

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО «АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт цифровых технологий, электроники и физики (ИЦТЭФ)
Кафедра вычислительной техники и электроники (ВТиЭ)

Изучение и сравнение AVR и ARM микроконтроллеров
Отчет по научно-исследовательской работе

Выполнил: студент 595 гр.

_____ А. В. Лаптев

Проверил: ст. преп. кафедры
ВТиЭ

_____ И. А. Шмаков

«____» _____ 2022 г.

Барнаул, 2022 г.

РЕФЕРАТ

Полный объём работы составляет 42 страницы, включая 6 рисунков и 0 таблиц.

Цель работы — изучение и сравнение AVR и ARM микроконтроллеров, а также последующая разработка программного продукта на основе полученных в результате исследовательской работы знаний.

В результате выполнения научно-исследовательской работы было проведено знакомство с архитектурой микроконтроллеров AVR и ARM на примере представителя каждого семейства, знакомство с наборами команд для микроконтроллеров семейств AVR и ARM, а также изучены некоторые аппаратные и программные средства для разработки под AVR и ARM микроконтроллеры.

Ключевые слова: AVR, ARM, Arduino, микроконтроллер.

Отчёт оформлена с помощью системы компьютерной вёрстки $\text{T}_{\text{E}}\text{X}$ и его расширения $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$ из дистрибутива *TeX Live*.

СОДЕРЖАНИЕ

| | |
|--|----|
| Введение | 4 |
| 1 Обзор AVR и ARM | 5 |
| 1.1 AVR | 5 |
| 1.1.1 Коротко об AVR | 5 |
| 1.1.2 Архитектура | 5 |
| 1.1.3 Система команд микроконтроллеров AVR | 8 |
| 1.1.4 Семейства и версии микроконтроллеров | 8 |
| 1.1.5 Программная модель AVR-микроконтроллеров | 10 |
| 1.1.6 Периферия | 11 |
| 1.1.7 Питание | 15 |
| 1.2 ARM | 16 |
| 1.2.1 Коротко об ARM | 16 |
| 1.2.2 Архитектура | 17 |
| 1.2.3 Наборы команд микроконтроллеров ARM | 21 |
| 1.2.4 Семейства и версии микроконтроллеров | 23 |
| 1.2.5 Программная модель ARM-микроконтроллеров | 23 |
| 1.2.6 Периферия | 26 |
| 1.2.7 Питание | 29 |
| 2 Программное и аппаратное обеспечение для AVR и ARM микро- контроллеров | 32 |
| 2.1 Arduino | 32 |
| 2.1.1 Программная часть | 32 |
| 2.1.2 Язык программирования | 33 |
| 2.1.3 Аппаратная часть | 34 |
| 2.1.4 Классический конструктив | 34 |
| 2.1.5 Миниатюрный конструктив | 35 |
| 2.1.6 Микроконтроллер | 36 |
| 2.1.7 Периферия | 38 |
| 2.2 Другое популярное программное обеспечение для AVR и ARM микроконтроллеров | 39 |
| Заключение | 40 |
| Список использованной литературы | 41 |

ВВЕДЕНИЕ

На сегодняшний день практически у каждого есть какое-либо портативное устройство или другая потребительская электроника. В такой технике приоритетом является низкое энергопотребление в сочетании с относительным быстродействием. Этим требованиям отвечают семейства микроконтроллеров AVR и ARM.

Сейчас встраиваемые системы и портативные устройства — одна из самых быстрорастущих технологических сфер. И, поэтому, разработка под микроконтроллеры и микропроцессоры этих семейств является актуальной.

В данной научно-исследовательской работе объектом исследования являются микроконтроллеры семейств AVR и ARM.

Целью работы является изучение и сравнение AVR и ARM микроконтроллеров, а также последующая разработка программного продукта на основе полученных в результате исследовательской работы знаний.

Задачи научно-исследовательской работы:

1. Знакомство с архитектурой микроконтроллеров AVR и ARM на примере представителя каждого семейства.
2. Знакомство с наборами команд для микроконтроллеров каждого семейства.
3. Изучение имеющихся аппаратных и программных средств для разработки под AVR и ARM микроконтроллеры.
4. Применение полученных знаний для разработки собственного программного продукта.
5. Разработка собственного программного продукта на выбранной аппаратно-программной платформе.

1. ОБЗОР AVR И ARM

1.1. AVR

1.1.1. Коротко об AVR

AVR — семейство восьмибитных микроконтроллеров фирмы Atmel, позднее Microchip, впервые выпущенные в 1996 г. Они представляют собой мощный инструмент, универсальную основу для создания современных экономичных встраиваемых систем многоцелевого назначения. [1]

Микроконтроллеры AVR имеют гарвардскую архитектуру (программа и данные находятся в разных адресных пространствах) и систему команд, близкую к идеологии RISC. Процессор AVR имеет 32 8-битных регистра общего назначения, объединенных в регистровый файл. В отличие от «идеального» RISC, регистры не абсолютно равноправны: [2]

- три «сдвоенных» 16-битных регистра-указателя X(r26:r27), Y(r28:r29), Z(r30:r31); команды, увеличивающие и уменьшающие 16-битное значение (в тех моделях, где они доступны) с непосредственным операндом, работают только с одной из этих пар;
- некоторые команды работают только с регистрами r16...r31; к ним относятся команды, работающие с непосредственным операндом;
- команда копирования пары регистров (в тех моделях, где доступна) работает только с соседними регистрами, начинающимися с нечётного (r1:r0, r3:r2, ..., r31:r30);
- некоторые варианты команд умножения принимают в качестве аргументов только регистры из диапазона r16...r23;
- результат умножения (в моделях, где есть такой модуль) всегда помещается в r0:r1; также только эта пара используется в качестве операндов для команды самопрограммирования, при котором основная программа может изменить часть своего кода (где доступна).

Сигналы JTAG (TMS, TDI, TDO и TCK) мультиплексированы на порт ввода-вывода. Режим работы — JTAG или порт — задаётся соответствующим битом в регистре fuses. МК AVR поставляются с включённым интерфейсом JTAG.

1.1.2. Архитектура

AVR-микроконтроллеры содержат на кристалле следующие аппаратные средства: 8-разрядное процессорное ядро, память программ, оператив-

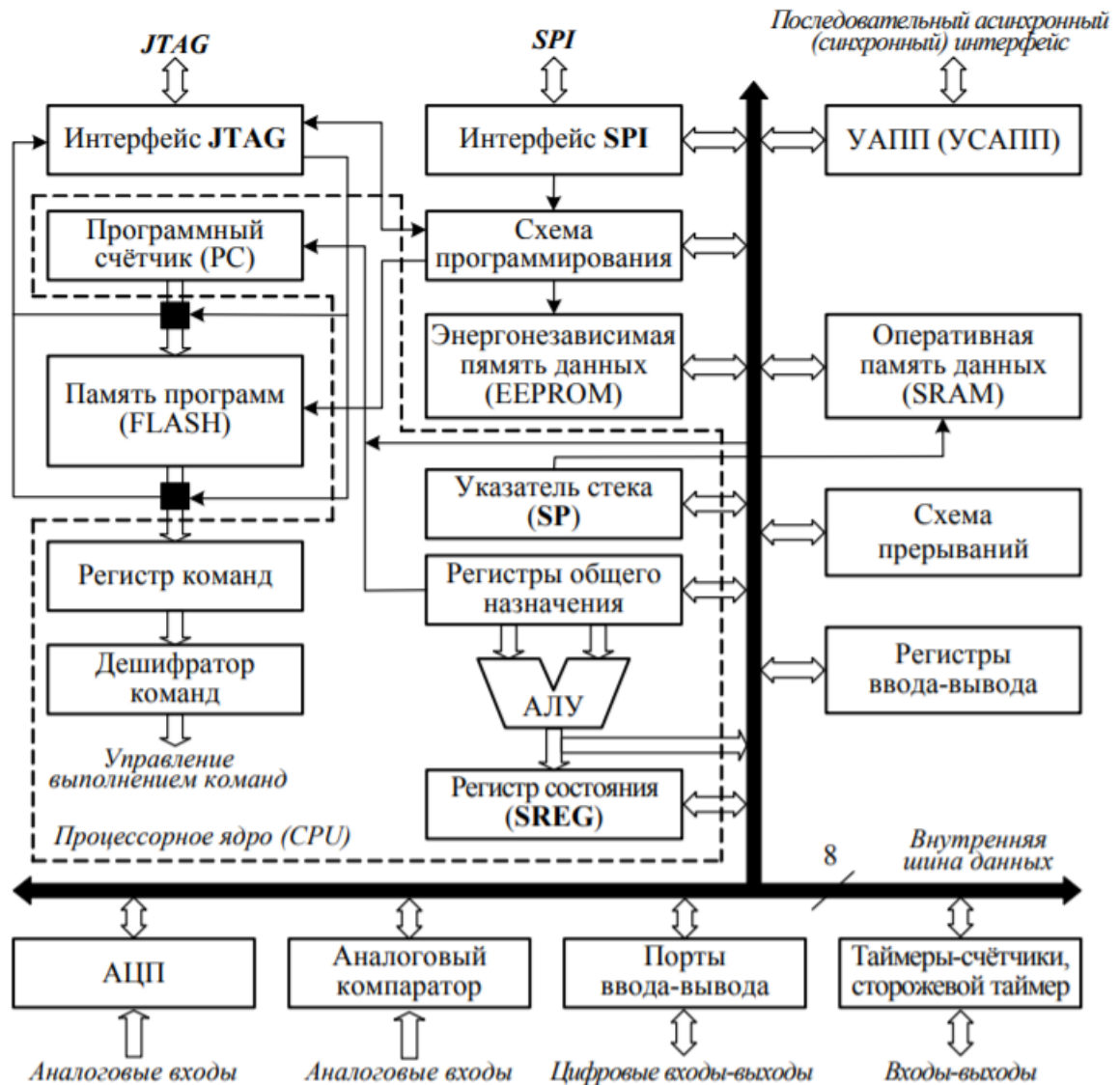


Рис. 1.1 Архитектура микроконтроллеров семейства AVR

Процессорное ядро (CPU) AVR-микроконтроллеров содержит арифметико-логическое устройство (АЛУ), регистры общего назначения (РОН), программный счетчик, указатель стека, регистр состояния, регистр команд, дешифратор команд, схему управления выполнением команд.

Программный счетчик (РС) содержит адрес следующей выполняемой команды.

Указатель стека (SP) служит для хранения адреса вершины стека.

Регистр состояния (SREG) содержит слово состояния процессора.

АЛУ — арифметико-логическое устройство, которое синхронно с тактовым сигналом и опираясь на состояние счетчика команд (РС) выбирает из

памяти программ (FLASH) очередную команду и производит ее выполнение. [4]

В АЛУ выполняются все вычислительные операции. Операции производятся только над содержимым РОН. На выборку содержимого регистров, выполнение операции и запись результата обратно в РОН затрачивается один машинный такт.

