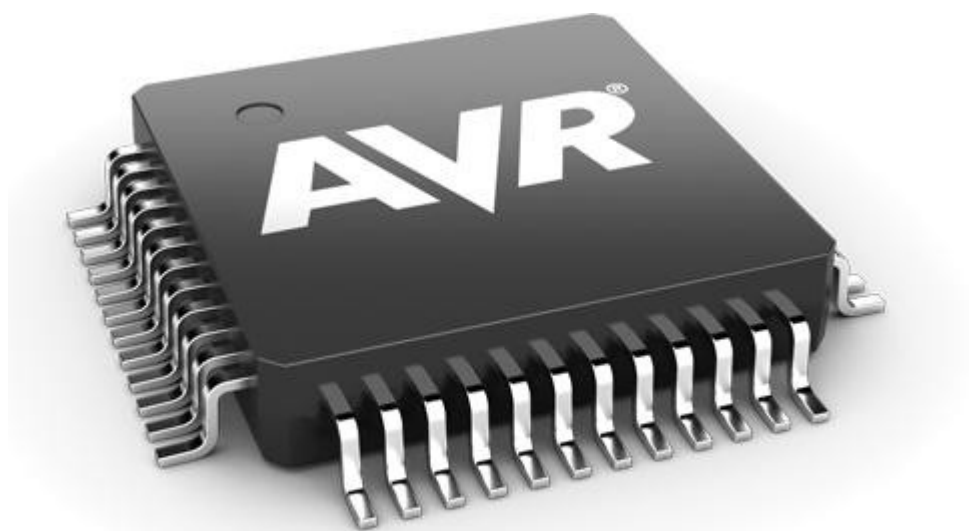


Микроконтроллеры avr для начинающих



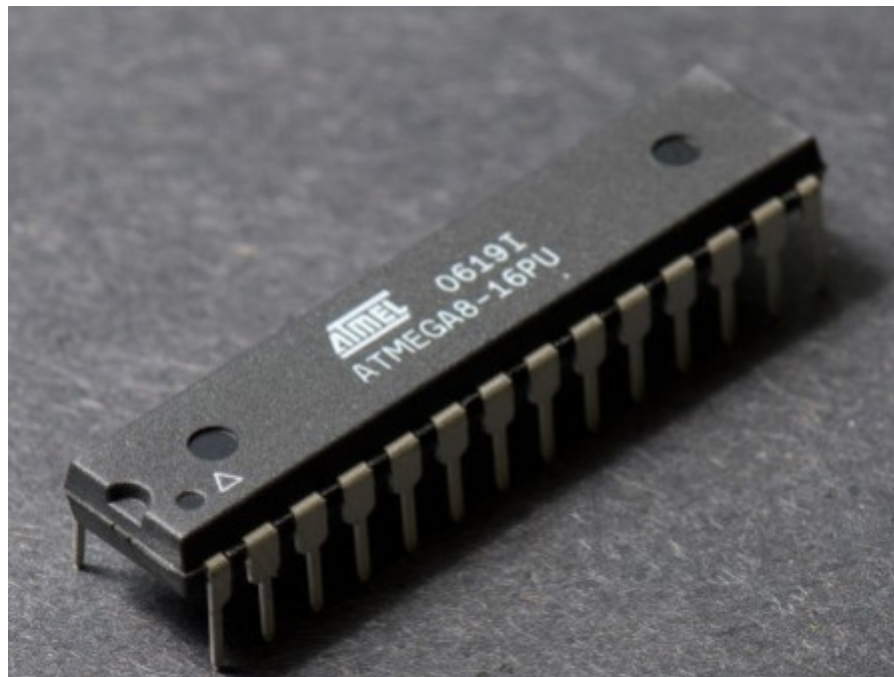
Алексей Бартош

Электрик Инфо – мир электричества
<http://elektrik.info>

Общие сведения об устройстве микроконтроллеров и основные даты.

Микроконтроллеры являются неотъемлемой частью быта современного человека. Применяются от детских игрушек до АСУТП. Благодаря использованию микроконтроллеров, инженерам удалось достигнуть большую скорость изготовления и качество продукции практически во всех сферах производства.

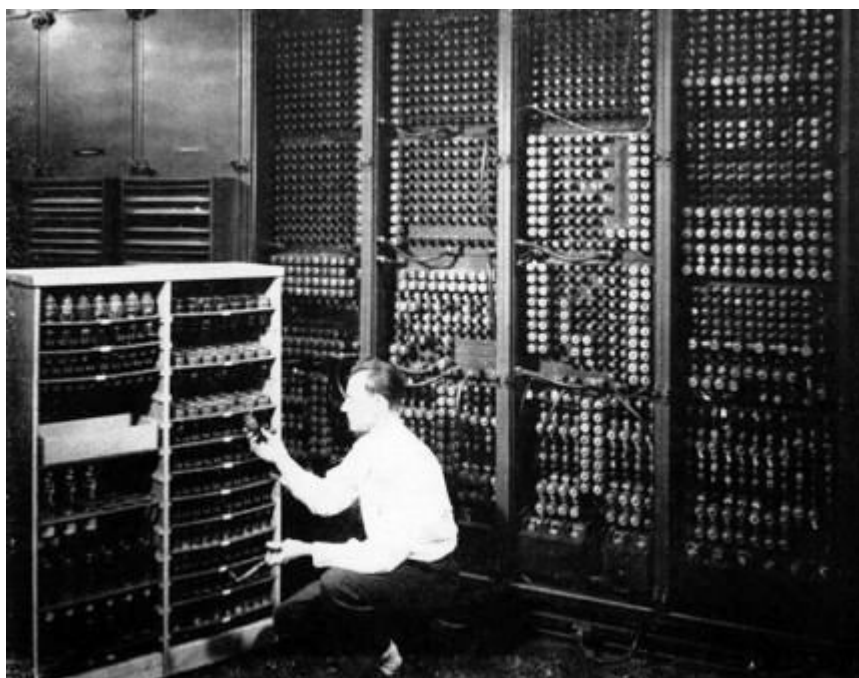
Данный материал является общим обзором ключевых дат в истории развития микроконтроллеров. Это не техническое пособие, многие тонкости и моменты упущены.



Предпосылки для появления микропроцессорных и микроконтроллерных систем.

Чтобы разобраться с причинами появления и развития микропроцессорной техники взгляните на характеристики и особенности первых компьютеров.

ENIAC – первый компьютер, 1946 год. Вес – 30 т, занимал целое помещение или 85 кубических метров объёма в пространстве. Большое тепловыделение, энергопотребление, постоянные неполадки из-за разъемов электронных ламп. Окислы приводили к исчезновению контактов и лампы теряли связь с платой. Требовали постоянного обслуживания.



Компьютерная техника развивалась и к концу 60-х в мире их было порядка 30 тысяч, в их числе как универсальные ЭВМ, так и мини-компьютеры. Мини – того времени были размерами со шкаф.

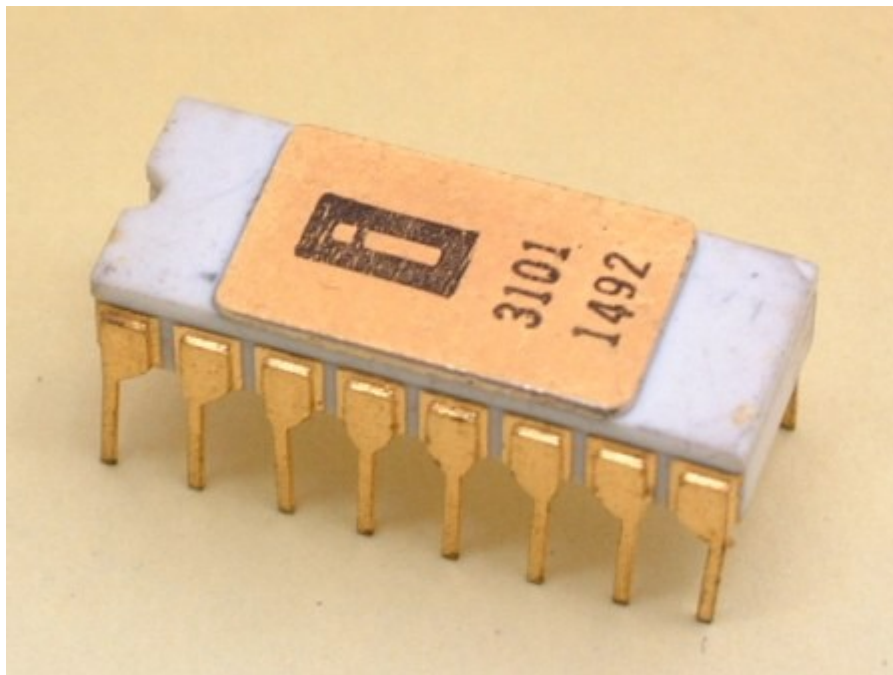
Кстати, в 1969 году уже был изобретен прообраз интернета - ARPANET (англ. Advanced Research Projects Agency Network).

