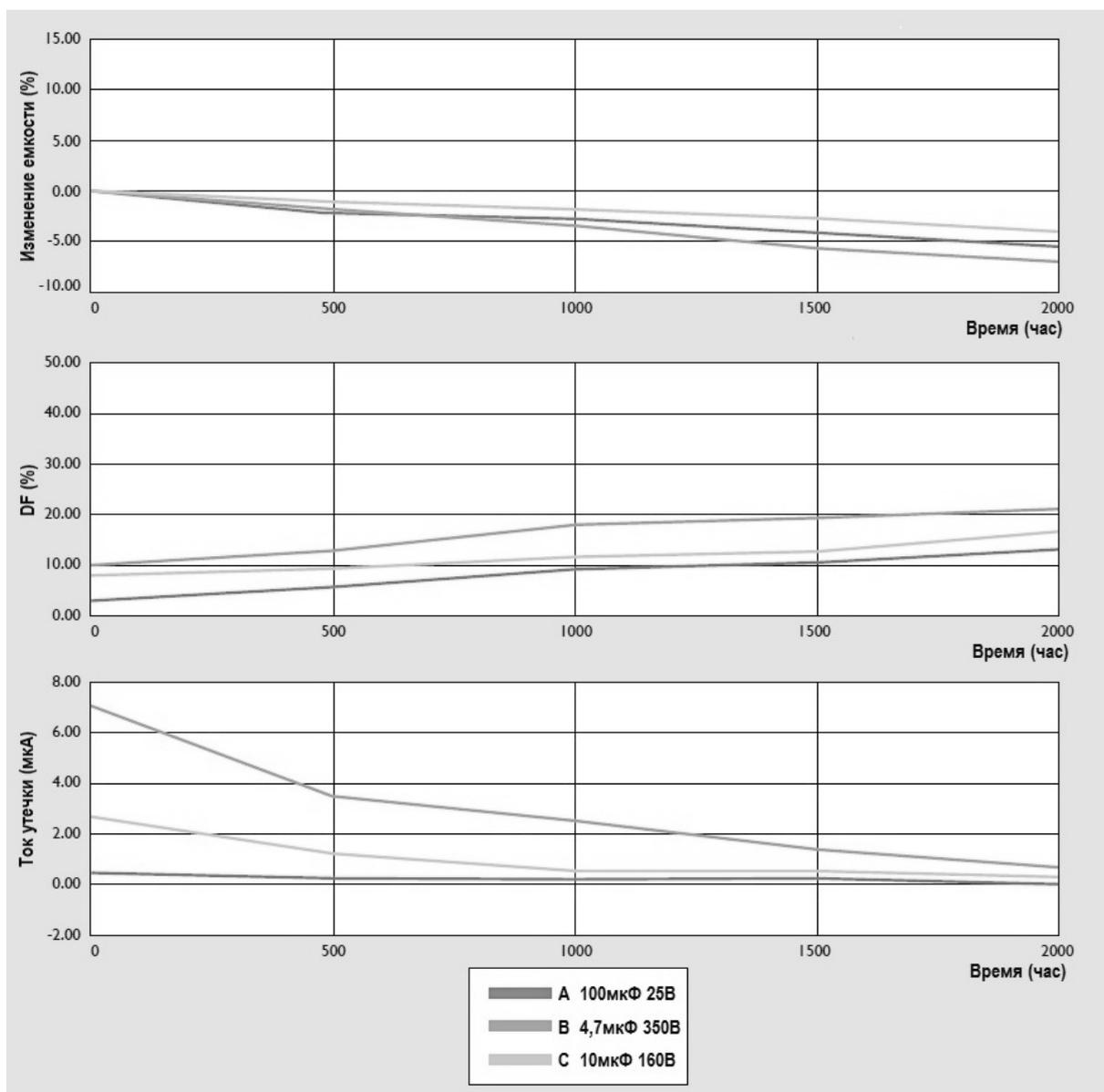


Рисунок 5.11: Срок службы электролитических конденсаторов [25].