Регистры общего назначения представляют собой 8-разрядные ячейки памяти с быстрым доступом, непосредственно доступные АЛУ. В AVR-микроконтроллерах имеется 32 РОН. Адресное пространство регистров общего назначения размещено в начале оперативной памяти (RAM) но не является ее частью.

FLASH — память программ, энергонезависимое ПЗУ, выполнено по технологии FLASH. Выполнена в виде последовательности 16-разрядных ячеек, так как большинство команд AVR-микроконтроллера являются 16-разрядными словами. Здесь хранится программа, которая будет исполняться блоком АЛУ микроконтроллера. Флеш-память чипа можно многократно перезаписывать. Данный тип памяти может сохранять записанные в нее данные в течение 40 лет, а количество возможных циклов стирания/записи может достигать 10 000. В зависимости от модели микроконтроллера размер FLASH-памяти может достигать 256 КБ.

EEPROM — энергонезависимая память данных. В данной памяти можно хранить настройки выполнения программы, собранные данные для статистики работы устройства и другую полезную информацию. Организована как последовательность 8-разрядных ячеек.

Для EEPROM выделено отдельное адресное пространство которое отличается от адресного пространства RAM и FLASH. Память EEPROM микроконтроллера — очень ценный ресурс, поскольку ее как правило очень мало — от 0,5 до 4 Кб на чип. Количество перезаписей для данного типа памяти составляет порядка 100 000.

Регистр команд, дешифратор команд и схема управления выполнением команд обеспечивают выборку из памяти программ команды, адрес которой содержится в программном счетчике, ее декодирование, определение способа доступа к указанным в команде аргументам и, собственно, выполнение команды. Для ускорения выполнения команд используется механизм конвейеризации, который заключается в том, что во время исполнения текущей команды программный код следующей выбирается из памяти и декодируется.

Оперативная память данных представляет собой статическое ОЗУ (SRAM) и организована как последовательность 8-разрядных ячеек. Оперативная память данных может быть внутренней (до 32 Кбайт) и внешней (до 64 Кбайт).

Регистры ввода-вывода предназначены для управления процессорным ядром и периферийными устройствами AVR-микроконтроллера.

1.1.3. Система команд микроконтроллеров AVR

Система команд микроконтроллеров AVR весьма развита и насчитывает в различных моделях от 90 до 135 различных инструкций.

Большинство команд занимает только 1 ячейку памяти (16 бит) и выполняется за 1 такт.

Все множество команд микроконтроллеров AVR можно разбить на несколько групп: [2]

- команды логических операций;
- команды арифметических операций и команды сдвига;
- команды операции с битами;
- команды пересылки данных;
- команды передачи управления;
- команды управления системой.

Управление периферийными устройствами осуществляется через адресное пространство данных. Для удобства существуют «сокращенные команды» IN/OUT. [2]

1.1.4. Семейства и версии микроконтроллеров

Стандартные семейства микроконтроллеров:

- tinyAVR (ATtinyxxx):
 - Флеш-память до 16 Кб;
 - SRAM до 512 б;
 - EEPROM до 512 б;
 - Число линий ввода-вывода 4-18 (общее количество выводов 6-32);
 - Ограниченный набор периферийных устройств.
- megaAVR (ATmegaxxx):
 - Флеш-память до 256 Кб;
 - SRAM до 8 Кб;
 - EEPROM до 4 Кб;

- Число линий ввода-вывода 23-86 (общее количество выводов 28-100);
- Аппаратный умножитель;
- Расширенная система команд и периферийных устройств.
- XMEGA AVR (ATxmegaxxx):
 - Флеш-память до 384 Кб;
 - SRAM до 32 Кб;
 - EEPROM до 4 Кб;
 - Четырехканальный DMA-контроллер;
 - Инновационная система обработки событий.

Также выделяют семейство микроконтроллеров AVR UC3 — микроконтроллеры, предназначенные для использования в задачах с высокими требованиями к производительности [5].

На основе стандартных семейств выпускаются микроконтроллеры, адаптированные под конкретные задачи:

- со встроенными интерфейсами USB, CAN, контроллером LCD;
- со встроенным радиоприемопередатчиком;
- для управления электродвигателями;
- для автомобильной электроники;
- для осветительной техники.

Кроме указанных выше семейств, ATMEL выпускает 32-разрядные микроконтроллеры семейства AVR32.

Версии контроллеров: [1]

- AT(mega/tiny)xxx — базовая версия.
- ATxxxL — версии контроллеров, работающих на пониженном (Low) напряжении питания (2,7 В).
- ATxxxV — версии контроллеров, работающих на низком напряжении питания (1,8 В).
- ATxxxP — малопотребляющие версии (до 100 нА в режиме Power-down), применена технология *riCoPower*, по выводу и функционально совместимы с предыдущими версиями.
- ATxxxA — уменьшен ток потребления, перекрывается весь диапазон тактовых частот и напряжений питания двух предыдущих версий (также, в некоторых моделях, добавлены новые возможности и новые регистры, но сохранена полная совместимость с предыдущими версиями).

1.1.5. Программная модель AVR-микроконтроллеров

Программная модель микропроцессора представляет собой совокупность программно доступных ресурсов. В программную модель микроконтроллеров семейства AVR входят РОН, регистры ввода-вывода, память программ, оперативная память данных и энергонезависимая память данных (рис. 1.2). [4]

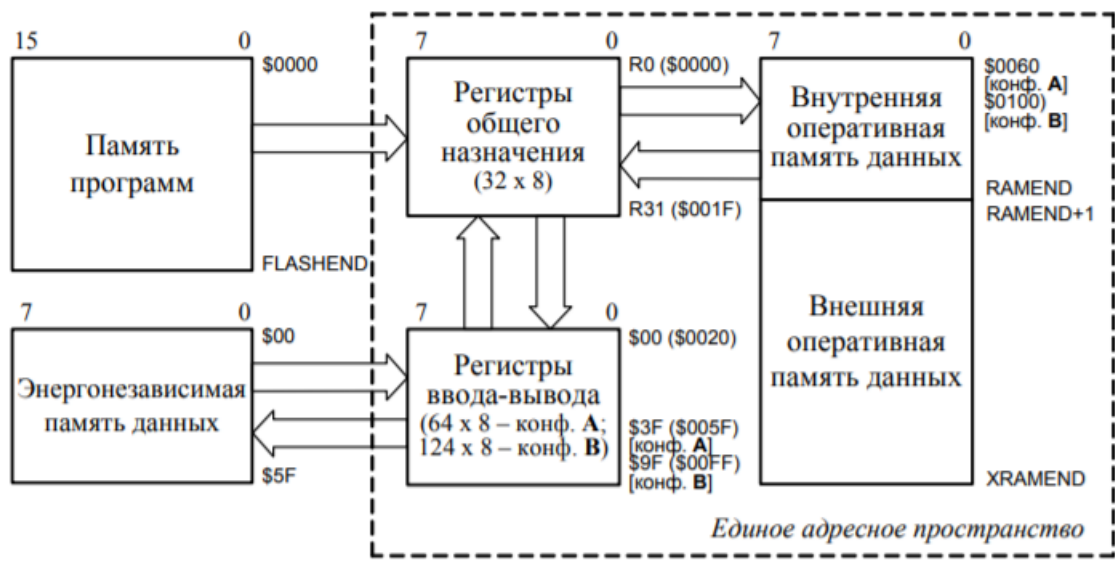


Рис. 1.2 Программная модель AVR-микроконтроллеров

РОН (R0...R31) могут использоваться в программе для хранения данных, адресов, констант и другой информации. Шесть старших регистров, как было сказано выше, объединены попарно и составляют три 16-разрядных регистра X [R27:R26], Y [R29:R28] и Z [R31:R30] (рис. 1.3).

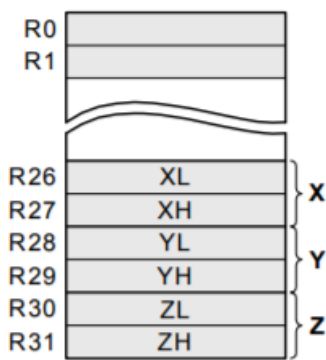


Рис. 1.3 Регистры общего назначения

РОН, регистры ввода-вывода и оперативная память данных образуют единое адресное пространство.

Существует две конфигурации единого адресного пространства памяти AVR-микроконтроллеров. В конфигурации А младшие 32 адреса

(\$0000...\$001F) соответствуют РОН, следующие 64 адреса (\$0020...\$005F) занимают регистры ввода-вывода, внутренняя оперативная память данных начинается с адреса \$0060. В конфигурации В, начиная с адреса \$0060, размещаются 160 дополнительных регистров ввода-вывода; внутренняя оперативная память данных начинается с адреса \$0100. Конфигурация А используется в младших моделях микроконтроллеров и в некоторых старших моделях в режиме совместимости с моделями, снятыми с производства; конфигурация В — в старших моделях.

В память программ, кроме собственно программы, могут быть записаны постоянные данные, которые не изменяются в процессе работы микроконтроллера (константы, таблицы линеаризации датчиков и т. п.). Выполнение программы при включении питания или после сброса микроконтроллера начинается с команды, находящейся по адресу 0000 памяти программ.

Энергонезависимая память данных предназначена для хранения информации, которая может изменяться непосредственно в процессе работы микропроцессорной системы. Энергонезависимая память данных имеет отдельное адресное пространство и может быть считана и записана программным путем.

Система команд микроконтроллера представляет собой совокупность выполняемых микроконтроллером операций и правил их кодирования в программе. Система команд AVR-микроконтроллеров включает команды арифметических и логических операций, команды ветвления, управляющие последовательностью выполнения программы, команды передачи данных и команды операций с битами. Всего в систему команд входит более 130 инструкций. Младшие модели микроконтроллеров не поддерживают некоторые из них.

1.1.6. Периферия

Микроконтроллеры AVR имеют хорошо развитую периферию.

К периферийным устройствам AVR-микроконтроллера относятся многофункциональные, двунаправленные порты ввода-вывода, таймеры, счетчики, сторожевой таймер, аналоговый компаратор, 10-разрядный 8-канальный АЦП (12-разрядный для XMEGA AVR), универсальный асинхронный (синхронно-асинхронный) приемопередатчик — УАПП (УСАПП), последовательный периферийный интерфейс (SPI), интерфейс JTAG, устройство сброса по понижению питания, широтно-импульсные модуляторы, датчики температуры и др. [2] [1]

Порты ввода-вывода AVR имеют число независимых двунаправленных GPIO линий «вход-выход» от 3 до 86, к которым можно подключить разнообразные датчики, исполняющие устройства и цепи. Каждая линия порта может быть запрограммирована на вход или на выход.