Параллельно развивались полупроводниковые технологии - в 1907 работы по детекторам и электролюминесценции полупроводников. В 1940-е диоды и

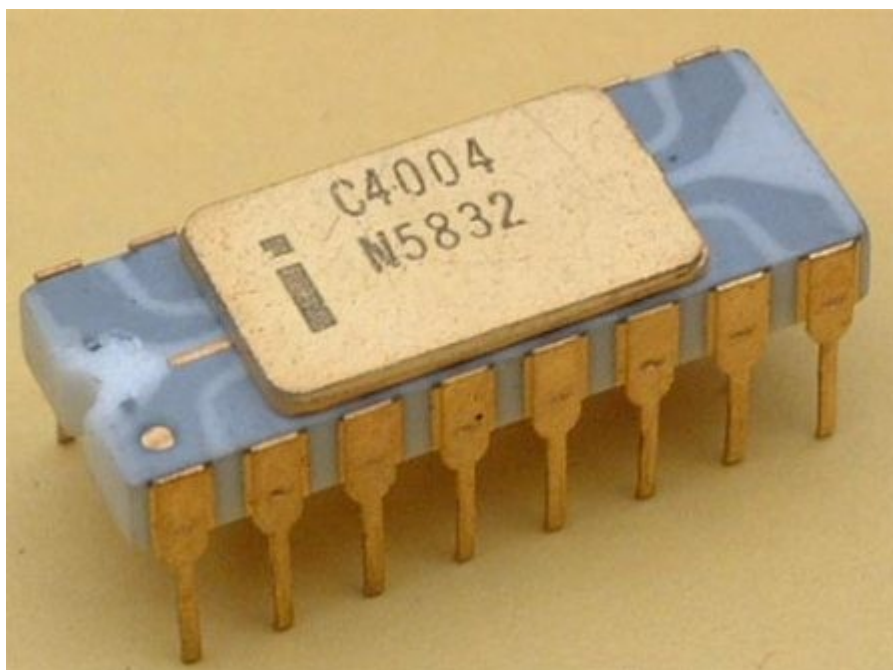
транзисторы. Это всё привело к появлению интегральных технологий. **Роберт Нойс** в 1959 году изобрел интегральную микросхему (далее *ИМС* или *МС*).

Важно:

Фирма Intel – внесла огромный вклад в развитие микроконтроллеров. Основатели: **Роберт Нойс**, Гордон Мур и Эндрю Гроув. Основана в 1968 году.



До определённых пор фирма производила п/п запоминающие устройства. Первым была МС «3101» - 64 разряда, Шотки – биполярная статическая ОЗУ. Следующим было изобретение «4004» - микропроцессора с 2300 п/п транзисторов в своём составе, по производительности не хуже, чем ENIAC, а размером меньше ладони. Т.е. размер 4004-го микропроцессора был на много порядков меньше.



Архитектура, программирование, физическая реализация

Разработчиком архитектуры первого микропроцессора стал – **Тед Хофф**, системы команд – **Стен Мейзор**. **Федерико Феджин** – спроектировал кристалл. Но изначально компания Intel не владела всеми правами на этот чип, и, заплатив 60 000 долларов компании Busicom, получила полные права. Вскоре, последняя обанкротилась.

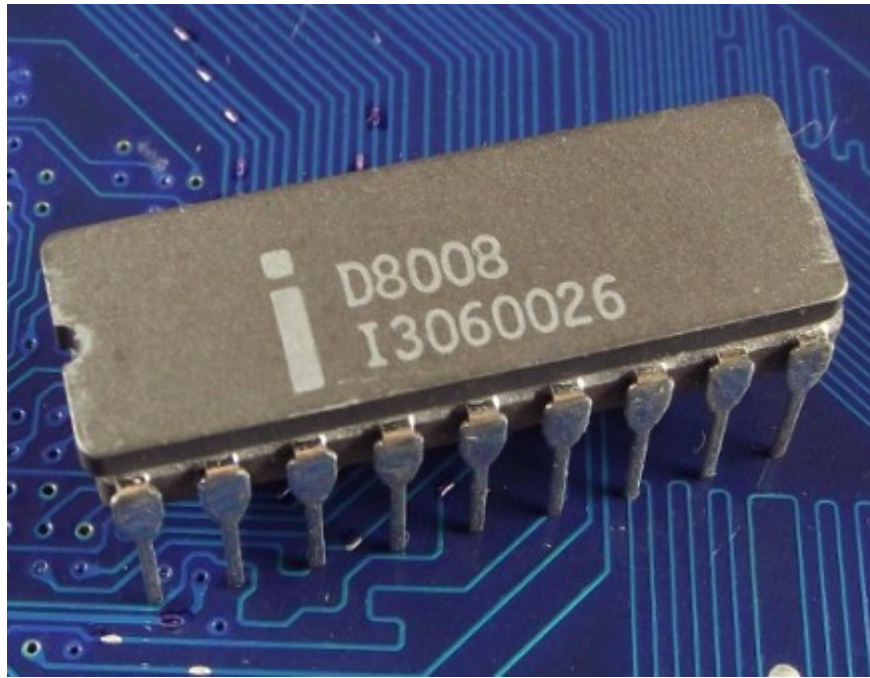
Для популяризации и внедрения новых технологий Intel вела как рекламную, так и образовательную кампанию.

Впоследствии и другие производители электроники объявляли о создании подобных устройств.

Это интересно:

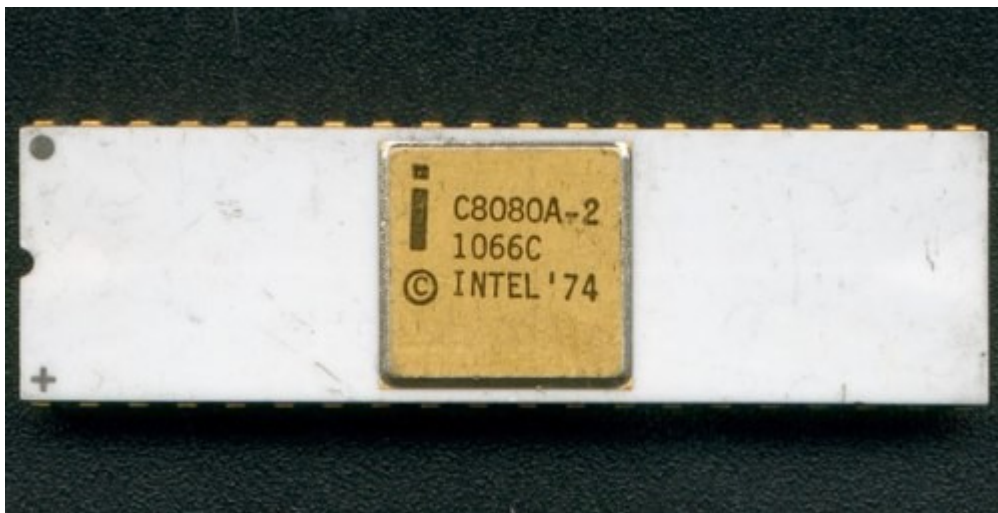
4004 – 4-разрядная, р-МОП микросхема.

Следующим этапом стал выпуск в 1972 году процессора «8008». В отличие от предыдущей модели он уже больше похож на современные модели. 8008 – 8 разрядный, имеет аккумулятор, 6 регистров общего назначения, указатель стека, 8 регистров адреса, команды ввода-вывода.



Событие:

А в 1973 году была изобретена наиболее удачная конфигурация микропроцессора, который до сих пор является классическим – это 8 разрядный «8080».



