

# Программирование семейства микроконтроллеров Atmel

На примере микропроцессорной платы  
Arduino Mega

Автор курса, лектор: Колкер Алексей  
Борисович  
Доцент кафедры автоматки  
НГТУ

[a.kolker@corp.nstu.ru](mailto:a.kolker@corp.nstu.ru)  
2018

# Благодарности

Идеи устройств и некоторые электронные схемы, которые приводятся в качестве примеров в рамках данного курса, были найдены на просторах интернета и, по возможности, снабжены ссылками на сайты-источники. Если где-то таких ссылок нет, или ссылки ведут не на первоисточник-это случайное недоразумение, надеюсь на понимание авторов, все данные будут незамедлительно скорректированы.

# Цель курса

- Научиться программировать микроконтроллеры AVR Atmel
- Научиться создавать устройства на базе микроконтроллеров
- Научиться строить системы, состоящие из различных интеллектуальных модулей

# ЛР

Что делаем:

- Hello-world blinker.
- Выключатель освещения с плавной регулировкой и подавлением.
- 1wire контроль доступа
- Термометр с выводом информации на текстовый дисплей.
- Работа с графикой и графический дисплей.
- 433МГц — приемопередатчик, исследование.
- W5100 и Ethernet.
- AT Tiny 13 — hello world blinker / пищалка.
- AT Tiny 13 — брелок 433МГц — управление исполнительными устройствами.

С помощью чего делаем:

- ATMEGA2560 в составе Arduino Mega 2560
- Си в обертки Wiring, где предпочтение на стороне теории алгоритмов и связей
- ASM, GCC без обертки Wiring, где предпочтение на стороне реализации

# РГР (следующий семестр)

- В процессе курса будет предусмотрена РГР, которая сопрягается с курсом теории сигналов (Чикильдин Г.П.)
- Задача: реализовать цифровую фильтрацию сигнала по заданным характеристикам на микроконтроллере AVR (серия AT Mega) согласно его функциональным возможностям и данным варианта заданий.

# Платформа для ЛР и КП

- Arduino mega 2560 (AT MEGA 2560)



# Заблуждение

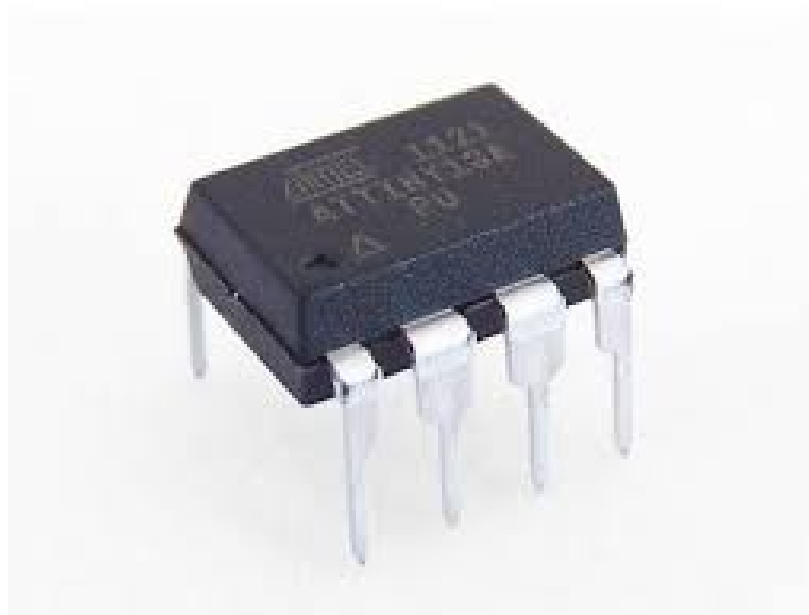
- Ардуино это такая игрушка, которой играют в школе на уроке робототехники, но мы же учимся в университете?
- Ответ: Doom(World of Tanks- кому как больше нравится) тоже запускают на компьютере с CPU. Давайте запишем все PC компьютеры в игрушки?