Архитектурная особенность построения портов ввода-вывода у AVR заключается в том, что для каждого физического вывода существует 3 бита контроля/управления, а не 2, как у распространенных 8-разрядных микроконтроллеров (Intel, Microchip, Motorola и т.д.). Это позволяет избежать необходимости иметь копию содержимого порта в памяти для безопасности и повышает скорость работы микроконтроллера при работе с внешними устройствами, особенно в условиях внешних электрических помех.

Система прерываний — одна из важнейших частей микроконтроллера. Все микроконтроллеры AVR имеют многоуровневую систему прерываний. При возникновении события, вызывающего прерывание, микроконтроллер сохраняет содержимое счетчика команд, прерывает выполнение центральным процессором текущей программы и переходит к выполнению подпрограммы обработки прерывания. После выполнения подпрограммы прерывания осуществляется восстановление предварительно сохраненного счетчика команд и процессор возвращается к выполнению прерванной программы.

Микроконтроллеры AVR имеют в своем составе от 1 до 4 таймеров/счетчиков с разрядностью 8 или 16 бит, которые могут работать и как таймеры от внутреннего источника тактовой частоты, и как счетчики внешних событий [6].

Их можно использовать для точного формирования временных интервалов, подсчета импульсов на выводах микроконтроллера, формирования последовательности импульсов, тактирования приемопередатчика последовательного канала связи. В режиме ШИМ (PWM) таймер/счетчик может представлять собой широтно-импульсный модулятор и используется для генерирования сигнала с программируемыми частотой и скважностью. Таймеры/счетчики способны вырабатывать запросы прерываний, переключая процессор на их обслуживание по событиям и освобождая его от необходимости периодического опроса состояния таймеров.

Сторожевой таймер предназначен для предотвращения катастрофических последствий от случайных сбоев программы.

Идея использования сторожевого таймера состоит в регулярном его сбрасывании под управлением программы или внешнего воздействия до того, как закончится его выдержка времени и не произойдет сброс процессора.

Аналоговый компаратор сравнивает напряжения на двух выводах микроконтроллера. Результатом сравнения будет логическое значение, которое может быть прочитано из программы. Выход аналогового компаратора можно включить на прерывание от аналогового компаратора. Пользователь может установить срабатывание прерывания по нарастающему или спадающему фронту или по переключению. Присутствует у всех современных AVR, кроме Mega8515.

Аналого-цифровой преобразователь необходим для получения числового значения напряжения, поданного на его вход. Этот результат сохраняется в регистре данных АЦП. Какой из выводов микроконтроллера будет являться входом АЦП, определяется числом, занесенным в соответствующий регистр. Программируемый коэффициент усиления перед АЦП 1, 10 и 200; в качестве опорного напряжения могут выступать: напряжение питания, внешнее напряжение, или внутреннее некалиброванное опорное напряжение около 2,56 В (для моделей, имеющих минимальное напряжение питания от 2,7 В и выше) либо 1,1 В (с минимальным напряжением питания 1,8 В).

Универсальный асинхронный или универсальный синхронно/асинхронный приемопередатчик (УАПП/УСАПП) — удобный и простой последовательный интерфейс для организации информационного канала обмена микроконтроллера с внешним миром. Способен работать в дуплексном режиме (одновременная передача и прием данных).

Последовательный периферийный трехпроводный интерфейс (SPI) предназначен для организации обмена данными между двумя устройствами. С его помощью может осуществляться обмен данными между микроконтроллером и различными устройствами, такими, как цифровые потенциометры, ЦАП/АЦП, FLASH ПЗУ и др. С помощью этого интерфейса удобно производить обмен данными между несколькими микроконтроллерами AVR. Кроме того, через интерфейс SPI может осуществляться программирование микроконтроллера.

SPI, зачастую, применяется для обмена данными между несколькими микроконтроллерами со скоростью до нескольких МГц. [4]

Для обмена данными по SPI интерфейсу между двумя устройствами достаточно 3-х проводников:

- MOSI (Master Output Slave Input) — данные от ведущего к ведомому;
- MISO (Master Input Slave Output) — данные от ведомого к ведущему;
- CLK (Clock) — тактовый сигнал.

Устройства с SPI-интерфейсом делятся на два типа: ведущий (Master) и ведомый (Slave). Если к интерфейсу подключено несколько устройств то для обмена данными между ними нужны дополнительные линии связи(проводники) чтобы мастеру можно было выбрать ведомое устройство и сделать запрос к нему.

Также SPI интерфейс используется для внутрисхемного SPI программирования, по этому интерфейсу к микроконтроллеру подключается программатор.

Двухпроводной последовательный интерфейс (TWI) является полным аналогом базовой версии интерфейса I2C. Этот интерфейс позволяет объединить вместе до 128 различных устройств с помощью двунаправленной шины, состоящей из линии тактового сигнала (SCL) и линии данных (SDA).

В отличие от SPI интерфейса (один мастер и один/несколько ведомых) интерфейс TWI — двунаправленный, что позволяет организовать между несколькими микроконтроллерами небольшую внутреннюю сеть.

Интерфейс JTAG был разработан группой ведущих специалистов по проблемам тестирования электронных компонентов. В состав интерфейса входят четыре проводника: TDI(Test Data In), TDO(Test Data Out), TCK(Test Clock), TMS (Test Mode Select). Интерфейс JTAG используется для тестирования печатных плат; внутрисхемной отладки, используя блок внутренней отладки (OCD), без использования эмуляторов; программирования микроконтроллеров. Многие микроконтроллеры семейства Mega имеют совместимый интерфейс JTAG или debugWIRE для встроенной отладки. Кроме того, все микроконтроллеры Mega с флэш-памятью емкостью 16 Кбайт и более могут программироваться через интерфейс JTAG.

LCD — интерфейс для подключения жидкокристаллического индикатора или дисплея.

CAN — интерфейс для объединения в сеть различных контроллеров и датчиков.

USB — интерфейс для подключения периферийных устройств.

Тактовый генератор вырабатывает импульсы для синхронизации работы всех узлов микроконтроллера.

Внутренний тактовый генератор AVR может запускаться от нескольких источников опорной частоты:

- керамический или кварцевый резонатор (не у всех моделей);
- внешний тактовый сигнал;

- внутренний RC-генератор (частота 1, 2, 4, 8 МГц, а также, для некоторых моделей ATtiny — 4,8, 6.4, 9.6 МГц и 128 кГц).

Минимальная допустимая частота ничем не ограничена (вплоть до пошагового режима). Максимальная рабочая частота определяется конкретным типом микроконтроллера и указывается Atmel в его характеристиках.

Установка источника тактовых импульсов производится с помощью FUSE-битов.

FUSES — специальные 4 байта данных, которые настраивают некоторые глобальные параметры микроконтроллера в процессе прошивки. После прошивки данные биты нельзя изменить через внутреннюю программу что записана в микроконтроллер. [4]

Данной конфигурацией бит указывается микроконтроллеру:

- какой использовать задающий генератор (внешний или внутренний);
- делить частоту генератора на коэффициент или нет;
- использовать ножку сброса (RESET) для сброса или же как дополнительный пин ввода-вывода;
- количество памяти для загрузчика;
- другие настройки зависимо от используемого микроконтроллера.

Система реального времени (RTC) реализована во всех микроконтроллерах серии Mega. Таймер/счетчик RTC имеет отдельный предделитель, который может быть программным способом подключен или к источнику основной тактовой частоты, или к дополнительному асинхронному источнику опорной частоты (кварцевый резонатор или внешний синхросигнал). Для этой цели зарезервированы два вывода микросхемы.

Набором периферийных устройств определяются функциональные возможности микроконтроллера.

Обмен информацией между устройствами AVR-микроконтроллера осуществляется посредством внутренней 8-разрядной шины данных.

1.1.7. Питание

AVR функционируют при напряжениях питания от 1,8 до 6,0 В.

AVR могут быть переведены программным путем в один из трех режимов пониженного энергопотребления:

- Режим холостого хода — прекращает работу только процессор и фиксируется содержимое памяти данных, а внутренний генератор синхросигналов, таймеры, система прерываний и сторожевой таймер продолжают функционировать.

- Стоповый режим — сохраняется содержимое регистрового файла, но останавливается внутренний генератор синхросигналов, и, следовательно, останавливаются все функции, пока не поступит сигнал внешнего прерывания или аппаратного сброса.
- Экономичный режим — продолжает работать только генератор таймера, что обеспечивает сохранность временной базы. Все остальные функции отключены.

В микроконтроллерах существует специальный режим — сброс при снижении напряжения питания (BOD). Схема BOD отслеживает напряжение источника питания. Если схема включена, то при снижении питания ниже некоторого значения она переводит микроконтроллер в состояние сброса. Когда напряжение питания вновь увеличится до порогового значения, запускается таймер задержки сброса. После формирования задержки внутренний сигнал сброса снимается и происходит запуск микроконтроллера. [1]

1.2. ARM

1.2.1. Коротко об ARM

Архитектура ARM — система команд и семейство описаний и готовых топологий 32-битных и 64-битных микропроцессорных/микроконтроллерных ядер, разрабатываемых компанией ARM Limited.

Среди лицензиатов готовых топологий ядер ARM — компании AMD, Apple, Analog Devices, Atmel, Xilinx, Cirrus Logic, Intel (до 27 июня 2006 года), Marvell, NXP, STMicroelectronics, Samsung, LG, MediaTek, Qualcomm, Sony, Texas Instruments, Nvidia, Freescale, Миландр, ЭЛВИС, HiSilicon, Байкал электроникс.

Значимые семейства: ARM7, ARM9, ARM11 и Cortex.

Многие лицензиаты проектируют собственные топологии ядер на базе системы команд ARM.

Процессоры и микроконтроллеры ARM широко используются в потребительской электронике — в том числе смартфонах, мобильных телефонах и плеерах, портативных игровых консолях, калькуляторах, умных часах и компьютерных периферийных устройствах, таких, как жесткие диски или маршрутизаторы.

Они имеют низкое энергопотребление, поэтому находят широкое применение во встраиваемых системах и преобладают на рынке мобильных устройств, для которых данный фактор немаловажен. [7]

1.2.2. Архитектура

Существует справочное руководство по архитектуре ARM, которое разграничивает все типы интерфейсов, которые поддерживает ARM, так как детали реализации каждого типа процессора могут различаться. Начиная с ARMv7, были определены 3 профиля:

- А (application) — для устройств, требующих высокой производительности (например, смартфоны, планшеты);
- R (real time) — для приложений, работающих в реальном времени;
- М (microcontroller) — для микроконтроллеров, недорогих встраиваемых устройств.

Профили могут поддерживать меньшее количество команд (команды определенного типа).