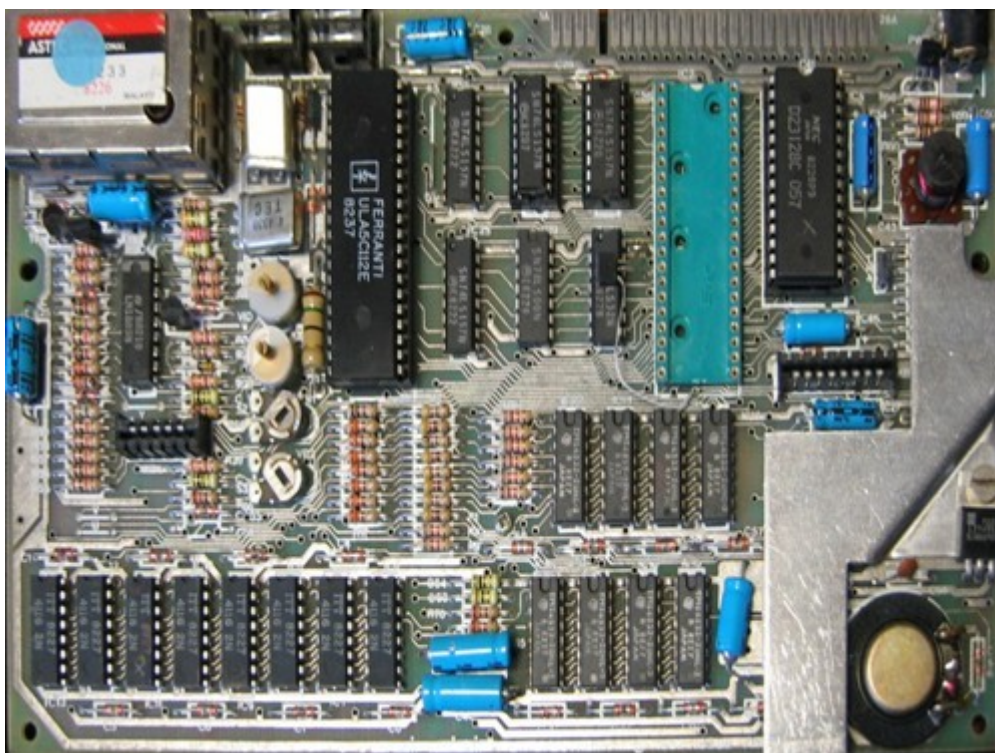
Через полгода у Intel появился серьезный конкурент – Motorola с процессором «6800», n-МОП технология, трёхшинную структуру с 16 разрядной шиной адреса. Более мощная система прерываний, для его питания достаточно одного напряжения, а не три, как у «8080». Кроме того, команды были проще и короче.

До нашего времени сохраняется противостояние семейств микропроцессоров этих производителей.

Ускорило скорость работы и расширило возможности микропроцессоров внедрение 16-разрядных микропроцессоров. Первым из них был разработан «8086» от Intel. Именно его использовали в компании IBM для создания первых персональных компьютеров.

Процессор «68000» - 16 разрядный ответ от «Motorola», использовался в компьютерах ATARI и Apple.

Для широкой аудитории в роли ПК стали популярны **ZX Spectrum**. В них устанавливались процессоры «Z80», от Sinclair Research Ltd. Одна из основных причин популярности – не нужно покупать монитор, ведь спектр, как современные приставки, подключался к телевизору, а обычный магнитофон как устройство для записи и хранения программ и данных.



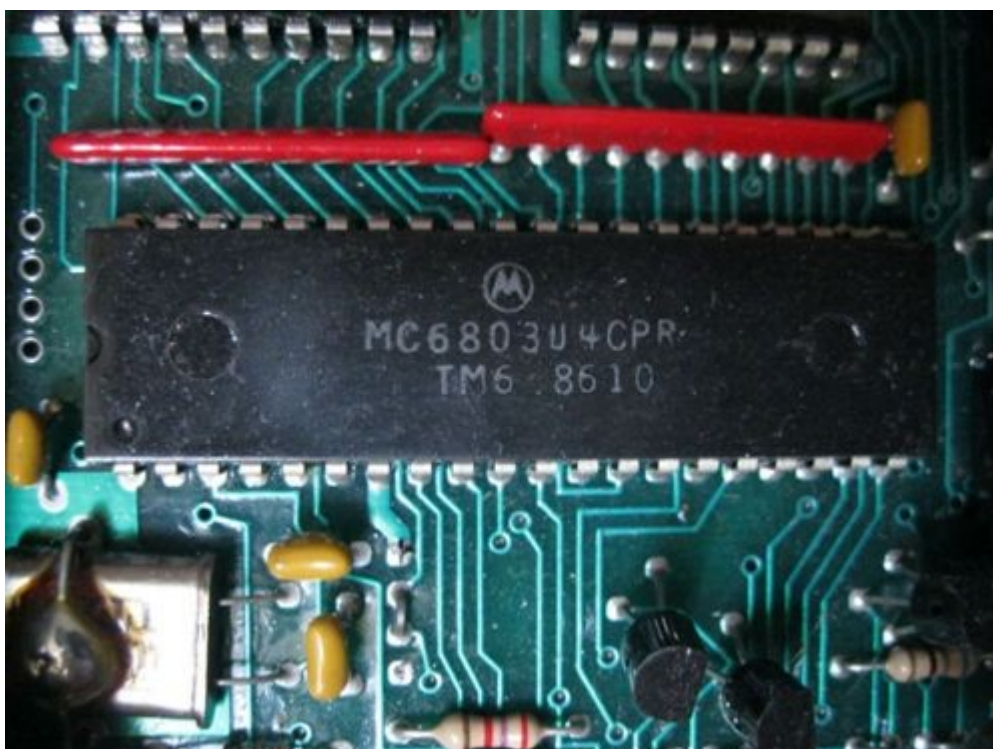
Микроконтроллеры

Микро-ЭВМ – главный шаг массового применения компьютерной автоматизации в области управления. Так как в автоматизации основная задача контроль и регулирование параметров, то термин «контроллер» закрепился и в этой среде.

После перестройки начался активный импорт вычислительной техники, и название «однокристалльная микро-ЭВМ» была вытеснена словом «Микроконтроллер».

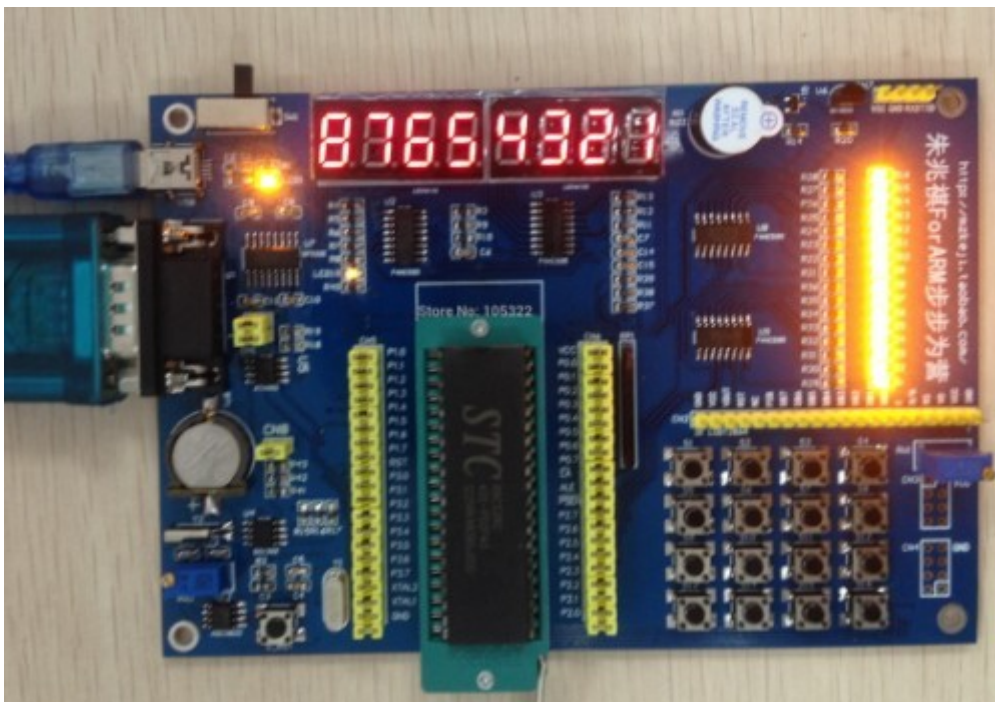
А первый патент в СССР на однокристалльные микро-ЭВМ был выдан в 1971 году М. Кочрену и Г. Буну, из Texas Instruments. С этих пор на кристалле кремния кроме процессора размещали еще память, и дополнительные устройства.

Конец семидесятых – это новая волна конкуренции между Intel и Motorola. Причиной этому стали две презентации, а именно в 76 году Intel выпустила i8048, а Motorola, только в 78 – mc6801, который был совместим с раним микропроцессором mc6800.



