

# Занятие 1: Устройство микроконтроллера

## Теоретическая часть

### Понятие "микроконтроллер".

Микроконтроллер это микросхема, сочетающая в себе процессор, встроенную память и периферию. Следует различать микроконтроллер и микропроцессор. Последний не имеет встроенной памяти и требует для запуска дополнительных микросхем. Итак, микроконтроллер представляет собой законченный компьютер в миниатюре. Постоянная память выполняет функцию винчестера, перезаписываемая память представляет собой оперативную память компьютера, процессор уступает современным процессорам для компьютера в скорости и масштабности, но способен на выполнение любых вычислительных задач; наконец периферия позволяет микроконтроллеру взаимодействовать с пользователем: принимать от него команды, выводить результаты и взаимодействовать с любыми электронными устройствами, к которым он подключен.

### Семейства микроконтроллеров.

В отличие от передовых процессоров для компьютера, микроконтроллеры выпускают множество фирм. Существуют и отечественные микроконтроллеры. Для удобства микроконтроллеры делят на семейства, обладающие например общей системой команд или общей архитектурой. Наиболее известные микроконтроллеры приведены в Таблице 1.1. Выбор семейства микроконтроллеров зависит от поставленной задачи. Важными

Название	Разрядность	Скорость	Производители	Описание
8051	8	Низкая-Средняя	Atmel, Infineon, Silicon Labs, STMicroelectronics, Maxim, ...	Представитель наиболее распространённого семейства микроконтроллеров. За счёт малой разрядности не подходит для вычислительных задач, но прекрасно справляется с задачами управления.
ARM	32	Средняя-Высокая	Analog Devices, Atmel, Luminary Micro, NXP, ...	Наиболее распространённое семейство высокопроизводительных микроконтроллеров. Оно получило такую популярность за счёт простого ядра с малым энергопотреблением.
PIC	8/16/32	Низкая/ /Средняя/ /Высокая	Microchip Technology	Семейство микроконтроллеров с долгой историей, за время которой разрядность росла и стала охватывать практически весь спектр задач.
AVR	8(+16)/32	Средняя/ /Высокая	Atmel	Семейство микроконтроллеров, отличающееся простотой и продвинутой архитектурой. 32-разрядная часть семейства претендует на убийцу архитектуры ARM за счёт большего быстродействия и меньшего энергопотребления.

параметрами являются разрядность ядра (см. далее), доступность программного обеспечения, цена, энергопотребление. Скорость ядра, количество входов и выходов, возможности периферии, - можно выбирать в пределах одного семейства.

## Применение микроконтроллеров.

Возможности и цена микроконтроллеров позволяют им заменить практически все цифровые микросхемы, предназначенные для управления электроникой. В тех областях, где производительности не хватает используются модифицированные микроконтроллеры с функциями быстрого вычисления определённых задач (цифровой сигнальный процессор). Микроконтроллеры используются в: автомобилях, наручных электронных часах, тамагочи, тетрисах, плеерах, сотовых телефонах и т. д. Возможности по перепрограммированию (далее "прошиванию") микроконтроллеров позволяют исправлять допущенные ошибки и/или наращивать функциональность устройств.

### Структура микроконтроллера: ядро, регистры, память, кэш, общий ввод-вывод, таймеры, прерывания, watchdog, периферия, возможность отладки.

Мы уже знаем, что микроконтроллер похож на компьютер и теперь необходимо глубже понять устройство этого компьютера. На Рис 1.1 приведена схема микроконтроллера. Основой работы микроконтроллера является процессор. Он способен выполнять набор операций с данными, определённый его системой команд. Сами команды находятся в памяти и называются программой или "прошивкой". Следует различать постоянную память, в которой хранится прошивка, и переменную, в которой хранятся переменные программы. Команды выполняются по очереди и результаты работы команд сораниются в память. Однако даже простая команда сложения в таком случае будет содержать 4 обращения к памяти: загрузка команды, загрузка первого слагаемого, загрузка второго слагаемого, запись суммы. Такое интенсивное общение с памятью становится узким местом в производительности системы. Для решения этой проблемы ввели специальную очень маленькую память, называемую регистрами. Регистры не расположены в памяти, у них нет адреса и благодаря этому канал связи с памятью