Микроконтроллеры ARM содержат следующие аппаратные средства: 32-разрядное процессорное ядро, память программ, оперативную память данных, регистры ввода-вывода, а также периферийные устройства (внутренние и внешние) (рис. 1.4) [8].

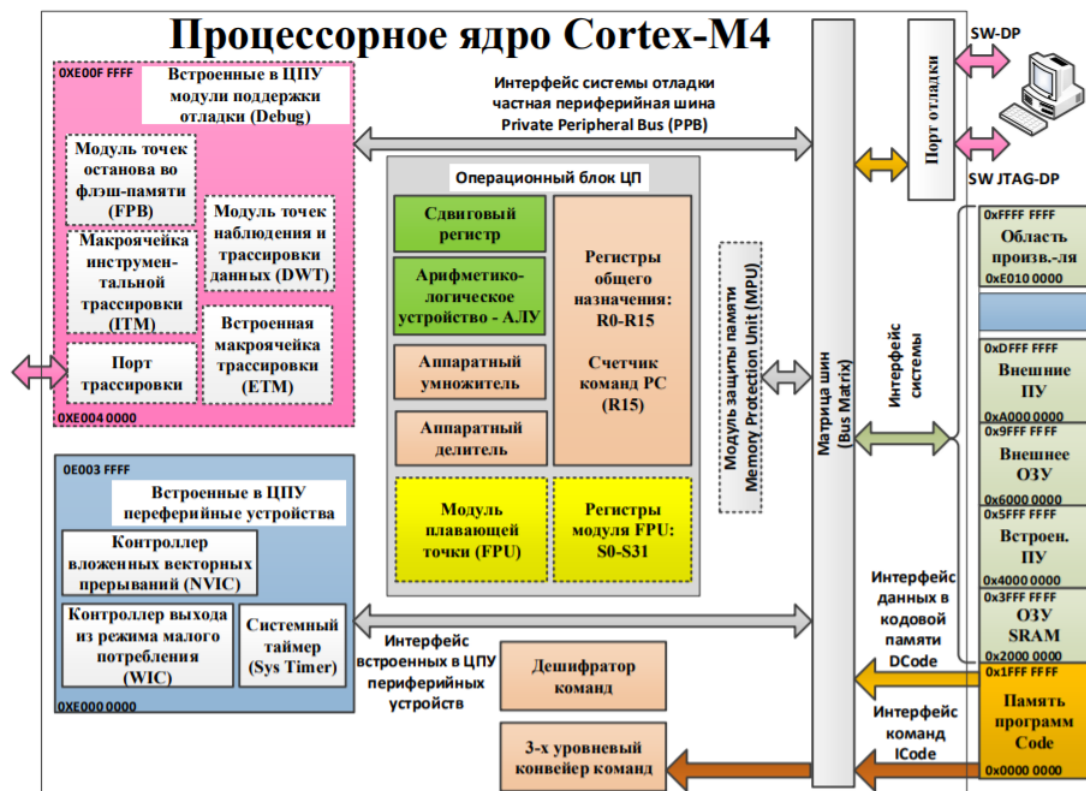


Рис. 1.4 Архитектура ARM-микроконтроллера Cortex M4

Процессорное ядро ARM-микроконтроллеров содержит в себе операционный блок (состоит из арифметико-логического устройства, аппаратный

умножитель, аппаратный делитель, файл регистров общего назначения, сопроцессор для обработки чисел в формате с плавающей точкой (присутствует не во всех моделях), файл регистров сопроцессора (также есть не во всех моделях)), встроенные модули поддержки отладки, встроенные периферийные устройства, дешифратор команд, конвейер команд, модуль защиты памяти.

Арифметико-логическое устройство (АЛУ) — двухпортовое устройство, которое является ядром операционного блока. Используется для выполнения арифметических, логических и других операций над 32-разрядными операндами. Все операции могут выполняться только над содержимым РОН. Загрузка результата операции также осуществляется в один из РОН.

Имеет дополнительный кольцевой сдвиговый регистр, позволяющий выполнять так называемые «попутные» операции сдвига при выполнении большинства обычных операций.

Аппаратный умножитель — устройство, которое служит для перемножения 32-разрядных чисел в формате с фиксированной точкой с поддержкой операций умножения-накопления, а также с возможностью получения 64-битного результата.

Аппаратный делитель — устройство, которое служит для деления 32-разрядных чисел со знаком или без знака.

Регистры общего назначения представляют собой 32-разрядные ячейки памяти с быстрым доступом, непосредственно доступные АЛУ. В ARM-микроконтроллерах имеется 16 РОН.

Модуль аппаратной поддержки вычислений в формате с плавающей точкой (FPU) выполняет функцию сопроцессора и обеспечивает:

- Обработку операндов, расположенных в 32-разрядных регистрах модуля плавающей точки, которые представляют собой дополнительное сверхоперативное ОЗУ модуля FPU (операнды для обработки могут располагаться только в этом регистровом файле);
- Операции обработки 32-разрядных чисел с плавающей точкой однократной точности: преобразования форматов, сложения, вычитания, сравнения, умножения, умножения с накоплением, деления, извлечения квадратного корня;
- Операции обычного последовательного умножения с накоплением и одновременного умножения с накоплением повышенной точности;
- Расширение диапазона чисел с плавающей точкой однократной точности за счет поддержки операций с денормализованными числами и всех режимов округления.

- Извлечение и выполнение команд обработки чисел с плавающей точкой на трехстадийном конвейере с независимой работой стадий конвейера.
- Выполнение большинства команд сопроцессора за один такт.

В микроконтроллерах ARM используются модули отладки, которые реализованы в архитектуре процессорного ядра.

С их помощью поддерживаются:

1. Доступ в режиме отладки ко всей памяти системы и регистрам процессора, включая доступ к регистрам периферийных устройств;
2. Установка в отлаживаемой программе точек останова, при достижении которых выполнение программы приостанавливается и управление передается отладчику;
3. Установка точек наблюдения, при достижении которых значения переменных сохраняются для последующего анализа;
4. Трассировка программы, когда она выполняется либо в пошаговом режиме, либо от одной точки останова до другой с возможностью контроля значений переменных;
5. Профилирование программы. Оценка быстродействия (времени выполнения) всей программы или отдельных ее фрагментов (подпрограмм), частоты обращения к подпрограммам с целью последующей оптимизации наиболее часто используемых фрагментов кода или определения вообще не используемых участков кода;
6. Вывод отладочной информации на дисплей системы верхнего уровня (компьютер), используемый для отладки;
7. Сопряжение с анализаторами данных в порту трассировки (TPA);
8. Доступ к встроенным в процессор отладочным устройствам через два внешних интерфейса, к которым подключаются компьютерные системы отладки верхнего уровня:

- Последовательный отладочный JTAG-порт (SWJDP), являющийся стандартом для производителей отладочного оборудования и микроконтроллеров;
- Последовательный отладочный порт (SW-DP), широко используемый производителями ARM-процессоров и микроконтроллеров;

В конкретном микроконтроллере может быть либо один из этих интерфейсов, либо оба для сопряжения со средствами отладки разных производителей.

Часто JTAG-порт является универсальным и совмещается с SW-DP-портом.

Функцию дешифрации команды, выполняет дешифратор команд. Дешифратор команд по коду операции запускает соответствующий данной команде дискретный управляющий автомат, задающий последовательность микроопераций, приводящих к выполнению команды. Перед декодированием команды происходит ее загрузка из памяти на 3-х уровневый конвейер команд и команда находится на конвейере до тех пор, пока не поступит на выполнение в операционный блок процессора.

Опциональный модуль защиты памяти (MPU), предназначен для защиты определенных областей памяти от случайного или несанкционированного доступа. Используется в основном в достаточно сложных системах, как правило, имеющих операционную систему реального времени. Области памяти, которые использует программное обеспечение верхнего уровня, могут быть заблокированы для доступа со стороны обычных пользовательских программ, что повышает надежность работы всей системы.

Память программ — ПЗУ, выполнено по технологии FLASH. Выполнено в виде последовательности 32-разрядных ячеек. Здесь хранится программа, которая будет исполняться блоком АЛУ микроконтроллера. Флеш-память чипа можно многократно перезаписывать.

Оперативная память данных представляет собой статическое ОЗУ (SRAM) и организована как последовательность 32-разрядных ячеек. Оперативная память данных может быть внутренней и внешней.

Среди внутренних периферийных устройств интегрированных в ЦП выделяют:

- Контроллер вложенных векторных прерываний (NVIC).
- Системный таймер (Sys Timer).
- Контроллер выхода из режима малого потребления энергии (WIC).

Интеграция такой периферии непосредственно в ЦП позволяет предельно минимизировать затраты времени на обращение к этим устройствам, в том числе, время реакции на прерывания, что является очень важным для систем управления реального времени.

Процессор может находиться в одном из следующих операционных режимов:

- User mode — обычный режим выполнения программ. В этом режиме выполняется большинство программ.
- Handler mode — режим, в который переходит процессор при возникновении исключения.

Переключение режима процессора происходит при возникновении соответствующего исключения или же модификацией регистра статуса.

ARM действует на основе RISC-команд, которые содержат готовый набор простейших элементов. [9]

Архитектура ARM обладает следующими особенностями RISC: [7]

- Архитектура загрузки/хранения;
- Нет поддержки нелинейного доступа к памяти (поддерживается в процессорах ARMv6, за некоторыми исключениями, и полностью в ARMv7 и более поздних);
- Равномерный 16х32-битный регистровый файл;
- Фиксированная длина команд (32 бита) для упрощения декодирования за счет снижения плотности кода. Позднее режим Thumb повысил плотность кода.
- Одноцикловое исполнение.

Чтобы компенсировать простой дизайн, в сравнении с современными процессорами были использованы некоторые особенности дизайна:

- Арифметические инструкции заменяют условные коды, только когда это необходимо;
- 32-битное многорегистровое циклическое сдвиговое устройство, которое может быть использовано без потерь производительности в большинстве арифметических инструкций и адресных расчетов.
- Мощные индексированные адресные режимы;
- Регистр ссылок для быстрого вызова функций листьев;
- Простые, но быстрые, с двумя уровнями приоритетов подсистемы прерываний с включенными банками регистров.

Архитектура предоставляет способ расширения набора команд, используя сопроцессоры.

В машинах на основе ARM периферийные устройства обычно подсоединяются к процессору путём сопоставления их физических регистров в памяти ARM или в памяти сопроцессора, или путём присоединения к шинам, которые, в свою очередь, подсоединяются к процессору.

1.2.3. Наборы команд микроконтроллеров ARM

В архитектуре ARM, на сегодняшний день, существует несколько наборов команд: ARM, Thumb, Thumb-2, Jazelle, A64.

Набор команд ARM представляет из себя базовый набор 32-битных команд.