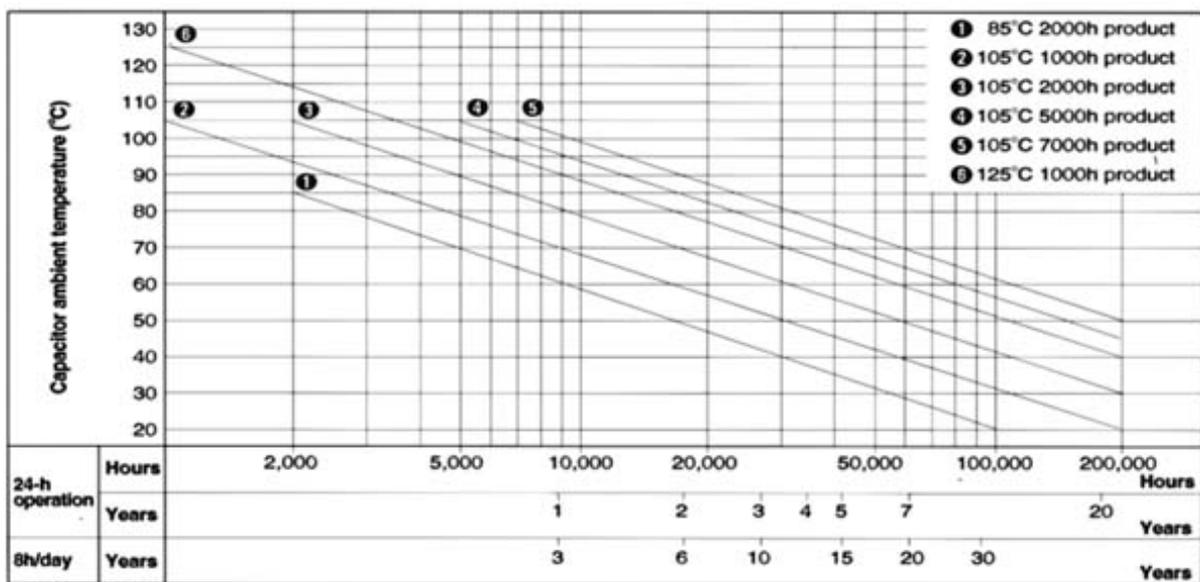


Рисунок 5.12: Зависимость срока службы от температуры алюминиевых электролитических конденсаторов [26].

Литература

1. Дубова Наталья. Очерки истории советской вычислительной техники. Очерк третий: Школа И.С. Брука. Малые и управляющие ЭВМ. // Открытые системы. 1999. Т. 3.
2. Программное обеспечение встроенных вычислительных систем / А.О. Ключев, П.В. Кустарев, Д.Р. Ковязина [и др.]. СПб ГУ ИТМО, 2009. С. 212.
3. Черняк Леонид. Киберфизические системы на старте. // Открытые системы. 2014. Т. 2.
4. Embedded System Market Analysis By Product (Hardware, Software), By Application (Automotive, Telecommunication, Healthcare, Industrial, Consumer Electronics, Military & Aerospace) And Segment Forecasts To 2020. 2014. — Режим доступа: <http://www.grandviewresearch.com/industry-analysis/embedded-system-market>, свободный.
5. ГОСТ Р 27.002-2009. Надежность в технике.
6. Федоров Кирилл. Что такое надежность и почему ею надо управлять? // Директор информационной службы. 2014. Т. 4.
7. Аппаратные и программные средства встраиваемых систем / А.О. Ключев, Д.Р. Ковязина, П.В. Кустарев [и др.]. СПб ГУ ИТМО, 2010. С. 286.
8. UM10107 P89LPC915/916/917 User manual. — Режим доступа: http://www.keil.com/dd/docs/datashts/philips/user_manual_p89lpc915_916_917.pdf, свободный.
9. ADuC812 MicroConverter, Multichannel 12-Bit ADC with Embedded FLASH MCU. — Режим доступа: http://www.analog.com/static/imported-files/data_sheets/ADUC812.pdf, свободный.