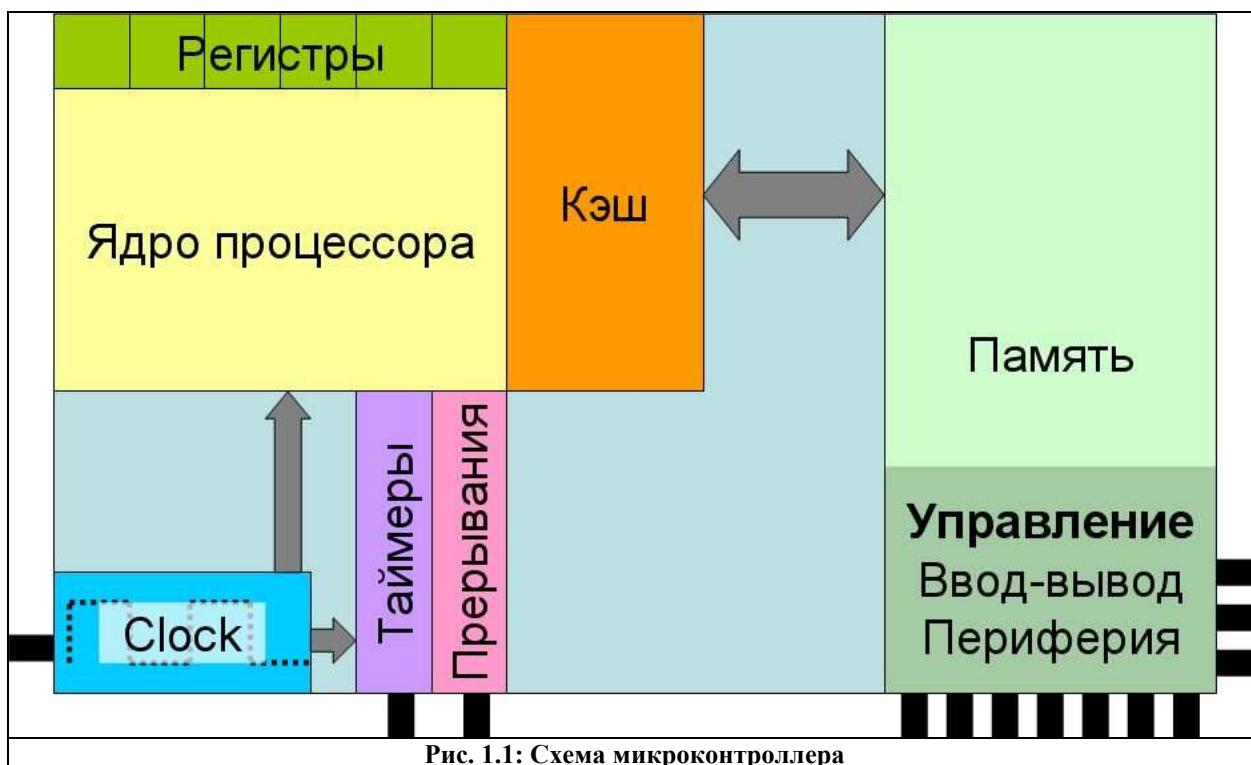


Рис. 1.1: Схема микроконтроллера

становится нужен только когда требуется информация не находящаяся в регистрах процессора. Например задача суммирования элементов массива может быть выполнена всего с двумя обращениями к памяти на элемент массива: загрузка команды, загрузка второго слагаемого. Первое слагаемое и сумма будут занимать только один регистр

(сначала там будет первое слагаемое, а потом туда запишется сумма). Однако использование регистров помогает решить проблему только для малого количества операндов в программе. Когда регистры кончаются, то приходится часть временно переписывать в память. Увеличить число регистров нельзя так как их количество чётко задано архитектурой семейства микроконтроллера. Но возможно добавить дополнительную быструю память, которая будет работать независимо от основной памяти. Такая память называется кэш. Он недоступен для прошивки и находится в подчинении только у процессора (иногда это не так, но скорее в порядке исключения). Использование кэша позволяет еще сильнее снизить нагрузку на память. Например возможно полностью загрузить все входные данные и саму прошивку в кэш и больше не обращаться к памяти. К сожалению кэш сильно увеличивает размер процессора и встраивание большого кэша удорожает цену микроконтроллера. Поэтому в большинстве микроконтроллеров кэша нет вообще. Рассмотренных элементов достаточно для функционирования микроконтроллера, но совершенно неясно как микроконтроллер взаимодействует с внешним миром.

Все возможности микроконтроллера по взаимодействию со своим окружением реализуются через специальную область памяти. Она может располагаться где угодно от нулевого адреса, до максимально возможного (например 0xFFFFFFFF). Она может не быть цельной и быть разбита на блоки. **Записью и считыванием в эту область память выполняются все команды вывода и ввода информации в систему.** Для того чтобы узнать какой адрес соответствует конкретному действию нужно читать описание микроконтроллера, который необходимо прошить. Все микроконтроллеры имеют возможность выводить на часть своих контактов цифровые сигналы (1 или 0) и считывать цифровые сигналы. Это позволяет реализовать любое взаимодействие с другой цифровой системой, но для некоторых стандартных взаимодействий придуманы специальные блоки микроконтроллера. Одной из таких систем является таймер. Он позволяет установить некоторую задержку и сообщить, когда эта задержка истечёт (естественно это сообщение появляется в некоторой области памяти отведённой для таймера). Другой важной системой является механизм прерываний. Без прерываний единственным способом отследить появление какого-либо сигнала на контакте микроконтроллера было бы постоянно считывать состояние этого контакта. Прерывания позволяют отвлечь процессор от любых задач для реагирования на пришедший сигнал и безболезненно вернуться к своей работе после реакции. Одной из таких "задач" может быть зависание. Конечно такое зависание есть свидетельство ошибки программиста, но в больших программах избежать всех ошибок невозможно. Поэтому нужен механизм переключения на проверку на зависание и возможный корректный выход из него. Одним из таких механизмов является watchdog таймер. Это система, которая постоянно пытается перезагрузить микроконтроллер и лишь постоянные правильные действия со стороны программы спасают систему от перезагрузки. В случае зависания эти действия прекратятся и система перезагрузится. Зависание обычно имеет худшие последствия чем перезагрузка.