Спустя 4 года, к 80 году, Intel выпускает популярны и до сих пор **МК i8051**. Это было зарождение огромного семейства, которое живет и до сих пор. Ведущие мировые производители выпускают на этой архитектуры сильно модифицированные микроконтроллеры для широкого спектра задач.

Для своего времени он имел немислимые 128 000 транзисторов. Это в четыре раза превосходило количество в i8086 процессоре.



В 2017 году, и последние десятилетие наиболее распространены:

- 8-битные микроконтроллеры PIC фирмы Microchip Technology и AVR фирмы Atmel;
- 16-битные MSP430 фирмы TI;
- 32-битные микроконтроллеры, архитектуры ARM. Она продаётся разработчиками различным фирмам, на базе которой выпускается масса различных продуктов.

В Советском союзе техника не стояла на месте. Ученные не только копировали наиболее удачные и интересные зарубежные разработки, но и занимались разработкой уникальных проектов. Таким образом к 1979 году в

НИИ ТТ была разработана K1801BE1, эта микроархитектура называлась «Электроника НЦ» и имела 16 разрядов.



Различия микроконтроллеров

Микроконтроллеры можно разделить по таким критериям:

- Разрядность;
- Система команд;
- Архитектура памяти.

Разрядность – это длина одного слова обрабатываемого контроллером или процессором, чем она больше, тем быстрее микроконтроллер может обработать большие массивы данных, но такой подход не всегда справедлив, для каждой задачи выдвигаются индивидуальные требования, как по скорости, так и по способу обработки, например, применение 32-х разрядного ARM микропроцессор для работы в простых устройствах, оперирующих с 8 битными словами может быть не обосновано как по удобству написания программы и обработки информации, так и по себе стоимость.

Однако, по статистике на 2017 год, стоимость таких контроллеров активно

снижается, и, если так будет продолжаться и далее – он будет дешевле простейших RISC контроллеров, при наличии гораздо большем наборе функций. Не понятно только одно – это маркетинговый ход и занижение цены, или реальный технологический прогресс.

Деление происходит на:

- 8-бит;
- 16-бит;
- 32-бита;
- 64-бита.

Деление по типу системы команд:

- **RISC**-архитектура, или сокращенная система команд. Ориентирована на быстрое выполнение базовых команд за 1, реже 2 машинных цикла, а также имеет большое количество универсальных регистров, и более длинный способ доступа к постоянной памяти. Архитектура характерна для систем под управление UNIX;

- **CISC**-архитектура, или полная система команд, характерна прямая работа с памятью, большее число команд, малое число регистров (ориентирована на работу с памятью), длительность команд от 1 до 4 машинных циклов. Пример – процессоры Intel.

Деление по типу памяти:

- **Архитектура Фон-Неймана** – основная черта общая область памяти для команд и данных, при работе с такой архитектурой в результате ошибки программиста данные могут записаться в область памяти программ и дальнейшее выполнение программы станет невозможным. Пересылка данных и выборка команды не может осуществляться одновременно по тем же причинам. Разработана в 1945 году.

- **Гарвардская архитектура** – отдельная память данных и память программ, использовалась в первые на компьютерах семейства Mark. Разработана в 1944 году.

В результате внедрения микропроцессорных систем размеры устройств снизились, а функционал увеличился. Выбор архитектуры, разрядности, системы команд, структуры памяти – влияет на конечную стоимость устройства, поскольку при единичном производстве разница в цене может быть незначительно, но при тиражировании – более чем ощутимой.

Давайте разберемся с микроконтроллерами AVR: базовые сведения

AVR – это название популярного семейства микроконтроллеров, которое выпускает компания **Atmel**. Кроме AVR под этим брендом выпускаются микроконтроллеры (*далее МК*) и других архитектур, например, ARM и i8051.

Какими бывают AVR микроконтроллеры?



Существует три вида микроконтроллеров:

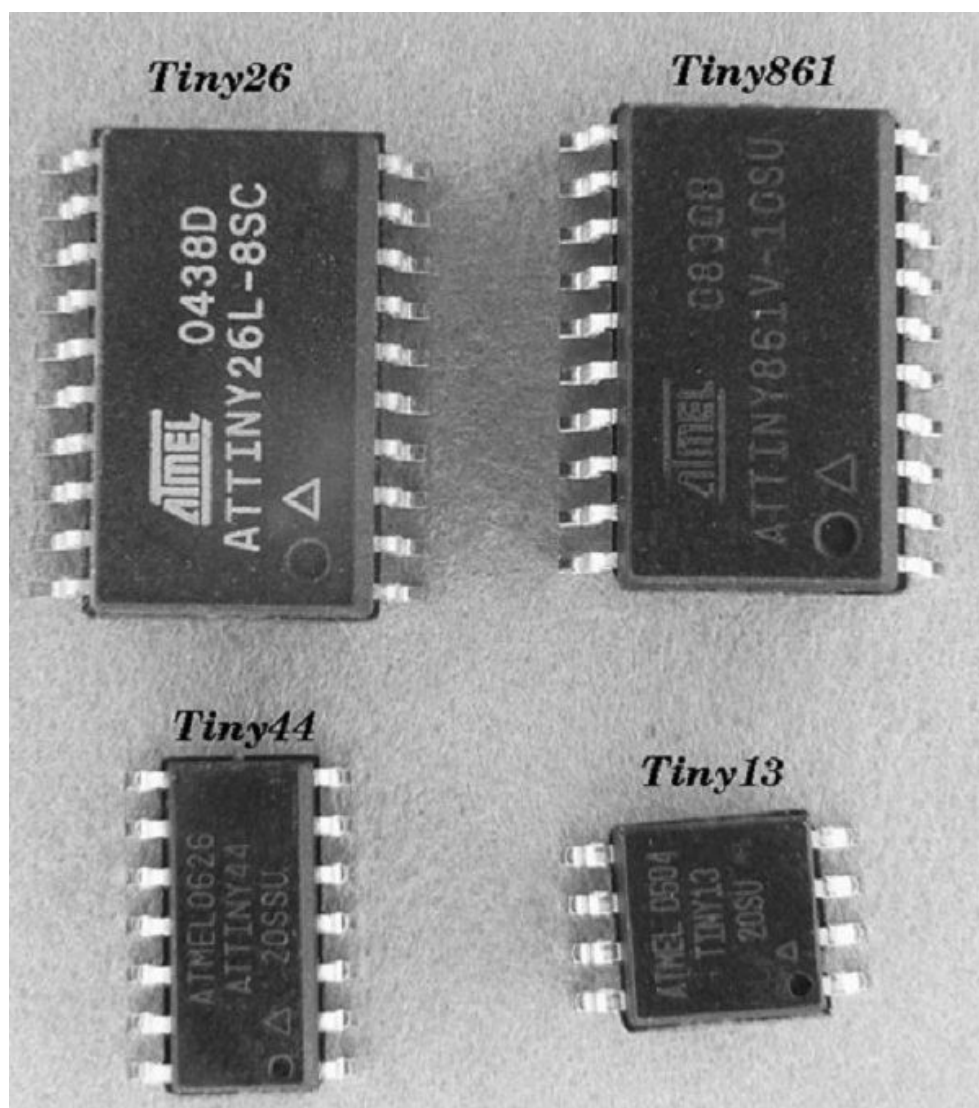
1. **AVR 8-bit.**
2. **AVR 32-bit.**

3. AVR xMega

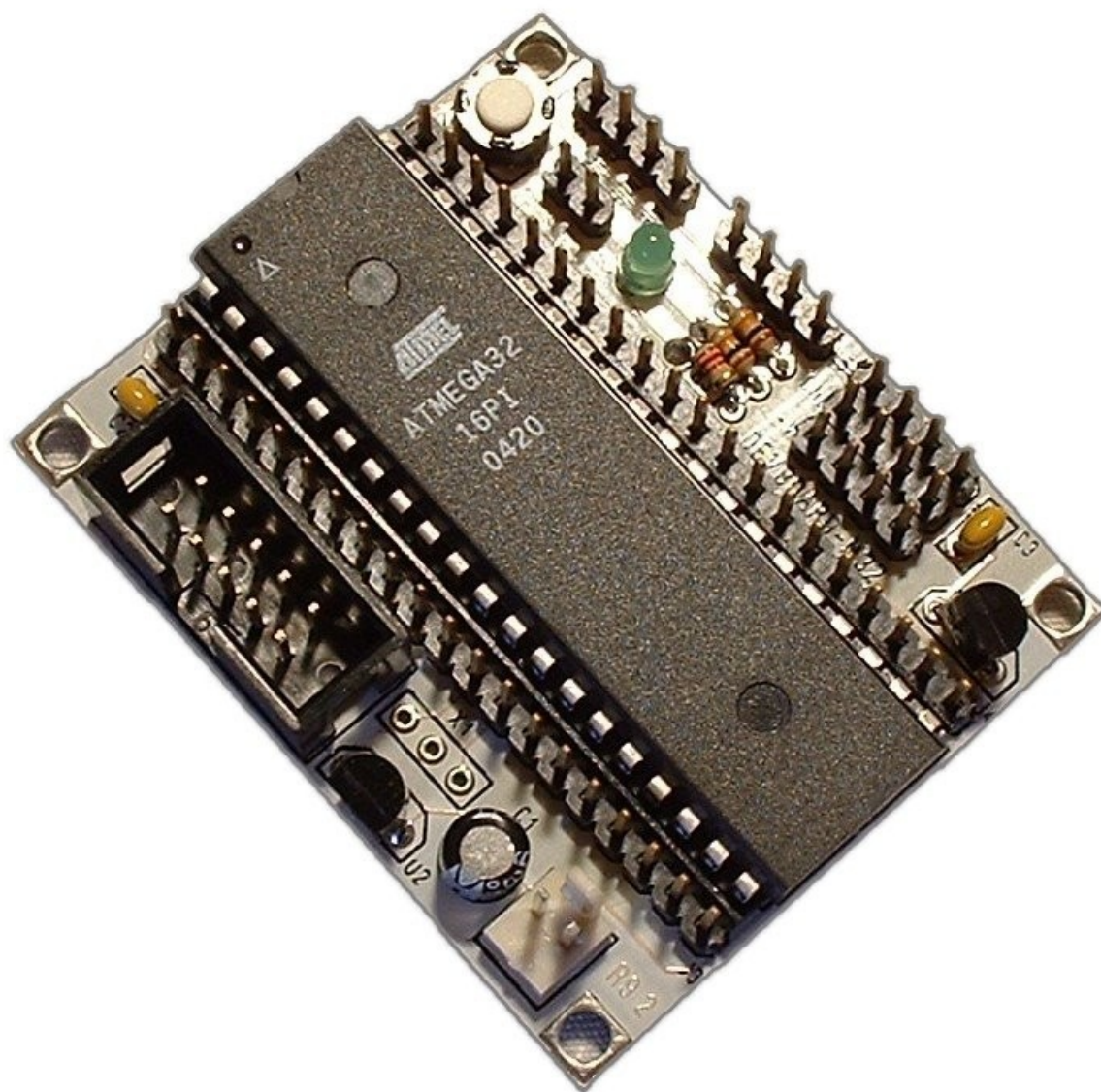
Самым популярным уже более десятка лет является именно 8-битное семейство МК. Многие радиолюбители начинали изучать микроконтроллеры с него. Почти все они познавали мир программируемых контроллеров делая свои простые поделки, вроде светодиодных мигалок, термометров, часов, а также простой автоматики, типа управления освещением и нагревательными приборами.

Микроконтроллеры AVR 8-bit в свою очередь делятся на два популярных семейства:

- **Attiny** – из названия видно, что младшее (tiny – юный, молодой, младший), в основном имеют от 8 пинов и более. Объём их памяти и функционал обычно скромнее, чем в следующем;



- **Atmega** – более продвинутые МК, имеют большее количество памяти, выводов и различных функциональных узлов;



Самым мощным подсемейством микроконтроллеров является xMega – эти МК выпускаются в корпусах с огромным количеством *пинов*, от 44 до 100. Столько необходимо для проектов с большим количеством датчиков и исполнительных механизмов. Кроме того, увеличенный объем памяти и скорость работы позволяют получить высокое быстродействие.

Расшифровка:

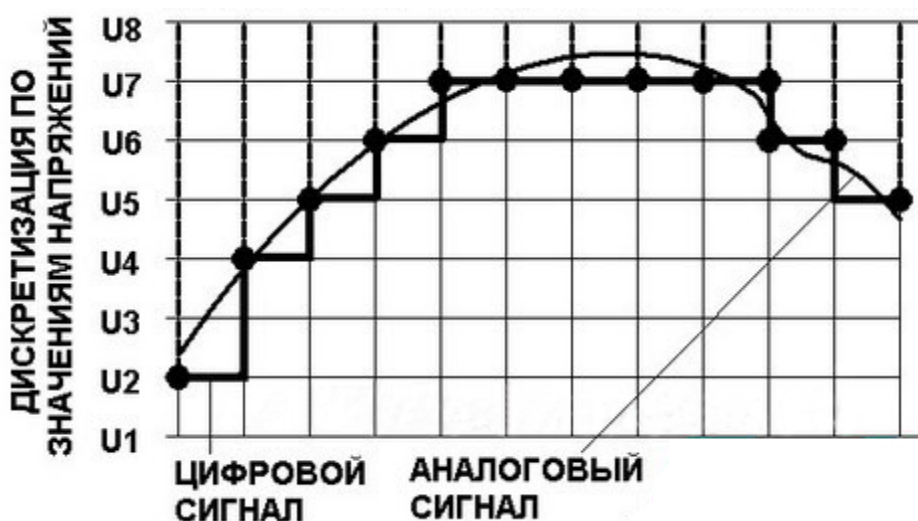
Пин (англ. pin – иголка, булавка) – это вывод микроконтроллера или как говорят в народе – ножка. Отсюда же слово «распиновка» - т.е. информация о назначении каждой из ножек.

Для чего нужны и на что способны микроконтроллеры?

Микроконтроллеры применяются почти везде! Практически каждое устройство в 21 веке работает на микроконтроллере: измерительные приборы, инструменты, бытовая техника, часы, игрушки, музыкальные шкатулки и открытки, а также многое другое; одно лишь перечисление займет несколько страниц текста.

Разработчик может использовать аналоговый сигнал подводя его на вход микроконтроллера и манипулировать с данными о его значении. Эту работу выполняет аналогово-цифровой преобразователь (АЦП). Данная функция позволяет общаться пользователю с микроконтроллером, а также воспринимать различные параметры окружающего мира с помощью датчиков.

В распространенных AVR-микроконтроллерах, например, **Atmega328**, который на 2017 году является сердцем многих плат **Arduino**, но о них позже. Используется 8 канальный АЦП, с разрядностью **10 бит**. Это значит вы сможете считать значение с 8 аналоговых датчиков. А к цифровым выводам подключаются цифровые датчики, что может быть очевидным. Однако цифровой сигнал может являться только 1 (единицей) или 0 (нулем), в то время как аналоговый может принимать бесконечное множество значений.

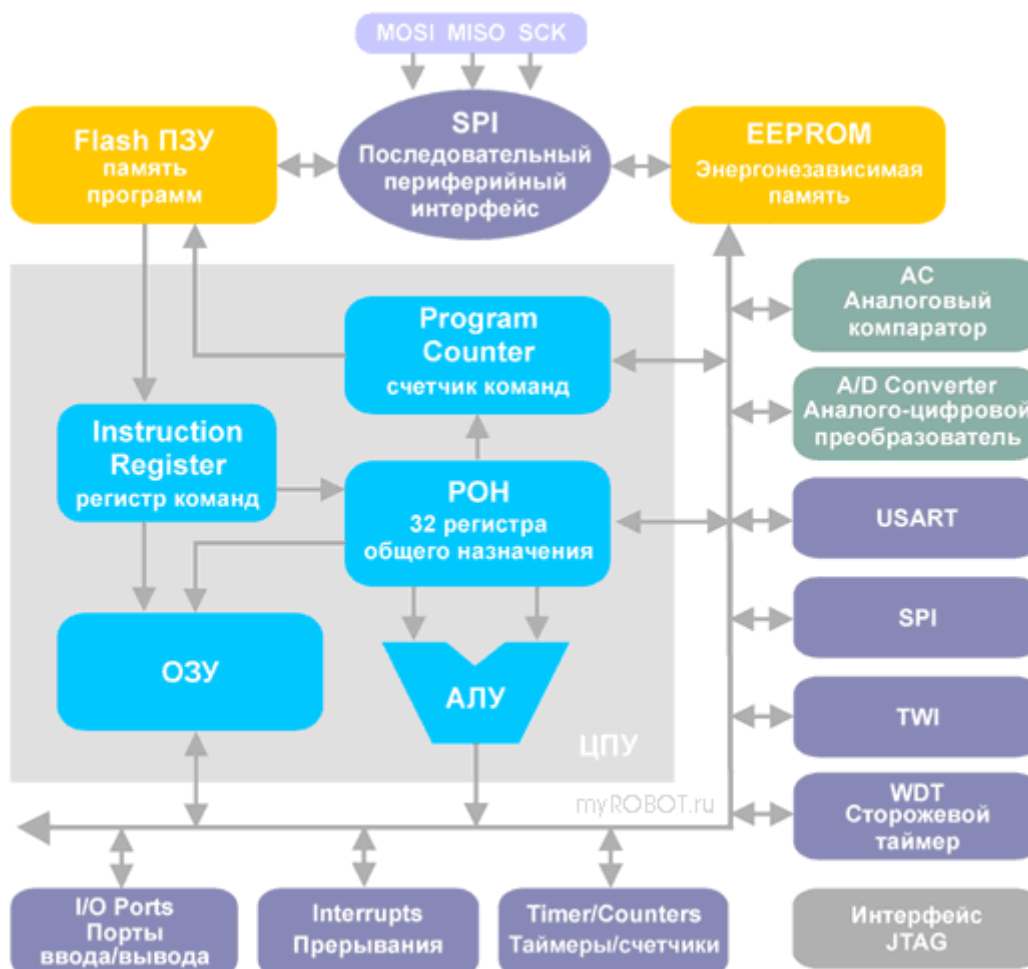


Пояснение:

Разрядность – это величина, которая характеризует качество, точность и чувствительность аналогового входа. Звучит не совсем понятно. Немного практики: 10 битный АЦП, записать аналоговую информацию с порта в 10 битах памяти, иначе говоря плавно изменяющийся цифровой сигнал микроконтроллером распознается как числовое значение от 0 до 1024. 12 битный АЦП видит тот же сигнал, но с более высокой точностью – в виде от 0 до 4096, а это значит, что измеренные значения входного сигнала будут в 4 раза точнее. Чтобы понять откуда взялись 1024 и 4096, просто возведите 2 в степени равную разрядности АЦП (2 в степени 10, для 10 разрядного и т.д.)

Чтобы управлять мощностью нагрузки к вашему распоряжению есть ШИМ-каналы, их можно задействовать, например, для регулировки яркости, температуры, или оборотов двигателя. В том же 328 контроллере их 6.

В общем структура AVR микроконтроллера изображена на схеме:



Все узлы подписаны, но всё же некоторые названия могут быть не столь очевидными. Давайте рассмотрим их обозначения.

- **АЛУ** – арифметико-логическое устройство. Нужно для выполнения вычисления.
- **Регистры общего назначения (РОН)** – регистры которые могут принимать данные и хранить их в то время пока МК подключен к питанию, после перезагрузки стираются. Служат как временные ячейки для операций с данными.
- **Прерывания** – что-то вроде события которое возникает по внутренним или внешним воздействиям на МК – переполнение таймера, внешнее прерывание с пина МК и т.д.
- **JTAG** – интерфейс для внутрисхемного программирования без снятия МК с платы.
- **Flash, ОЗУ, EEPROM** – виды памяти – программ, временных рабочих данных, долгосрочного хранения независимая от подачи питания к МК соответственно порядку в названиях.
- **Таймеры и счетчики** – важнейшие узлы в МК, в некоторых моделях их количество может быть до десятка. Нужны для того, чтобы отчитывать количество тактов, соответственно временные отрезки, а счетчики увеличивают свое значение по какому-либо из событий. Их работа и её режим зависят от программы, однако выполняются эти действия аппаратно, т.е. параллельно основному тексту программы, могут вызвать прерывание (по переполнению таймера, как вариант) на любом этапе выполнения кода, на любой его строке.
- **A/D (Analog/Digital)** – АЦП, его назначение мы уже описали ранее.
- **WatchDogTime (Сторожевой таймер)** – независимый от МК и даже его тактового генератора RC-генератор, который отсчитывает определенный промежуток времени и формирует сигнал сброса МК, если тот работал,

и пробуждения – если тот был в режиме сна (энергосбережения). Его работу можно запретить, установив бит WDTE в 0.

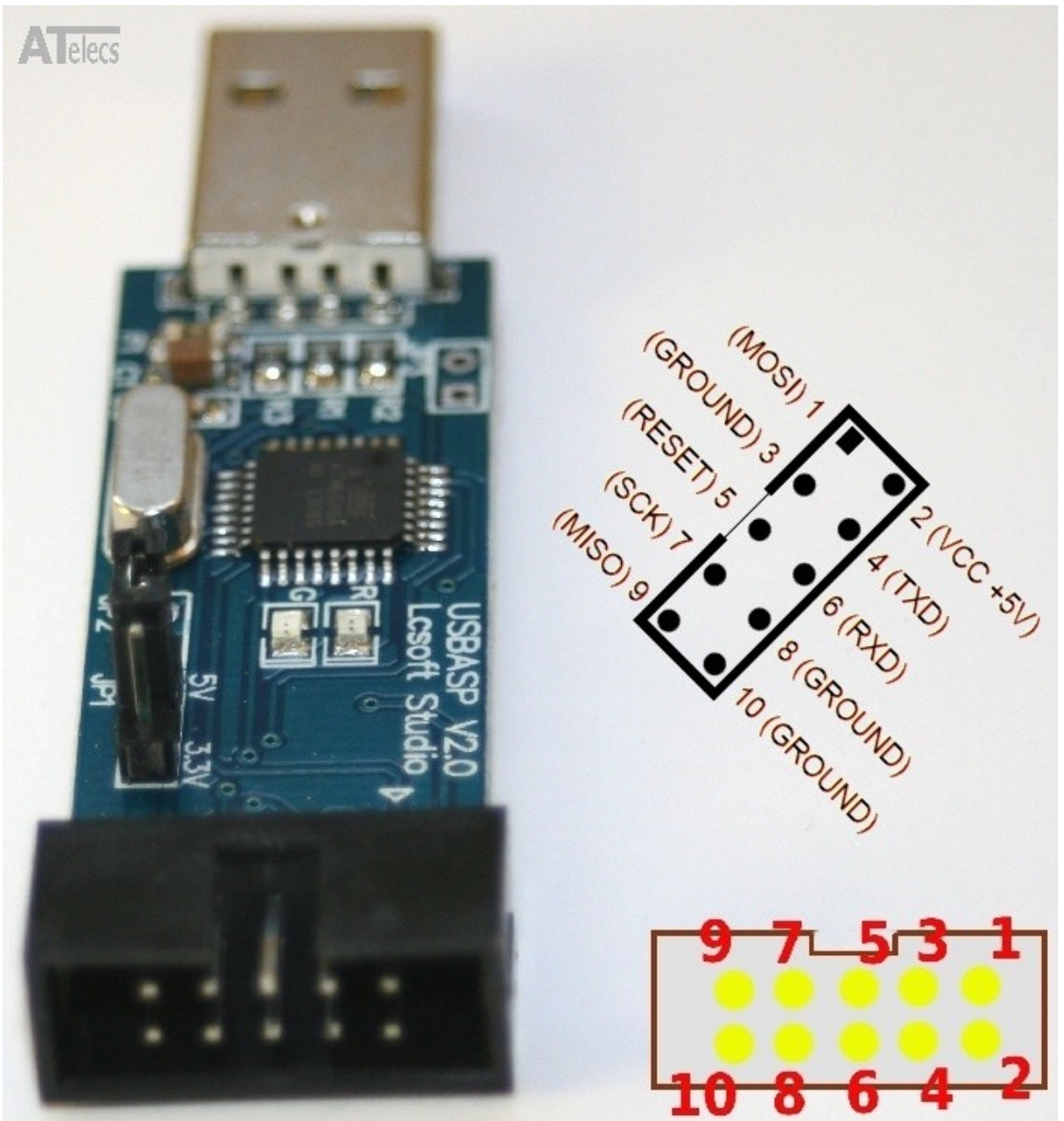
Выходы микроконтроллера довольно слабые, имеется в виду то, что ток через них обычно до 20-40 миллиампер, чего хватит для розжига светодиода и LED-индикаторов. Для более мощной нагрузки – необходимы усилители тока или напряжения, например, те же транзисторы.

Что нужно чтобы начать изучение микроконтроллеров, что такое Arduino?