10. RM0090 Reference manual STM32F405xx/07xx, STM32F415xx/17xx, STM32F42xxx and STM32F43xxx advanced ARM®-based 32-bit MCUs. 2014. — Режим доступа: http://www.st.com/web/en/resource/technical/document/reference_manual/DM00031020.pdf, свободный.
11. System-on-Chip: Reuse and Integration / R. Saleh, S. Wilton, S. Mirabbasi et al. // Proceedings of the IEEE. 2006. Jun.. Vol. 94, no. 6. P. 1050–1069. — Режим доступа: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1652898>, свободный.
12. Sundaramoorthy Navanee. Put a configurable 32-bit processor in your FPGA. 2007. — Режим доступа: <http://www.embedded.com/design/mcus-processors-and-socs/4007080/Put-a-configurable-32-bit-processor-in-your-FPGA>, свободный.
13. EMBEDDED MARKET STUDY. 2013. — Режим доступа: <http://e.ubmelectronics.com/2013EmbeddedStudy/ubm-thanks.html>, свободный.
14. Sundaramoorthy Navanee. Put a configurable 32-bit processor in your FPGA. 2007. — Режим доступа: <http://www.embedded.com/design/mcus-processors-and-socs/4007080/Put-a-configurable-32-bit-processor-in-your-FPGA>, свободный.
15. Kahn Gilles. The semantics of a simple language for parallel programming // In Information Processing'74: Proceedings of the IFIP. 1974. — Режим доступа: http://www.tik.ee.ethz.ch/tik/education/lectures/hswcd/papers/2_KahnProcessNetworks.pdf, свободный.
16. MAX6369–MAX6374 Pin-Selectable Watchdog Timers. — Режим доступа: <http://datasheets.maximintegrated.com/en/ds/MAX6369-MAX6374.pdf>, свободный.
17. I2C-bus specification and user manual (Rev. 06, 04/2014). 2014. — Режим доступа: http://www.nxp.com/documents/user_manual/UM10204.pdf, свободный.
18. NXP. PCF8583 – Clock/calendar with 240 x 8-bit RAM. 2014. — Режим доступа: http://www.nxp.com/documents/data_sheet/PCF8583.pdf, свободный.

19. Two-wire Serial EEPROM AT24C02A/AT24C04A. 2005. — Режим доступа: <http://www.atmel.com/Images/doc0976.pdf>, свободный.
20. Motorola. Motorola MC68HC11 Reference Manual. Prentice H изд. 1989. С. 250.
21. Atmel. AT25010B, AT25020B, AT25040B SPI Serial EEPROM. 2014. — Режим доступа: <http://www.atmel.com/images/atmel-8707-seeeprom-at25010b-020b-040b-datasheet.pdf>, свободный.
22. DS1820 1-Wire™ Digital Thermometer. — Режим доступа: <http://www.systronix.com/Resource/ds1820.pdf>, свободный.
23. МЭК 60529(2001). Степени защиты, обеспечиваемые корпусами (Код IP).
24. Международная лаборатория «Архитектура и методы проектирования встраиваемых систем и систем на кристалле». — Режим доступа: <http://irc.ifmo.ru/ru/87825>, свободный.
25. Электролитические конденсаторы: особенности применения. — Режим доступа: <http://www.compel.ru/lib/articles/elektroliticheskie-kondensatoryi-osobennosti-primeneniya/>, свободный.
26. Таблица расчета срока службы алюминиевых электролитических конденсаторов. — Режим доступа: http://www.smd-component.ru/catalog/18_3.htm, свободный.

Миссия университета — генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки, технологий и других областей для содействия решению актуальных задач.

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Кафедра ВТ Университета ИТМО создана в 1937 году и является одной из старейших и авторитетнейших научно-педагогических школ России.

Традиционно основной упор в подготовке специалистов на кафедре делается на фундаментальную базовую подготовку в рамках общепрофессиональных и специальных дисциплин, охватывающих наиболее важные разделы вычислительной техники.

Кафедра является одной из крупнейших в университете. Учебными курсами и научно-исследовательскими работами руководят 8 профессоров и 16 доцентов. На кафедре обучаются более 500 студентов и 30 аспирантов.

Кафедра имеет собственные компьютерные классы и специализированные исследовательские лаборатории, оснащенные современной вычислительной и оргтехникой, уникальным инструментальным и технологическим оборудованием, измерительными приборами и программным обеспечением.

В 2007-2008 гг. коллективом кафедры была успешно реализована инновационная образовательная программа СПбНИУ ИТМО по научно-образовательному направлению «Встроенные вычислительные системы».

Начиная с 2009 года кафедра вычислительной техники является активным участником реализации программы развития национального исследовательского университета НИУ ИТМО, вошла в состав крупнейшего в НИУ ИТМО научно-исследовательского центра «Интеллектуальные системы управления и обработки информации».

Начиная с 2013 года кафедра ВТ принимает активное участие в реализации мероприятий программы повышения международной конкурентоспособности Университета ИТМО «5 в 100».

Аркадий Олегович Ключев
Павел Валерьевич Кустарев
Алексей Евгеньевич Платунов

АППАРАТНЫЕ СРЕДСТВА ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

Учебное пособие

В авторской редакции
Редакционно-издательский отдел Университета ИТМО
Зав. РИО
Подписано к печати
Заказ №
Тираж
Отпечатано на ризографе

Н. Ф. Гусарова

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверкский пр., 49