Существуют различные специализированные способы взаимодействия микроконтроллера с окружающей средой – периферия. Микроконтроллер способен передавать данные по различным стандартам, измерять и задавать напряжения, считать количество срабатываний внешних устройств и т.п.

Наконец существуют механизмы отладки прошивки прямо в процессе её работы. Для этого нужно правильно спроектировать плату с микроконтроллером. Возможно знать состояние регистров процессора, содержимое памяти, текущую выполняемую команду. Возможно устанавливать точки останова и выполнять программу пошагово. Помимо отладки прошивки вживую можно отлаживать её не имея самого микроконтроллера, используя симулятор.

## Свойства микроконтроллеров: объём памяти, частота работы, напряжение питания, семейство и разрядность.

Память микроконтроллера для хранения прошивки ограничена. Обычно её хватает с запасом для хранения самой прошивки и связанных с ней данных, но не хватает для хранения привычных пользователям компьютера файлов. Память можно расширить, используя внешние микросхемы или устройства хранения данных. В общем можно подключить к микроконтроллеру хоть винчестер, но в этом случае данные не будут располагаться в общем пространстве памяти, а будут доступны через периферию.

Частота работы является одним из основных параметров быстродействия. Каждая команда процессора выполняется фиксированное число тактов. Частота работы процессора означает количество совершаемых тактов в секунду. Возможно использовать микроконтроллер не на максимальной частоте. Например для энергосбережения, когда высокая производительность не нужна.

Напряжение питания влияет на энергопотребление. Важно учитывать его при соединении микроконтроллера и любого другого устройства. Цифровая логика оперирует цифрами 0 и 1, но в реальности они предстают в виде напряжений 0 и VCC (напряжение питания).

Семейства микроконтроллеров были рассмотрены выше.

Разрядность означает количество бит в регистрах процессора. Или иными словами количество бит в переменных с которыми выполняются операции в процессоре. Разрядность сильно влияет на быстродействие: например если частота 32-разрядного процессора 1 МГц, а 8-разрядного – 10 МГц, то какой из них быстрее сложит два 16-разрядных числа? А два 32-разрядных? Обычно в задачах управления работа с большими числами не нужна и хватает 8-разрядного ядра. Но в вычислительных задачах без 32-разрядных операций пришлось бы очень туго.

## Тестовые платы.

Сам микроконтроллер заработает только если к нему приделать генератор частоты, питание и еще несколько необходимых элементов. Поэтому для изучения микроконтроллеров производитель выпускает специальные платы, которые позволяют легко прошиваться, обладают широкими встроенными и потенциальными возможностями и поддерживаются в программных пакетах. Смотри иллюстрацию на Рис. 1.2.

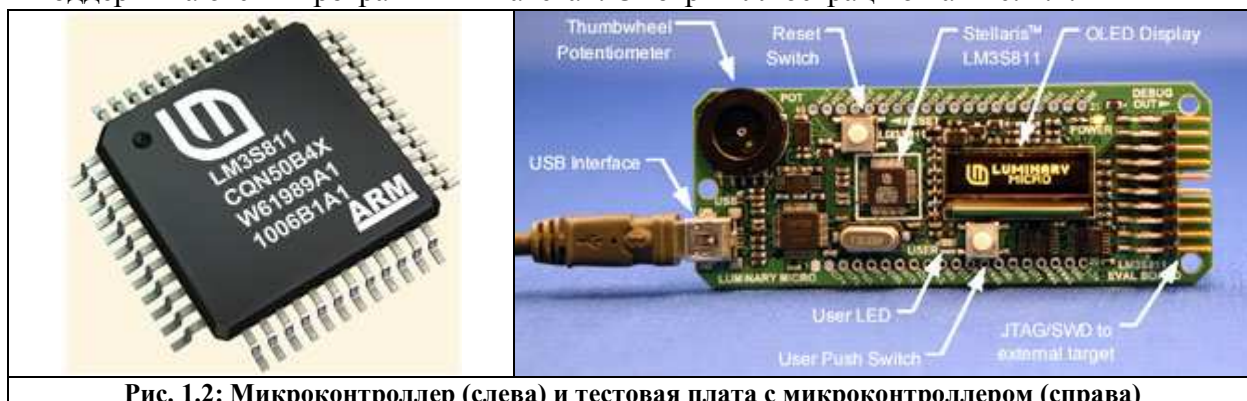


Рис. 1.2: Микроконтроллер (слева) и тестовая плата с микроконтроллером (справа)

## Практическая часть

### Демонстрация процесса работы с тестовой платой

Включение питания. Демонстрация работы дисплея. Показ стартовой игрушки. Демонстрация загрузки прошивки в Keil. Показ пошагового выполнения загруженной прошивки.