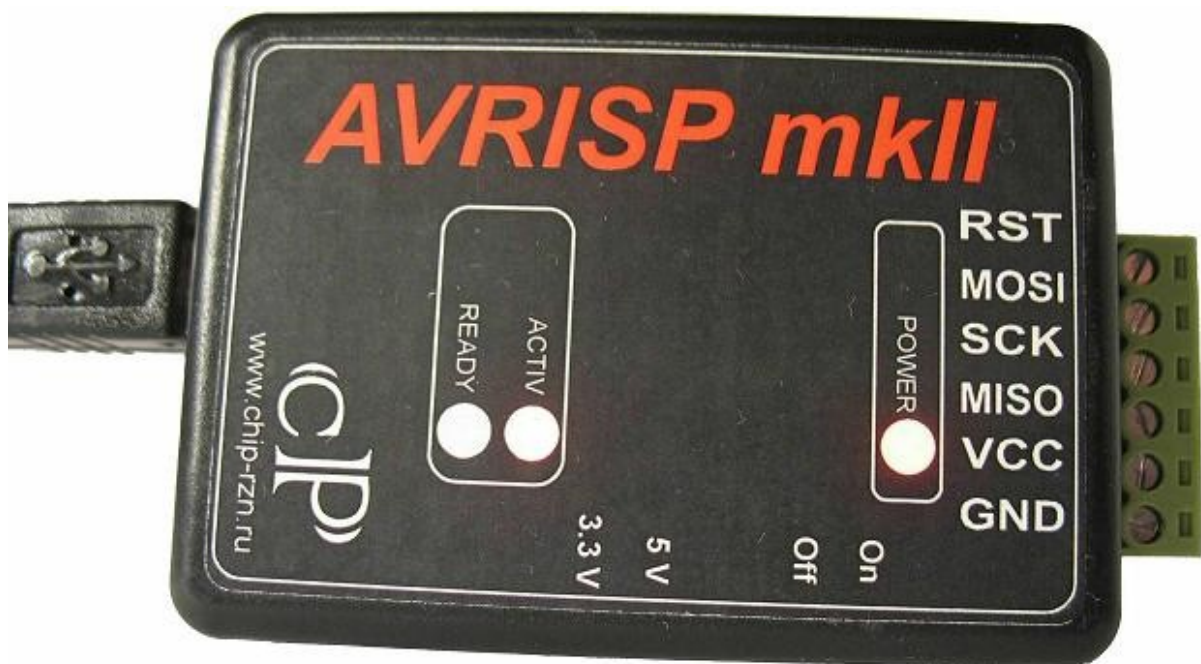
Для начала нужно приобрести сам микроконтроллер. В роли первого микроконтроллера может быть любой Attiny2313, Attiny85, Atmega328 и другие. Лучше выбрать ту модель, которая описана в уроках, по которым вы будете заниматься.



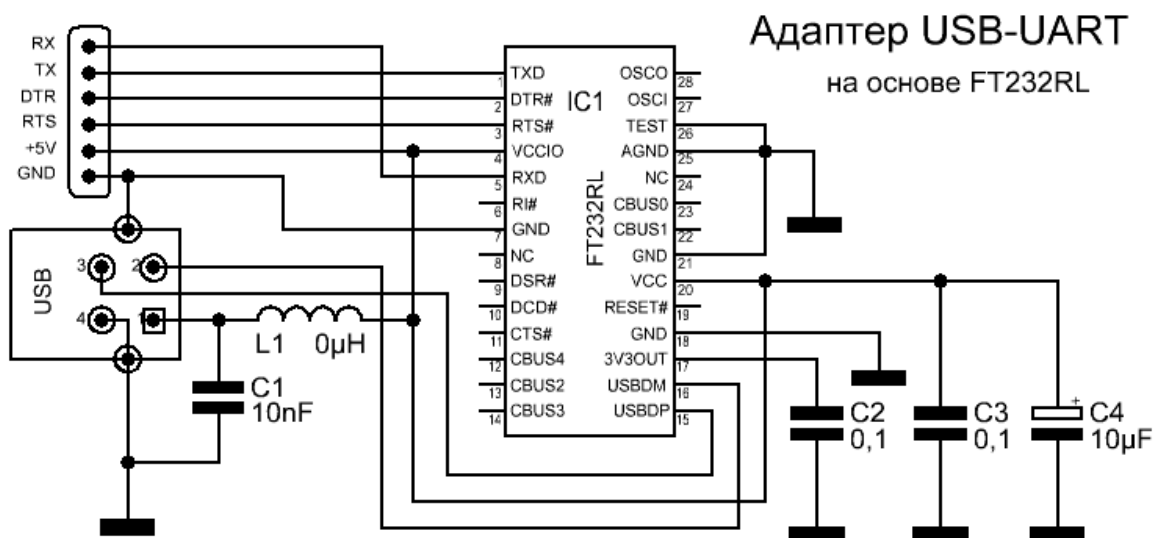
Следующее что Вам нужно – **программатор**. Он нужен для загрузки прошивки в память МК, самым дешевым и популярным считается **USBASP**.



Немногим дороже, но не менее распространенный программатор AVRISP MKII, который можно сделать своими руками – из обычной платы Arduino



Другой вариант – прошивать их через **USB-UART** переходник, который обычно делается на одном из преобразователей: **FT232RL**, **CH340**, **PL2303** и **CP2102**.

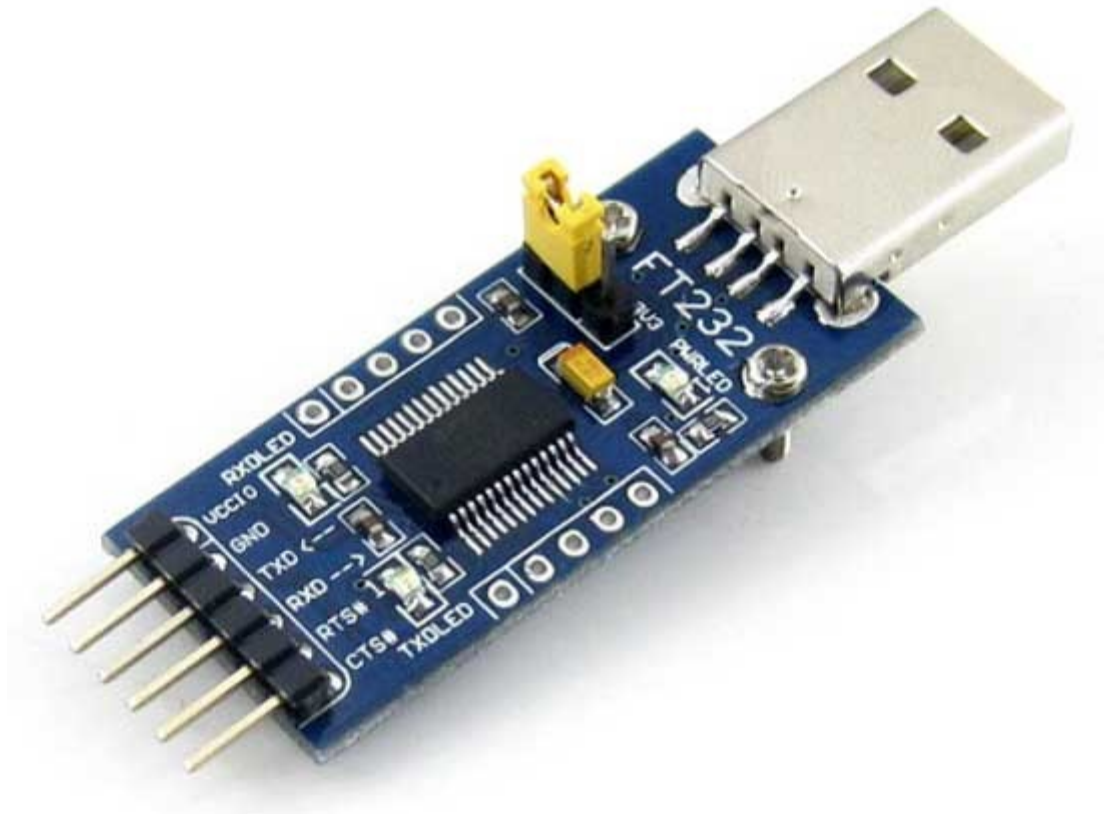


Перемычка +5V USB - подача питания на контроллер с порта USB (нагрузка не более 500мА)
 L1 - ферритовый дроссель с нулевой индуктивностью (защита от помех)

MSEVM.com

В некоторых случаях для такого преобразователя используют микроконтроллеры AVR с аппаратной поддержкой USB, таких моделей не слишком много. Вот некоторые:

- ATmega8U2;
- ATmega16U2;
- ATmega32U2;



Одно лишь «но» – в память МК предварительно нужно загрузить **UART бутлоадер**, который можно скачать отсюда (https://www.chip45.com/categories/chip45boot2_avr_atmega_xmega_uart_bootloader_frei_kostenlos_hexfiles.php). Разумеется, для этого все равно **нужен программатор для AVR-микроконтроллеров**.