Резюме: это инструмент, и то как вы им воспользуетесь зависит от исключительно от количества «интеллекта»

- Сердце платы — совершенно обыкновенный микроконтроллер серии AVR, однако от в отличие профессиональных программаторов, ее можно купить за сумму в несколько долларов и создавать с помощью нее ( но не на базе нее- об этом позже) устройства, программируя их на ASM,C



# AT Tiny13 DIP



# TFT Touch 2.4'



# Ethernet Sheild



# LCD 1602



# Сенсоры и трансиверы

T



IR



433MHz



RS 485



# Краткая история линейки AVR

Идея разработки нового прогрессивного RISC-ядра зародилась в норвежском городе Тронхейм (Trondheim) в светлых головах двух студентов Norwegian University of Science and Technology (NTNU) Альф Боген (Alf-Egil Bogen) и Вегард Воллен (Vegard Wollen).

Будущие директора Atmel Norway создали архитектуру, которая стала одной из самых удачных на мировом рынке микроконтроллеров.



В 1995 году Боген и Воллен решили предложить американской корпорации Atmel, известной на тот момент своим "ноу-хау" изготовления чипов с Flash-памятью, выпускать новый 8-битный RISC-микроконтроллер и снабдить его Flash-памятью программ на кристалле. Идея настолько понравилась руководству Atmel Corp., что было принято решение незамедлительно инвестировать данный проект.

В 1996 году был основан исследовательский центр Atmel в Тронхейме. Стоит сказать, что 150-тысячный Тронхейм усилиями своего университета каждый год порождает до 20-ти новых компаний, специализирующихся в секторах рынка начиная от автоматизации и до передачи и обработки данных. В конце 1996 года был выпущен опытный кристалл AT90S1200, а во второй половине 1997-го корпорация Atmel приступила к серийному производству нового семейства микроконтроллеров, к их рекламной и технической поддержке.



Новое ядро было запатентовано и получило название AVR, которое по прошествии уже нескольких лет стало трактоваться самыми различными способами. Кто-то утверждает, что это не иначе как **Advanced Virtual RISC**, другие полагают, что не обошлось здесь без **Alf Egil Bogen Vegard Wollan RISC**.

Держателями патента при этом являются:  
Wollan, Vegard (NO); Bogen, Alf-Egil (NO);  
Myklebust, Gaute (NO); Bryant, John, D. (US).

Интересно, что система команд и внутреннее устройство чипов AVR разрабатывались совместно с фирмой IAR Systems - производителем компиляторов языков программирования C/C++, что обеспечило уникальные характеристики этих микроконтроллеров. В результате для AVR стало возможным получать высокую плотность кода при использовании языков высокого уровня, практически не теряя в производительности по сравнению с программами, написанными на низкоуровневом языке Ассемблера.

Использование прогрессивной технологии конвейеризации у AVR сокращало цикл "выборка - исполнение" команды. Например, у микроконтроллеров семейства x51 короткая команда выполняется за 12 тактов генератора.

В PIC-контроллерах фирмы Microchip(с конвейером), короткая команда выполняется за 4 периода тактовой частоты.

В микроконтроллерах AVR короткая команда в общем потоке выполнялась всего за один период тактирующего сигнала.

Такое построение кристалла обеспечило существенное повышение производительности, которая в пределе может достигать значения 1MIPS на 1МГц. Это во многих случаях при заданной производительности позволяло снизить тактовую частоту, а значит, и потребляемую мощность устройства. AVR-микроконтроллеры предоставляли более широкие возможности по оптимизации производительности/энергопотребления, что было особенно важно при разработке приложений с батарейным питанием.

# Линейка AT

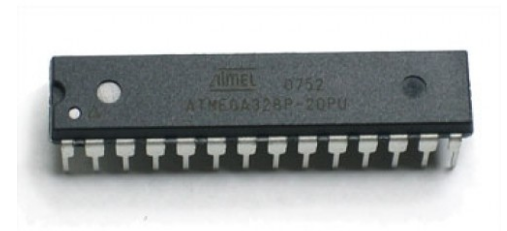
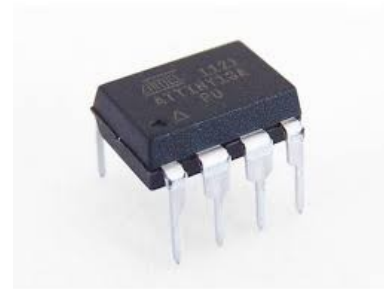
Сегодня Atmel выпускает три семейства микроконтроллеров с ядром AVR:

AT Tiny (Flash <16Kb, SRAM < 512b , EEPROM <512b)

- I/O 3-18 ( 6-32 Pin)
- Нет умножения в АЛУ

AT Mega (Flash <256K,SRAM <16Kb EEPROM)