Набор команд Thumb — альтернативный набор 16-разрядных команд, который использовался для улучшения плотности кода.

Набор команд Thumb-2 расширяет набор 16-разрядных команд Thumb дополнительными 32-разрядными, чтобы задать набору команд дополнительную ширину. Цель Thumb-2 — достичь плотности кода, как у Thumb, и производительности, как у набора команд ARM на 32 битах. В ARMv7 эта цель была достигнута.

Thumb-2 расширяет как команды ARM, так и команды Thumb ещё большим количеством команд, включая управление битовым полем, табличное ветвление, условное исполнение.

Jazelle — это технология, которая позволяет байткоду Java исполняться прямо в архитектуре ARM в качестве 3-го состояния исполнения (и набора команд) наряду с обычными командами ARM и режимом Thumb.

A64 — 64-битный набор команд, появившийся с приходом версии архитектуры ARMv8.

Как видно из схемы, микроконтроллеры ARM семейства Cortex M из всех перечисленных наборов команд поддерживают лишь наиболее универсальный Thumb-2. Помимо этого у этих микроконтроллеров нет поддержки сопроцессоров VFP и SIMD (NEON) (рис. 1.5).

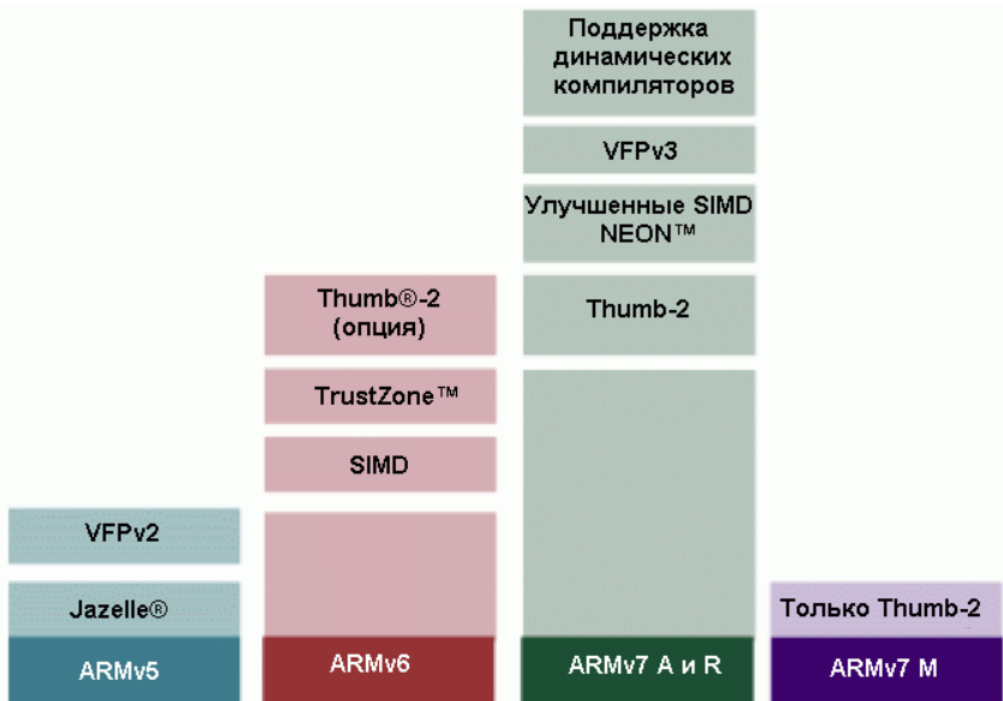


Рис. 1.5 Поддержка наборов команд разными версиями архитектуры ARM

1.2.4. Семейства и версии микроконтроллеров

Как было описано выше, в ARM были определены 3 профиля процессоров:

- A (application) — для устройств, требующих высокой производительности (например, смартфоны, планшеты);
- R (real time) — для приложений, работающих в реальном времени;
- M (microcontroller) — для микроконтроллеров, недорогих встраиваемых устройств.

Среди микроконтроллеров в настоящее время значимым являются семейство Cortex M [7].

Cortex M — новое семейство микроконтроллеров, призванное занять новую для ARM нишу встраиваемых решений низкой производительности. В семействе присутствуют четыре значимых ядра:

- Cortex-M0, Cortex-M0+ (более энергоэффективное) и Cortex-M1 (оптимизировано для применения в ПЛИС) с архитектурой ARMv6-M;
- Cortex-M3 с архитектурой ARMv7-M;
- Cortex-M4 (добавлены SIMD-инструкции, опционально FPU) и Cortex-M7 (FPU с поддержкой чисел одинарной и двойной точности) с архитектурой ARMv7E-M;
- Cortex-M23 и Cortex-M33 с архитектурой ARMv8-M.

1.2.5. Программная модель ARM-микроконтроллеров

Программная модель микроконтроллера ARM состоит из следующих блоков: регистры общего назначения, регистр управления Control, регистр команд, Сдвиговый регистр, регистр статуса программы, память программ (ROM), память данных (RAM), а также регистров модуля поддержки вычислений с плавающей точкой (FPU) и регистра статуса и управления FPU (последние два блока встречаются не во всех моделях микроконтроллеров).

Регистры общего назначения (РОН), которые также называют регистровым окружением центрального процессора, необходимы, в первую очередь, для работы операционного блока процессора (включает в себя арифметико-логическое устройство (АЛУ), аппаратный умножитель, аппаратный делитель, сдвиговый регистр), поскольку этот блок оптимизирован для работы только с регистровым окружением. При этом последние 3 регистра имеют специальные функции (рис. 1.6).

Регистр r15 служит счетчиком команд (PC) и содержит адрес следующей выполняемой команды.

Регистр r14 — регистр связи (LR). Содержит адрес возврата в основную программу после команды вызова подпрограммы.

Регистр r13 — регистр-указатель стека (SP). Состоит из двух регистров:

- PSP — указатель стека процесса пользователя (программ нижнего уровня);
- MSP — указатель стека обработчиков исключений и программ верхнего уровня.

Перед выполнением любой операции в регистры-источники должны быть предварительно загружены исходные данные — операнды. Результат операции также сохраняется в одном из «регистров окружения» ЦП (рис. 1.6).

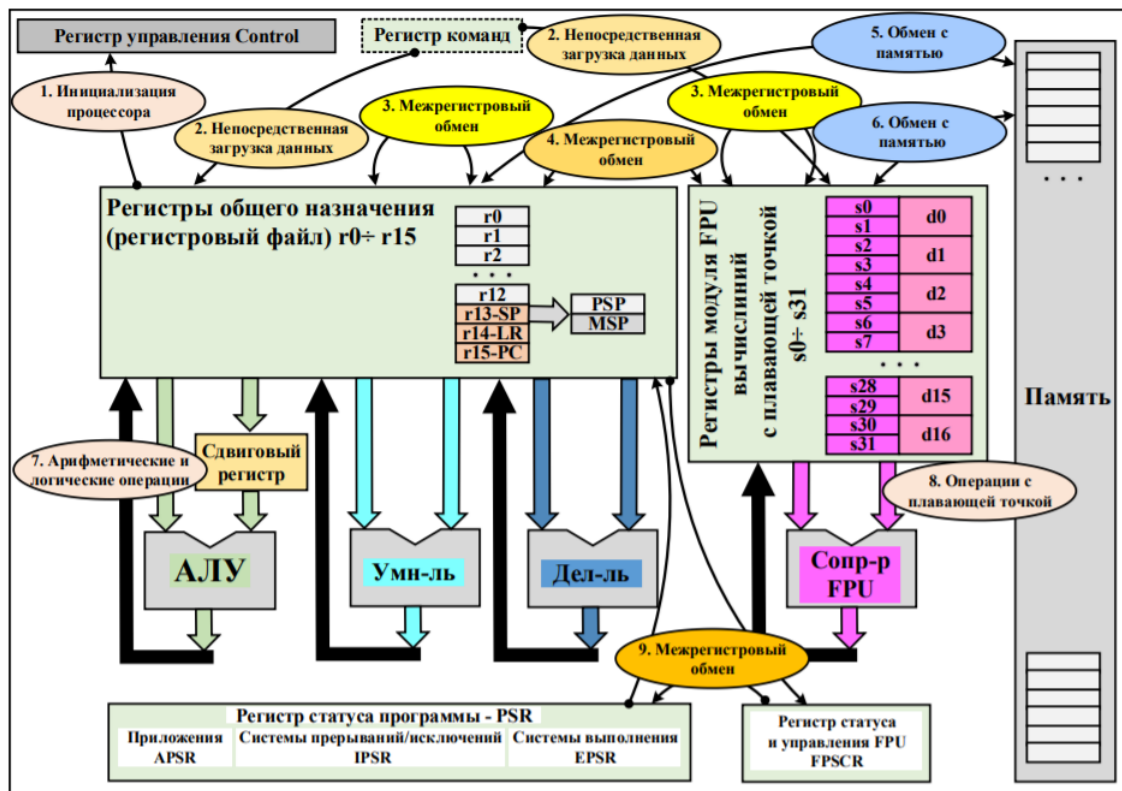


Рис. 1.6 Программная модель микроконтроллера ARM Cortex M [8]

Регистр управления Control — регистр специального назначения, который позволяет установить нужный режим работы процессора, задать тип доступа к памяти (привилегированный или непривилегированный), разрешить или запретить работу встроенных в процессор устройств, в частности, модуля поддержки вычислений с плавающей точкой (FPU).

Регистр статуса программы (PSR) состоит из трех регистров: регистр статуса программы-приложения (APSR), регистр статуса системы прерываний и исключений (IPSR), регистр статуса системы выполнения команд (EPSR). К каждому из регистров можно обращаться по его персональному имени. Также, по символическому имени можно обращаться к регистру PSR.

Все регистры статуса являются регистрами специального назначения и доступ к ним возможен только с помощью специальных команд обмена данными между регистрами общего назначения ЦП и регистрами специального назначения.

Регистр команд – содержит код операции, подлежащий расшифровке в данный момент.

В памяти программ (ROM) хранится исполняемый код программы, а также могут храниться некоторые данные (константы, таблицы констант).

В памяти данных (RAM) хранятся данные исполняемой программы, кроме того, в ней может храниться код программы. Также, RAM содержит побитово-, побайтово-адресуемую память.

Регистры модуля поддержки вычислений с плавающей точкой (FPU), которые еще называют регистровым окружением сопроцессора FPU, необходимы для хранения операндов, которые используются модулем FPU. Поскольку, FPU может выполнять операции только над операндами, расположенными в его регистрах окружения.

Имеется 32 регистра s0-s31, предназначенных для хранения данных в формате с плавающей точкой однократной точности. Все регистры 32-разрядные. Каждая пара регистров s0-s31 с точки зрения программиста может рассматриваться как один 64-разрядный регистр. Он может содержать данные в формате с плавающей точкой двойной точности или два слова с плавающей точкой однократной точности. Всего в окружении сопроцессора 16 таких регистров с именами d0-d15.