Интересно:

***Bootloader** – это обычная программа для МК, только с необычной задачей – после запуска микроконтроллера (подключения к питания) он ожидает какое-то время, что в него могут загрузить прошивку. Преимуществом такого метода – можно прошить любым USB-UART переходником, а они очень дешевы. Недостаток – долго загружается прошивка.*

Для работы **UART (RS-232)** интерфейса в МК AVR выделен целый регистр **UDR (UART data register)**. **UCSRA** (настройки битов приемопередатчика RX, TX), **UCSRB** и **UCSRC** – набор регистров отвечающие за настройки интерфейса в целом.

В чем можно писать программы?

Кроме программатора для написания и загрузки программы нужно IDE – среда для разработки. Можно конечно же писать код в блокноте, пропускать через компиляторы и т.д. Зачем это нужно, когда есть отличные готовые варианты. Пожалуй, один из наиболее сильных – это IAR, однако он платный.

Официальным IDE от Atmel является AVR Studio, которая на 6 версии была переименована в Atmel studio. Она поддерживает все микроконтроллеры AVR (8, 32, xMega), автоматически определяет команды и помогает ввести, подсвечивает правильный синтаксис и многое другое. С её же помощью можно прошивать МК.

Наиболее распространённым является - С AVR, поэтому найдите самоучитель по нему, есть масса русскоязычных вариантов, а один из них - *Хартов В.Я. «Микроконтроллеры AVR. Практикум для начинающих».*

Пошаговое обучение программированию и созданию устройств на микроконтроллерах AVR

1. [Программирование микроконтроллеров для начинающих](#)

Курс для тех, кто уже знаком с основами электроники и программирования, кто знает базовые электронные компоненты, собирает простые схемы, умеет держать паяльник и желает перейти на качественно новый уровень, но постоянно откладывает этот переход из-за сложностей в освоении нового материала.

Курс замечательно подойдет и тем, кто только недавно предпринял первые попытки изучить программирование микроконтроллеров, но уже готов все бросить от того, что у него ничего не работает или работает, но не так как ему нужно (знакомо?!).

Курс будет полезен и тем, кто уже собирает простенькие (а может и не очень) схемы на микроконтроллерах, но плохо понимает суть того как микроконтроллер работает и как взаимодействует с внешними устройствами.

2. Программирование микроконтроллеров на языке Си

Курс посвящен обучению программирования микроконтроллеров на языке Си. Отличительная особенность курса - изучение языка на очень глубоком уровне. Обучение происходит на примере микроконтроллеров AVR. Но, в принципе, подойдет и для тех, кто использует другие микроконтроллеры.

Курс рассчитан на подготовленного слушателя. То есть, в курсе не рассматриваются базовые основы информатики и электроники и микроконтроллеров. Но, что бы освоить курс понадобятся минимальные знания по программированию микроконтроллеров AVR на любом языке. Знания электроники желательны, но не обязательны.

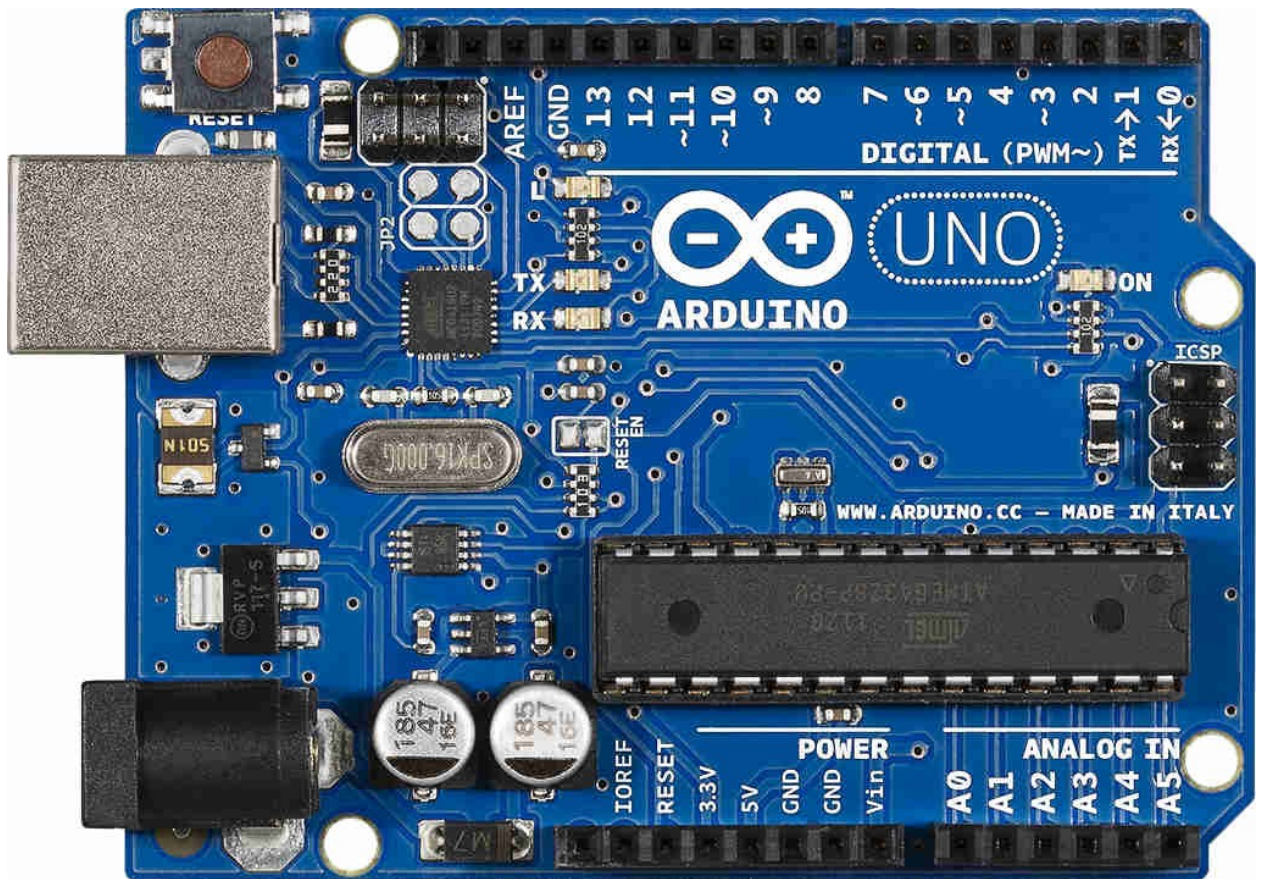
Курс идеально подойдет тем, кто только начал изучать программирование AVR микроконтроллеров на языке C и хочет углубить свои знания. Хорошо подойдет и тем, кто немного умеет программировать микроконтроллеры на других языках. И еще подойдет обычным программистам, которые хотят углубить знания в языке Си.

3. Создание устройств на микроконтроллерах на языке Си

Этот курс для тех, кто не хочет ограничиваться в своем развитии простыми или готовыми примерами. Курс отлично подойдет тем, кому важно создание интересных устройств с полным пониманием того, как они работают. Курс хорошо подойдет и тем, кто уже знаком с программированием микроконтроллеров на языке Си и тем, кто уже давно программирует их.

Самый простой способ изучить AVR

Купите или сделайте своими руками плату Arduino. Проект ардуино разработан специально для учебных целей. Он насчитывает десятки плат различных формами и количеством контактов. Самое главное в ардуино – это то что вы покупаете не просто микроконтроллера, а полноценную отладочную плату, распаянную на качественной текстолитовой печатной плате, покрытой маской и смонтированными SMD компонентами. Самые распространенные – это Arduino Nano и Arduino UNO, они по сути своей идентичны, разве что «Нано» меньше примерно в 3 раза чем «Уно».



Несколько фактов:

- Ардуино может программироваться стандартным языком – «C AVR»;
- своим собственным – wiring;
- стандартная среда для разработки – Arduino IDE;
- для соединения с компьютером достаточно лишь подключить USB шнур к гнезду micro-USB на плате ардуино нано, установить драйвера (скорее всего это произойдет автоматически, кроме случаев, когда преобразователь на CH340, у меня на Win 8.1 драйвера не стали, пришлось скачивать, но это не заняло много времени.) после чего можно заливать ваши «скетчи»;
- «Скетчи» – это название программ для ардуино;

Подробнее про Arduino:

[Подключение и программирование Ардуино для начинающих](#)

Выводы

Микроконтроллеры станут отличным подспорьем в вашей радиолюбительской практике, что позволит вам открыть для себя мир цифровой электроники, конструировать свои измерительные приборы и средства бытовой автоматики.