- I/O 23-86 (26-100Pin)
- Аппаратное умножение АЛУ
- Несколько UART
- Расширенная система команд

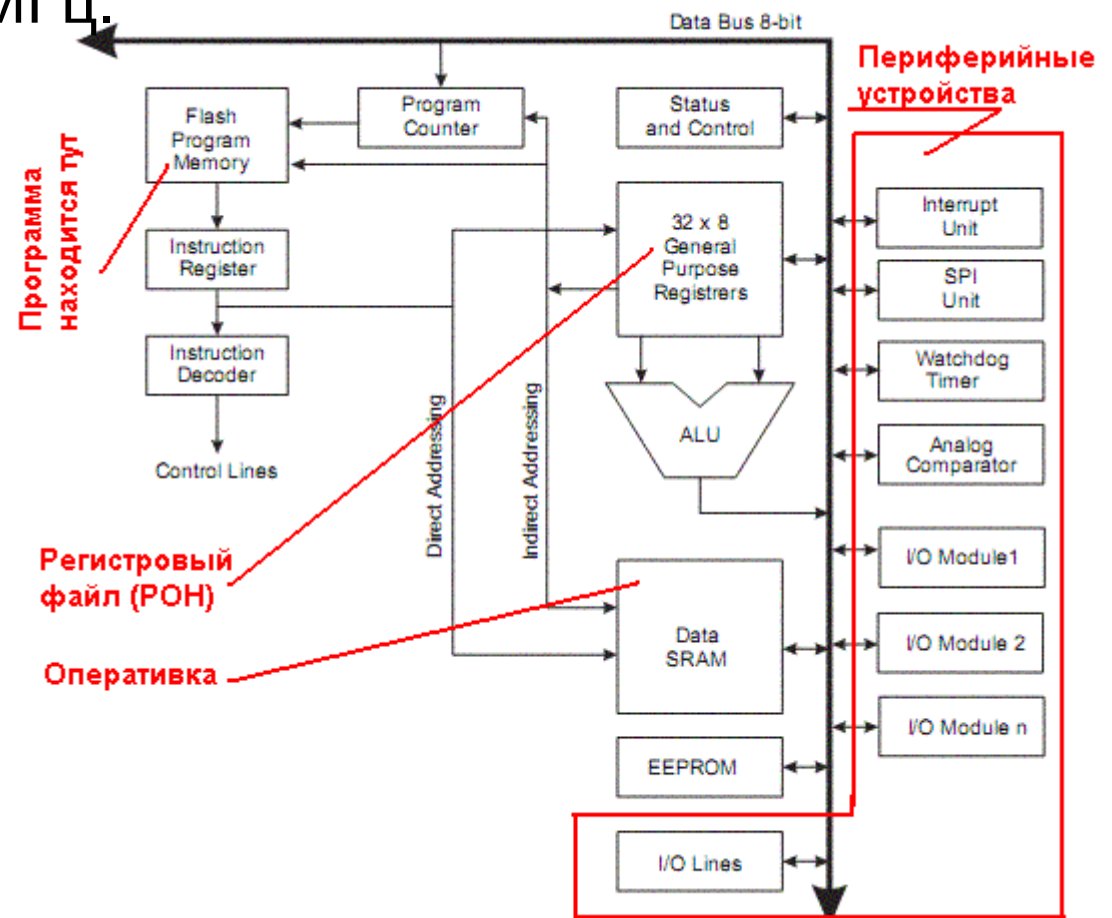


# Линейка ATxMega

- Flash до 384 Кб, SRAM до 32 К; EEPROM до 4К
- 4х канальный DMA
- Расширенная система команд