Регистр статуса и управления модулем поддержки вычислений с плавающей точкой (FPSCR) обеспечивает полный контроль программиста за режимом работы и текущим состоянием модуля FPU. Он позволяет установить нужный для приложения пользователя режим работы, проверить текущее состояние вычислительного процесса по результатам сравнения переменных в формате с плавающей точкой или на предмет возникновения исключительной ситуации (возможной некорректности выполненных вычислений с операндами в формате с плавающей точкой).

В соответствии с этими тремя важнейшими функциями в регистре имеются три группы битовых полей: (r31-r28) — флаги результатов операций сравнения чисел с плавающей точкой; (r26-r22) — биты управления режимом работы сопроцессора; (r7-r0) — биты флагов исключительных ситуаций, возникающих при обработке данных.

1.2.6. Периферия

Периферия микроконтроллеров с архитектурой ARM хорошо развита и включает в себя порты ввода-вывода, система прерываний, АЦП, таймеры общего назначения, сторожевой таймер, расширенный таймер, часы реального времени, последовательный периферийный трехпроводный интерфейс SPI, модуль I2C, модуль УСАПП и др [10].

Порты ввода-вывода (GPIO) микроконтроллеров ARM являются двунаправленными и, в зависимости от модели микроконтроллера, могут содержать до 80 линий и больше. При этом, для каждого порта имеется 2 конфигурационных регистра, благодаря которым можно настроить порты, узнать в каком направлении работает линия: на ввод или на вывод, а также защитить конфигурационные параметры с помощью регистра блокировки.

Прерывания в микроконтроллерах ARM делится на внешние и внутренние.

Внутренние прерывания осуществляются при помощи контроллера вложенных векторизованных прерываний (КВВП). У любого Cortex-микроконтроллера будет присутствовать одна и та же структура прерываний, независимо от его производителя. Таким образом, прикладной код и операционные системы можно легко портировать с одного МК на любой другой.

КВВП поддерживаются вложенные прерывания и, зачастую, в МК используется несколько уровней приоритетов. Структура прерываний КВВП разработана с учетом программирования полностью на C и исключает потребность в написании каких-либо ассемблерных макросов или специальных, несовместимых с ANSI, директив. О работе системы прерываний подробнее написано в главе об AVR-микроконтроллерах.

Блок внешних прерываний имеет линии прерывания, которые связываются с векторами прерываний посредством КВВП. Блок внешних прерываний играет важную роль для управления энергопотреблением микроконтроллера. Данный блок является асинхронным и, поэтому, может использоваться

для возобновления работы микроконтроллера, находящегося в режиме STOP, когда оба основных генератора отключены.

Аналого-цифровой преобразователь необходим для получения числового значения напряжения, поданного на его вход. В зависимости от модели микроконтроллера в него может быть встроено от 1 до 3 аналого-цифровых преобразователей. Вся настройка АЦП выполняется через регистры управления и статуса. Режим работы АЦП задается через два регистра управления.

АЦП питаются отдельным напряжением, которое в зависимости от типа корпуса может находиться в пределах 2.4-3.6 В. Источник опорного напряжения АЦП соединен либо внутренне с напряжением питания АЦП, либо со специальными внешними выводами. АЦП характеризуется 12-битной разрешающей способностью и частотой преобразования 1 МГц. У него имеется до 18 мультиплексированных каналов, 16 из которых можно использовать для измерения внешних сигналов. Оставшиеся два канала связаны со встроенным датчиком температуры и внутренним источником опорного напряжения.

Также АЦП поддерживают функцию оконного компаратора, которая заключается в генерации прерывания при выходе результата преобразования за пределы заданных пользователем нижней и верхней границ.

Все блоки таймеров общего назначения выполнены на основе 16-битного перезагружаемого счетчика, который синхронизируется с выхода 16-битного предделителя. Перезагружаемое значение хранится в отдельном регистре. Счет может быть прямой, обратный или двунаправленный. Вход синхронизации счетчика можно связать с одним из восьми различных источников. В их число входят: специальный сигнал синхронизации, производный от сигнала главной системной синхронизации; выходной сигнал синхронизации одного из других таймеров или внешний сигнал синхронизации, связанный с выводами захвата/сравнения.

Каждый из таймеров может генерировать прерывания и поддерживает прямой доступ к памяти (DMA).

Расширенный таймер архитектурно похож на таймеры общего назначения с той разницей, что он дополнен рядом аппаратных узлов, предназначенных для управления электродвигателем.

В каждом из трех каналов этого таймера предусмотрены два противофазных выхода. Таким образом, он представляет собой 6-канальный ШИМ-блок.

Входящие в МК часы реального времени (ЧРВ) представляют собой 32-битный счетчик, оптимизированный под счет секунд при синхронизации

частотой 32.768 кГц. Во время настройки системы синхронизации, в качестве источника синхронизации часов реального времени можно выбрать внутренний низкочастотный генератор, внешний низкочастотный генератор или внешний высокочастотный генератор. После выбора источник синхронизации соединяется с часами реального времени через делитель частоты с фиксированным коэффициентом деления. Для организации точного счета секунд далее может быть задействован еще один предделитель ЧРВ.

Сброс МК может быть выполнен встроенными сторожевыми таймерами, программно через контроллер вложенных векторизованных прерываний, встроенными схемами сброса при подаче/отключении и снижении ниже допустимого уровня напряжения питания.

В МК на базе архитектуры ARM, зачастую, входят два типа отдельных сторожевых таймера: независимый и оконный. Независимый сторожевой таймер полностью отделен от основной системы МК. Он расположен в домене с резервированием питания и синхронизируется встроенным низкочастотным генератором (LSI). Оконный сторожевой таймер является частью основной системы МК и связан с сигналом синхронизации первой шины устройства ввода-вывода. Оба сторожевых таймера поддерживают возможность раздельного включения/отключения и могут использоваться одновременно.

Для организации быстродействующей связи с интегральными схемами у микроконтроллеров ARM имеется модуль SPI, предназначенный для полнодуплексной передачи данных. Подробнее про этот интерфейс написано в главе про микроконтроллеры AVR.

Для связи с интегральными схемами у МК имеется еще один специальный интерфейс — I2C. Интерфейс I2C может работать в ведущем или подчиненном режиме и поддерживает возможность арбитра шины, что необходимо в мультимастерных системах.

Модуль I2C использует 7- и 10-битные режимы адресации. Модуль полностью реализует протокол передачи данных по шине и требует для управления только необходимой протоколу передачи информации. Модуль I2C может генерировать два прерывания: одно при обнаружении ошибок и другое для управления адресом связи и передаваемыми данными. Кроме того, блок DMA предоставляет два канала, которые можно использовать для чтения из буфера передачи и записи данных в этот буфер. Таким образом, сразу после задания адреса и подлежащих передаче данных можно начать двунаправленную передачу данных полностью под аппаратным управлением.

Модуль I2C является аналогом интерфейса TWI в МК AVR.

Несмотря на то, что порты последовательной связи уже практически не используются в ПК, они все еще остаются популярными во многих встраиваемых применениях для организации простого интерфейса последовательной связи. В ARM-микроконтроллерах интегрируется от одного до нескольких модулей УСАПП, каждый из которых поддерживает несколько расширенных режимов работы, позволяющие использовать МК в современных коммуникационных применениях.

Каждый УСАПП имеет собственный генератор скорости связи с возможностями дробного деления частоты. В отличие от обычных делителей частоты, такой генератор позволяет получить стандартные скорости связи при любой частоте синхронизации шины. Также как и остальные модули последовательных интерфейсов, каждый модуль УСАПП оснащен двумя каналами прямого доступа к памяти для двунаправленной связи с буфером данных. В конфигурации УАПП, модуль УСАПП может работать в нескольких режимах работы. УСАПП имеет возможность работы с однопроводной полудуплексной линией. Для связи с модемами, а также для аппаратного управления передачей потока данных, у каждого УСАПП предусмотрены дополнительные линии управления CTS и RTS.

Помимо работы в роли высокоскоростного интерфейса, каждый УСАПП может быть переведен в режим синхронной связи. Его можно использовать для связи с внешними устройствами ввода-вывода, оснащенными SPI-совместимым интерфейсом, по 3-проводной линии. Работая в данном режиме, УСАПП работает в роли ведущего шины SPI и поддерживает возможность программирования полярности и фазы синхронизации. Благодаря этому, возможна связь с практически любой подчиненной SPI ИС.

Как и микроконтроллеры AVR, ARM-микроконтроллеры поддерживают такие модули как CAN, USB, LCD. Подробнее о них написано в главе про AVR [10].

1.2.7. Питание

Для работы микроконтроллеров ARM их необходимо питать одним напряжением в диапазоне от 1.2 до 5.5 В. В домене с отдельным питанием расположены часы реального времени и небольшое число регистров, что позволяет организовать их резервное питание и обеспечить работоспособность даже при нахождении МК в режиме полного отключения (deep power down) [10].

ARM микроконтроллер может работать в одном из следующих режимов [11]:

- Run — активный режим работы. В данном режиме процессорное ядро активно и может работать с максимальным быстродействием. Память и периферия доступны без каких-либо ограничений, программный код выполняется из SRAM или Flash;
- Low-Power Run — малопотребляющий активный режим с отключенным основным регулятором (MR) и питанием домена VCORE от малопотребляющего регулятора (LPR). В данном режиме ограничена максимальная частота системы и возможность записи и стирания Flash, однако программный код по-прежнему может выполняться из SRAM или Flash;
- Sleep — режим работы, при котором тактирование ядра отключено, периферия (в том числе системная), а также память находятся в активном состоянии. Выход из режима и пробуждение процессора происходят при возникновении определенных событий или прерываний;
- Low-Power Sleep — режим, сравнимый по принципу работы со Sleep, но для питания домена VCORE используется регулятор (LPR), а основной регулятор (MR) отключается для снижения энергопотребления;
- Stop 0 — при переходе в этот режим содержимое памяти SRAM и всех регистров фиксируется. Все сигналы тактирования в домене VCORE останавливаются. Генераторы отключены. Однако некоторые периферийные блоки могут активировать HSI16 для своих потребностей. Для тактирования используются LSI и LSE. В режиме Stop 0 используется регулятор MR, который обеспечивает минимальную задержку при пробуждении ядра, но при этом увеличивает энергопотребление;
- Stop 1 — режим, аналогичный Stop 0, но для питания домена VCORE используется регулятор LPR, а регулятор MR отключается с целью снижения потребления, однако такой подход увеличивает время пробуждения контроллера;
- Standby With SRAM2 — режим ожидания с памятью SRAM2. При работе в данном режиме питание домена VCORE отключено, однако содержимое SRAM2 может быть сохранено. Питание SRAM2 осуществляется от регулятора LPR. Все сигналы тактирования в домене VCORE, а также генераторы — остановлены. Для тактирования могут использоваться LSI и LSE. Выход из режима может происходить при сбросе,

некоторых событиях, а также при подаче сигнала на соответствующие выводы микроконтроллера;

- Standby Without SRAM2 — аналогичен предыдущему режиму, однако содержимое SRAM2 не сохраняется. Регулятор LPR отключен;
- Shutdown — режим, при котором питание домена VCORE отключено. Все сигналы тактирования в домене VCORE, а также генераторы — остановлены. Для тактирования доступен только LSE. В данном режиме также отключаются блоки мониторинга напряжения.

2. ПРОГРАММНОЕ И АППАРАТНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ AVR И ARM МИКРОКОНТРОЛЛЕРОВ

2.1. Arduino

Arduino — торговая марка аппаратно-программных средств для построения и прототипирования простых систем, моделей и экспериментов в области электроники, автоматики, автоматизации процессов и робототехники [12].

2.1.1. Программная часть

Программная часть состоит из бесплатной программной оболочки (IDE) для написания программ, их компиляции и программирования аппаратуры. Аппаратная часть представляет собой набор смонтированных печатных плат, продающихся как официальным производителем, так и сторонними производителями. Полностью открытая архитектура системы позволяет свободно копировать или дополнять линейку продукции Arduino (рис. 2.1).

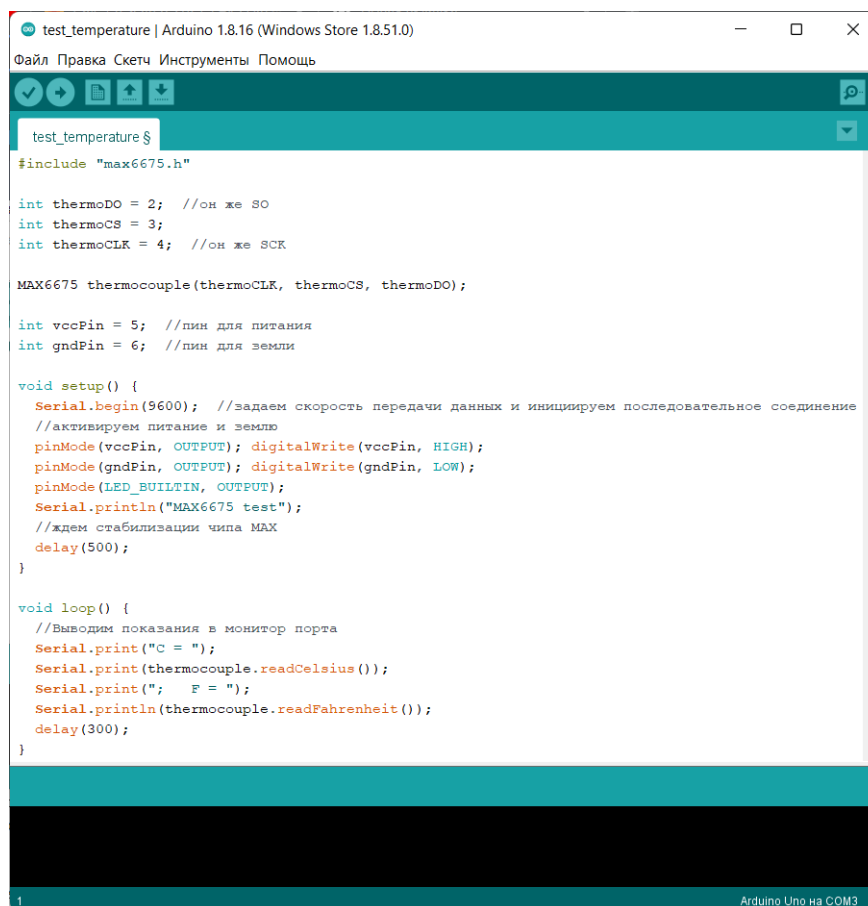


Рис. 2.1 Arduino IDE с примером простой программы

Используется как для создания автономных объектов, так и подключения к программному обеспечению через проводные и беспроводные интерфейсы. Подходит для начинающих пользователей с минимальным входным порогом знаний в области разработки электроники и программирования.

Программирование ведется целиком через собственную бесплатную программную оболочку Arduino IDE. В этой оболочке имеется текстовый редактор, менеджер проектов, препроцессор, компилятор и инструменты для загрузки программы в микроконтроллер. Оболочка написана на Java на основе проекта Processing, работает под Windows, Mac OS X и Linux. Используется комплект библиотек Arduino.

2.1.2. Язык программирования

Язык программирования Arduino называется Arduino C и представляет собой язык C++ с фреймворком Wiring, он имеет некоторые отличия по части написания кода, который компилируется и собирается с помощью avr-gcc, с особенностями, облегчающими написание работающей программы — имеется набор библиотек, включающий в себя функции и объекты. При компиляции программы IDE создает временный файл с расширением *.cpp.

Программы, написанные программистом Arduino, называются скетчи и сохраняются в файлах с расширением *.ino. Эти файлы перед компиляцией обрабатываются препроцессором Arduino. Также существует возможность создавать и подключать к проекту стандартные файлы C++.

Программист должен написать две обязательные для Arduino функции `setup()` и `loop()`. Первая вызывается однократно при старте, вторая выполняется в бесконечном цикле.

В текст своей программы (скетча) программист не обязан вставлять заголовочные файлы используемых стандартных библиотек. Эти заголовочные файлы добавит препроцессор Arduino в соответствии с конфигурацией проекта. Однако пользовательские библиотеки нужно указывать.

Менеджер проекта Arduino IDE имеет нестандартный механизм добавления библиотек. Библиотеки в виде исходных текстов на стандартном C++ добавляются в специальную папку в рабочем каталоге IDE. При этом название библиотеки добавляется в список библиотек в меню IDE. Программист отмечает нужные библиотеки, и они вносятся в список компиляции.

Arduino IDE не предлагает никаких настроек компилятора и минимизирует другие настройки, что упрощает начало работы для новичков и умень-

шает риск возникновения проблем; но присутствуют директивы препроцессора, такие как `#define`, `#include` и много других.

Закачка программы в микроконтроллер Arduino происходит через предварительно запрограммированный специальный загрузчик (все микроконтроллеры от Arduino продаются с этим загрузчиком). Загрузчик создан на основе Atmel AVR Application Note AN109. Загрузчик может работать через интерфейсы RS-232, USB или Ethernet в зависимости от состава периферии конкретной процессорной платы. В некоторых вариантах, таких как Arduino Mini или неофициальной Boarduino, для программирования требуется отдельный переходник.

Пользователь может самостоятельно запрограммировать загрузчик в чистый микроконтроллер. Для этого в IDE интегрирована поддержка программатора на основе проекта AVRdude. Поддерживается несколько типов популярных дешёвых программаторов.

2.1.3. Аппаратная часть

Под торговой маркой Arduino выпускается несколько плат с микроконтроллером и платы расширения. Большинство плат с микроконтроллером снабжено минимально необходимым набором обвязки для нормальной работы микроконтроллера (стабилизатор питания, кварцевый резонатор, цепочки сброса и т. п.).

В концепцию Arduino не входит корпусного или монтажного конструктива. Разработчик выбирает метод установки и механической защиты плат самостоятельно либо с помощью сторонних компаний. Сторонними производителями также выпускаются наборы робототехнической электромеханики, ориентированной на работу совместно с платами Arduino. Независимыми производителями также выпускается большое количество всевозможных датчиков и исполнительных устройств, в той или иной степени совместимых с Arduino (рис. 2.2) [13].

2.1.4. Классический конструктив

Классические Arduino и Arduino-совместимые платы спроектированы для монтажа в стопки через штыревые разъёмы. Таким образом базовую микропроцессорную плату дополняют необходимой периферией и внешними подключениями.



Рис. 2.2 Отладочная плата Arduino Uno

Существуют платы Uno, Pro, Leonardo, Mega 2560, Due и платы, например Zero, с расширенным набором штыревых разъёмов для них. Платы расширения стандартной длины могут устанавливаться и в расширенные процессорные платы [14].

2.1.5. Миниатюрный конструктив

Arduino

Выпускаются отдельные платы уменьшенных габаритов — Nano, Nano Every и Micro — в габарите DIP-корпусов микросхем. Они предназначены для установки в макетные платы. Плат расширения для них нет.

Позже была выпущена линейка Arduino MKR в похожем конструктиве. К ним есть небольшой набор плат расширения периферии.

Сторонние проекты

Помимо стандартных конструктивов Arduino, сторонние разработчики создали множество миниатюрных клонов, сохранив только архитектурную и программную совместимость. Среди этих клонов выделяется линейка продуктов Microduino. Линейка содержит полноценный набор конструктивно

совместимых процессорных модулей, модулей связи, датчиков и исполнительных устройств, практически не уступая ассортименту классических модулей Arduino. Как и Arduino, сборка плат производится в стопки. Линейка оформлена в двух оригинальных конструктивах:

- Бескорпусной, с соединениями на миниатюрных цанговых штыревых линейках.
- В стиле конструкторов Лего с электрическими соединениями на подпружиненных контактах и механической фиксацией, совместимой с конструкторами Лего.

2.1.6. Микроконтроллер

Микроконтроллеры для Arduino отличаются наличием предварительно прошитого в них загрузчика. С помощью этого загрузчика пользователь загружает свою программу в микроконтроллер без использования традиционных отдельных аппаратных программаторов. Загрузчик соединяется с компьютером через интерфейс USB (если он есть на плате) или с помощью отдельного переходника UART-USB. Поддержка загрузчика встроена в Arduino IDE.

На случай затирания загрузчика или покупки микроконтроллера без загрузчика разработчики предоставляют возможность прошить загрузчик в микроконтроллер самостоятельно. Для этого в Arduino IDE встроена поддержка нескольких популярных дешевых программаторов, а большинство плат Arduino имеет штыревой разъём для внутрисхемного программирования (ICSP для AVR, JTAG или SWD для ARM).

В Arduino IDE встроена возможность создания своих программно-аппаратных платформ. Этой возможностью пользуются сторонние компании, которые добавляют в Arduino IDE свои наборы плат и компиляторов-загрузчиков к ним.

AVR

В классической линейке устройств Arduino в основном применяются микроконтроллеры Atmel AVR. Следующие микроконтроллеры можно встретить на указанных распространённых платах: [15]

- ATmega2560 (16 МГц, 256 Kb Flash, 8 Kb RAM, 54 порта, из них до 15 с ШИМ и 16 АЦП). Платы Mega.
- ATmega32U4 (16 МГц, 32 Kb Flash, 2,5 Kb RAM, 20 портов, из них до 7 с ШИМ и 12 АЦП). Платы Leonardo, Micro, Yun.