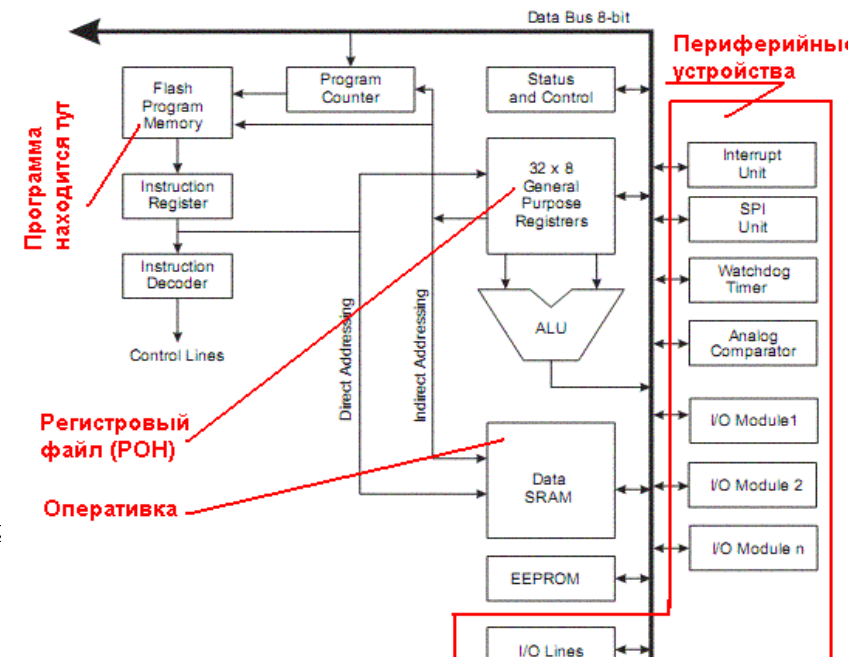
Сердцем микроконтроллеров AVR является 8-битное микропроцессорное ядро или центральное процессорное устройство (ЦПУ), построенное на принципах RISC-архитектуры. Основой этого блока служит арифметико-логическое устройство (АЛУ). По системному тактовому сигналу из памяти программ в соответствии с содержимым счетчика команд (Program Counter - PC) выбирается очередная команда и выполняется АЛУ. Во время выбора команды из памяти программ происходит выполнение предыдущей выбранной команды, что и позволяет достичь быстродействия 1 MIPS на 1 МГц.



# Что такое RISC архитектура?

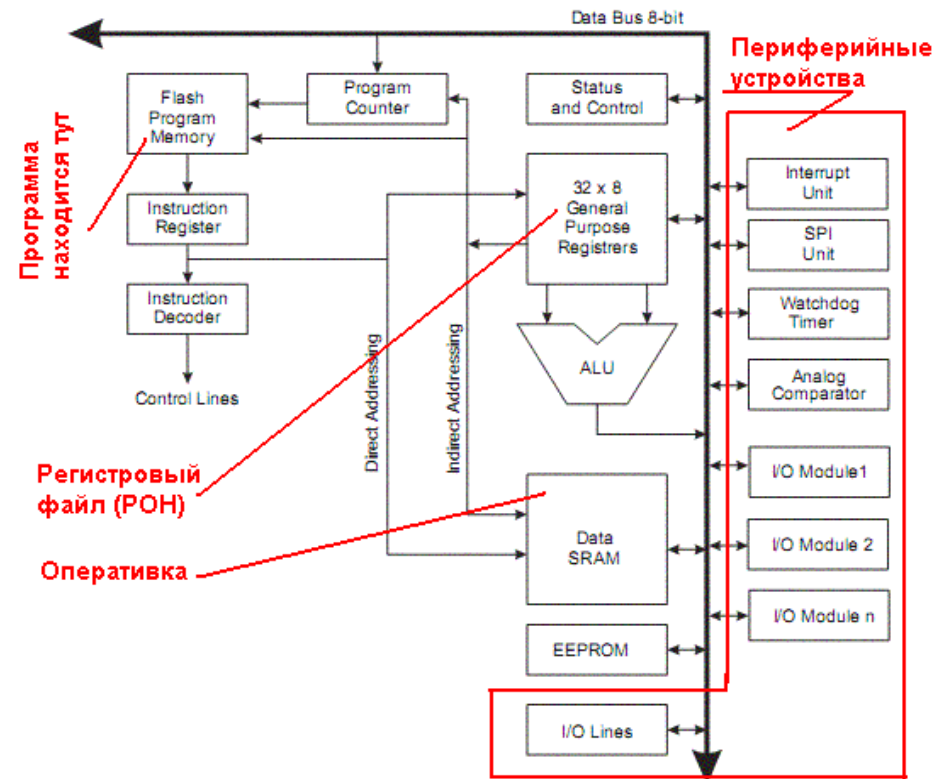
- Архитектура максимально упрощающая команды, обычно:
  - фиксированная длина инструкций;
  - доступ в память только чтение или запись;
  - регистры общего назначения ( в МК отвечают еще и за периферию);
  - отсутствие микропрограмм ( серий инструкций) внутри процессора.

АЛУ подключено к регистрам общего назначения РОН (General Purpose Registers - GPR). Регистров общего назначения всего 32, они имеют байтовый формат, то есть каждый из них состоит из восьми бит. РОН находятся в начале адресного пространства оперативной памяти, но физически не являются ее частью. Поэтому к ним можно обращаться двумя способами (как к регистрам и как к памяти). Такое решение является особенностью AVR и повышает эффективность работы и производительность микроконтроллера.





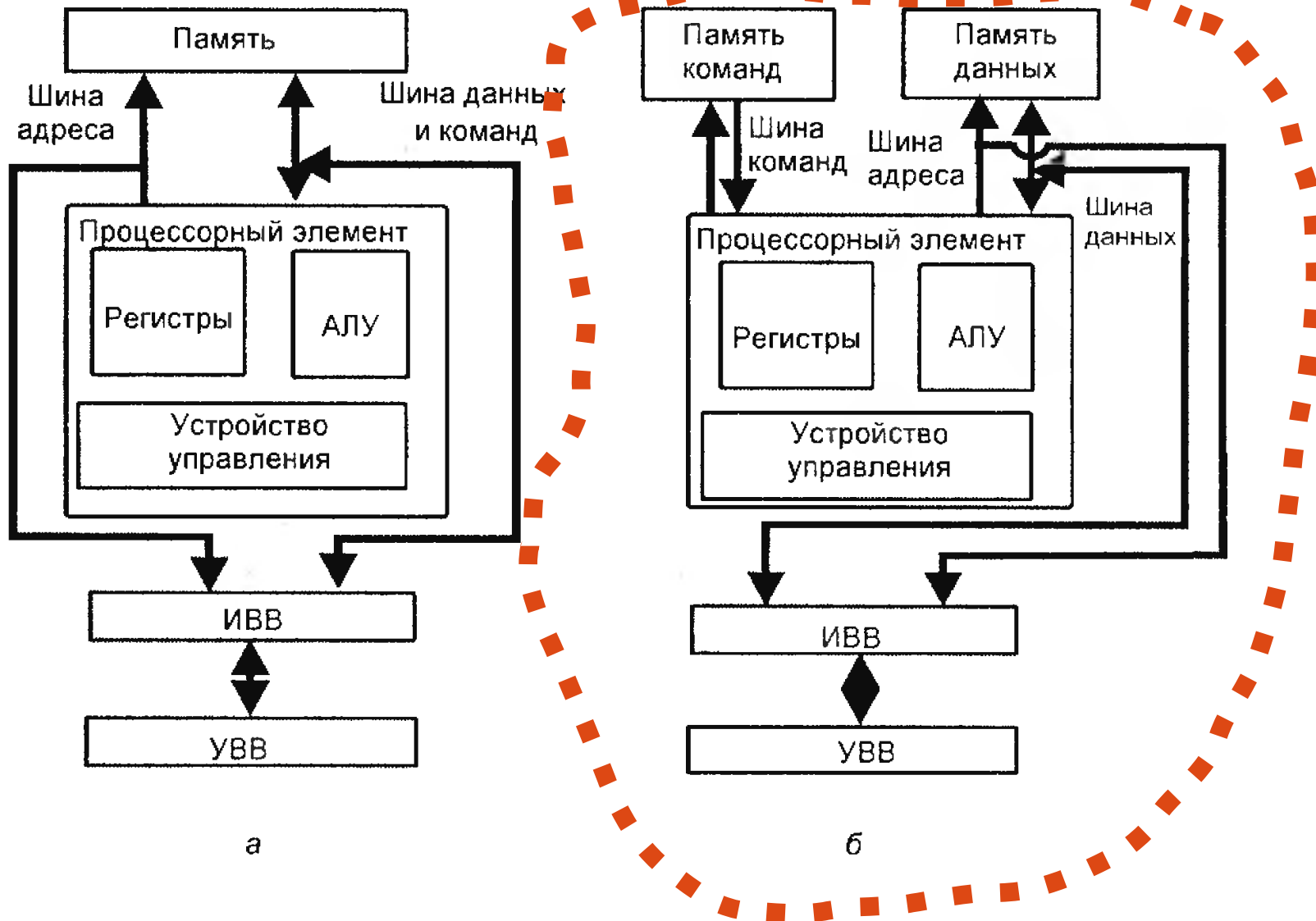
Отличие между регистрами и оперативной памятью состоит в том, что с регистрами можно производить любые операции (арифметические, логические, битовые), а в оперативную память можно лишь записывать данные из регистров.