- ATmega328 (16 МГц, 32 Kb Flash, 2 Kb RAM, 14 портов, из них до 6 с ШИМ и 8 АЦП). Платы UnoR3, Mini, NanoR2, Pro, Pro mini, различные варианты плат uno и nano, такие как Wifi Uno и nano + nrf42l01
- ATtiny85 (20 МГц, 8 Kb Flash, 512 b RAM, 6 портов, из них 4 ШИМ и 4 аналоговых). Платы Digispark, также часто применяются вне плат.
- ATmega168 (16 МГц, 16 Kb Flash, 1 Kb RAM, порты и распиновка аналогично ATmega328) Платы Uno R1, Uno R2, Pro mini, NanoR1.

В некоторых платах состав доступных портов и частота тактирования могут отличаться.

ARM

Постепенно в линейке плат стали появляться процессоры ARM. Первоначально это был AT91SAM3X8E на плате классического конструктива (Due). Позже появилась линейка плат Arduino MKR в конструктиве DIP, оснащенная контроллером SAMD21 (Cortex-M0, 48 MHz, 256 Kb Flash, 32 Kb RAM).

С 2020 года в таком же конструктиве MKR появились модули Portenta с ARM Cortex-M7 (STM32H747 @ 480 МГц).

ESP8266

Сторонние разработчики портировали в Arduino поддержку популярного Wi-Fi микроконтроллера ESP8266 и его клона ESP12. Компилировать и загружать прошивку для ESP8266 со своими скетчами и поддержкой Wi-Fi можно прямо из Arduino IDE, получая одноплатную схему с поддержкой сети Wi-Fi.

Intel x86

В рамках сотрудничества со сторонними производителями в Arduino IDE была включена поддержка некоторых аппаратных средств Intel x86. Intel Galileo, Intel Edison и Arduino 101 — Arduino-совместимые платы на Intel x86 архитектуре. Платы механически и электрически совместимы с периферийными платами Arduino. Платы функционируют под собственной ОС Linux, поверх которой работает приложение, позволяющее загружать и исполнять скетчи Arduino.

2.1.7. Периферия

Порты ввода-вывода микроконтроллеров оформлены в виде штыревых линеек. Никакой буферизации, защиты, конвертации уровней, как правило, нет. Микроконтроллеры питаются от 5 В или 3,3 В в зависимости от модели платы. Соответственно, порты имеют такой же размах допустимых входных и выходных напряжений. Программисту доступны некоторые специальные возможности портов ввода-вывода микроконтроллеров, например широтно-импульсная модуляция (ШИМ), аналогово-цифровой преобразователь (АЦП), интерфейсы UART, SPI, I2C. Количество и возможности портов ввода-вывода определяются конкретным вариантом микропроцессорной платы.

Помимо портов, на платах микроконтроллеров иногда устанавливается периферия в виде интерфейсов USB или Ethernet. Опциональный набор внешней периферии на модулях расширения включает в себя:

- USB Device (чаще всего как виртуальный COM порт, имеются также версии с эмуляцией USB HID Class клавиатур и мышек).
- Проводной и беспроводной Ethernet как на основной плате, так и на платах расширения.
- Модуль GSM и другие беспроводные интерфейсы.
- USB Host.
- SD card.
- Модуль управления низковольтным мотором.
- Графический ЖКИ-индикатор.
- Модуль с макетным полем.

Сторонние производители выпускают широкий выбор датчиков и исполнительных устройств, подключаемых к Arduino. Например, гироскопы, компасы, манометры, гигрометры, термометры, релейные модули, индикаторы, клавиатуры и т. п.

FPGA

Существуют совместимые с Arduino процессорные платы, на которых в качестве периферийного устройства установлена микросхема программируемой логики (FPGA). Например, сама компания Arduino выпускает плату Arduino MKR Vidor 4000, на которой, помимо процессора, установлена ПЛИС Intel Cyclone. Программист в среде Arduino может загружать в FPGA предустановленные функции, например, работу с изображениями, звуком,

дополнительные порты UART, SPI, ШИМ и т.д. Однако, свободное программирование FPGA из среды Arduino не предусмотрено, для этого необходимо использовать среду разработки производителя FPGA — Intel Quartus.

Также существует проект Papilio, в котором развивается совместимая с Arduino линейка плат с программируемой логикой Xilinx в качестве периферии. Помимо готовых решений для использования FPGA в качестве периферии, проект предлагает интеграцию среды программирования Arduino и среды программирования FPGA Xilinx ISE schematic editor. Пользователь может редактировать FPGA аналогично рисованию электрических схем.

2.2. Другое популярное программное обеспечение для AVR и ARM микроконтроллеров

Помимо Arduino IDE большой популярностью пользуются следующие решения: [16] [17]

1. Atmel Studio — интегрированная среда разработки (IDE) от компании Atmel для разработки приложений под микроконтроллеры ARM Cortex-M и AVR. Распространяется бесплатно.
2. AVRDUDE — интерфейс программы консольный, предназначена, чтобы изменять и записывать данные в памяти устройств с AVR архитектурой. В программе применяется технология внутрисхемного программирования. Распространение свободное.
3. WinAVR — мощная среда разработки с открытым исходным кодом, созданная с целью написания программ для микроконтроллеров серии AVR от компании Atmel. Распространяется свободно и бесплатно.
4. VM LAB — комплекс утилит для создания и настройки кода программы, наряду с этим создает модель работы устройства с контроллерами AVR серии. Распространяется свободно.
5. MPLAB — единая бесплатная интегрированная среда разработки для контроллеров производства Microchip.
6. CoocoxCoIDE — работает с устройствами в чью архитектуру заложен ARM, как программная среда с высокой степенью интеграции.

Также существует еще ряд программ: BascomAVR, CodeVisionAVR, WinPic800, PICPgm, Keil uVision, IAR Embedded Workbench, Flow Code, Atollic TrueSTUDIO, Sourcery CodeBench и др. Но такие решения в основном предоставляются на платной основе или с урезанным функционалом для бесплатных версий.

ЗАКЛЮЧЕНИЕ

В результате научно-исследовательской работы были выполнены следующие задачи:

1. Осуществлено знакомство с архитектурой микроконтроллеров AVR и ARM на примере представителя каждого семейства.
2. Осуществлено знакомство с наборами команд для микроконтроллеров семейств AVR и ARM.
3. Изучены некоторые аппаратные и программные средства для разработки под AVR и ARM микроконтроллеры, подходящие для дальнейшего обучения и разработки на начальном уровне.

В дальнейшем предполагается применение полученных знаний для разработки собственного программного продукта на выбранной аппаратно-программной базе.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. *Кочегаров И. И. Трусев В. А.* Микроконтроллеры семейства AVR. — 2012. — URL: https://dep_kipra.pnzgu.ru/files/dep_kipra.pnzgu.ru/spravochnai_i_metodicheskai_literatyra/mikrokontrollery_avr.pdf.
2. [Электронный ресурс] AVR — Википедия. — URL: <https://ru.wikipedia.org/wiki/AVR>.
3. *Пьявченко А. О. Переверзев В. А.* Архитектура, основы программирования и применения AVR-микроконтроллеров и ARM-микросистем. — 2021. — URL: https://www.google.ru/books/edition/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D1%8B_%D0%BF%D1%80/GEUkEAAAQBAJ?hl=ru&gbpv=0.
4. [Электронный ресурс] Архитектура микроконтроллеров, из чего состоит микроконтроллер — ph0en1x.net. — URL: <https://ph0en1x.net/69-avr-microcontroller-architecture-what-is-inside-avr-microcontroller.html>.
5. *Водовозов А. М.* Микроконтроллеры для систем автоматики. — 2016. — URL: https://www.google.ru/books/edition/%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%BB%D0%B5%D1%80%D1%8B_%D0%B4%D0%BB%D1%8F/Qe7TDAAAQBAJ?hl=ru&gbpv=0.
6. *Ревич Ю. В.* Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. — 2014. — URL: https://www.google.ru/books/edition/%D0%9F%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC/6iU4BAAAQBAJ?hl=ru&gbpv=0.
7. [Электронный ресурс] ARM — Википедия. — URL: [https://ru.wikipedia.org/wiki/ARM_\(%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0\)#%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0](https://ru.wikipedia.org/wiki/ARM_(%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0)#%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0).
8. *Козаченко В. Ф.* Практический курс микропроцессорной техники на базе процессорных ядер ARM-Cortex-M3/M4/M4F. — 2019. — URL: <http://motorcontrol.ru/wp-content/uploads/2019/04/%D0%9F%D1%80%D0%>

B0%D0%BA%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BA%D1%83%D1%80%D1%81_%D0%BC%D0%B8%D0%BA%D1%80%D0%BE%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80.pdf.

9. [Электронный ресурс] Процессоры ARM — you2you.ru. — URL: <https://you-2-you.ru/articles/interesnoe/protssessoryi-arm-chto-eto-takoe-i-schem-ih-edyat>.
10. [Электронный ресурс] Ознакомительное руководство по ARM-микроконтроллерам Cortex-M3 — gaw.ru. — URL: http://www.gaw.ru/html.cgi/txt/doc/micros/arm/cortex_arh/index.htm.
11. [Электронный ресурс] STM32G4. Системная архитектура и режимы питания — compel.ru. — URL: <https://www.compel.ru/lib/139736>.
12. [Электронный ресурс] Arduino — Википедия. — URL: <https://ru.wikipedia.org/wiki/Arduino#%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%BB%D0%B5%D1%80>.
13. [Электронный ресурс] Отладочная плата Arduino Uno — platan.ru. — URL: <https://www.platan.ru/cgi-bin/qwery.pl/id=91278075>.
14. *Аливерти П.* Изучаем Arduino.Руководство для начинающих. — 2021. — URL: https://www.google.ru/books/edition/%D0%98%D0%B7%D1%83%D1%87%D0%B0%D0%B5%D0%BC_Arduino_%D0%A0%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81/E4RDEAAAQBAJ?hl=ru&gbpv=0.
15. *Блум Д.* Изучаем Arduino: инструменты и методы технического волшебства. — 2015. — URL: https://www.google.ru/books/edition/%D0%98%D0%B7%D1%83%D1%87%D0%B0%D0%B5%D0%BC_Arduino_%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5/aqsUCAAQBAJ?hl=ru&gbpv=0.
16. [Электронный ресурс] Программы для микроконтроллеров — radio-magic.ru. — URL: <https://radio-magic.ru/programmu-dlya-radiolubitelej/programmu-mk>.
17. [Электронный ресурс] Программы для микроконтроллеров — cxem.net. — URL: https://cxem.net/software/soft_mcu.php.