# Фоннеймановская и гарвардская архитектура

В 1945 г. американский математик Джон фон Нейман сформулировал основные принципы работы современных компьютеров. Им была предложена архитектура, получившая его имя (von Neumann architecture) и предполагающая хранение программ и данных в общей памяти (1946 г.). Сегодня такая архитектура наиболее характерна для микропроцессоров, ориентированных на использование в компьютерах. Примером могут служить микропроцессоры семейства x86.

а) фон-неймановская б) гарвардская



ИВВ- интерфейсы ввода вывода, УВВ-устройства ввода/вывода

# Гарвардская архитектура

Архитектура, предполагающая раздельное использование памяти программ и данных, носит название гарвардской (Harvard architecture). Гарвардская архитектура позволяет центральному процессору работать одновременно как с памятью программ, так и с памятью данных, что существенно увеличивает производительность.

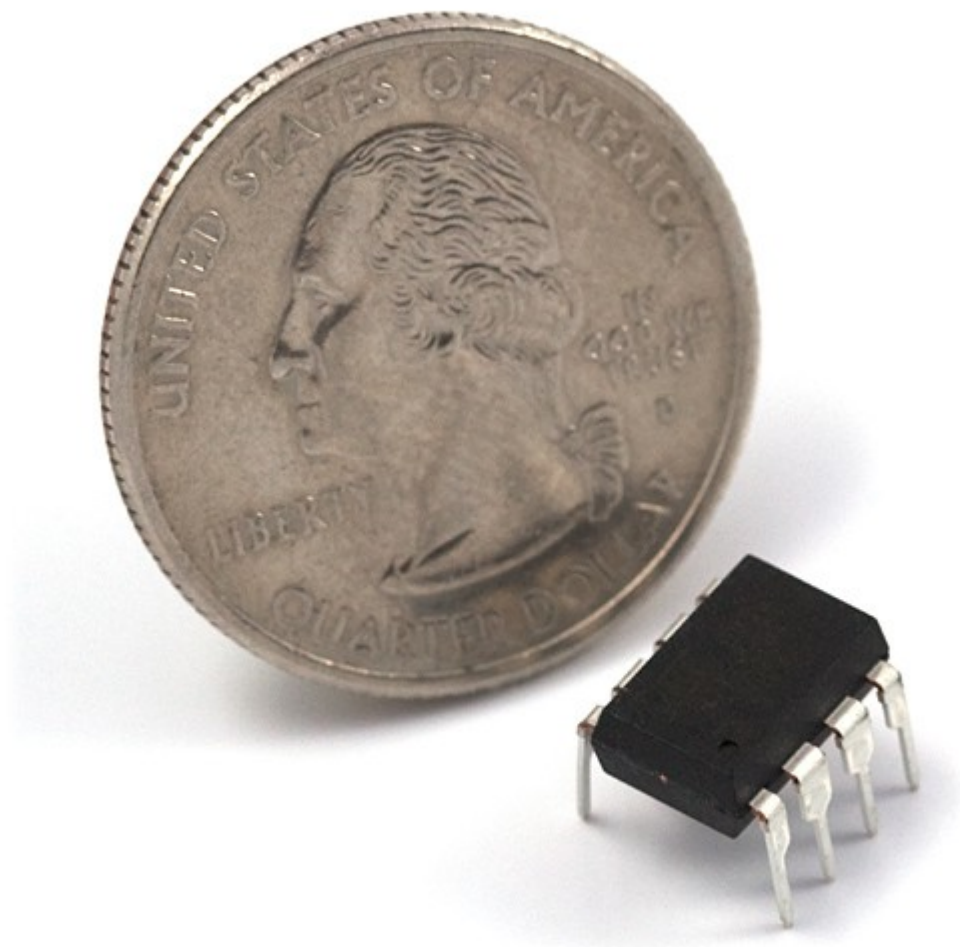
# AVR- Гарвардская архитектура

В микроконтроллерах AVR реализована Гарвардская архитектура, в соответствии с которой разделены не только адресные пространства памяти программ и памяти данных, но и шины доступа к ним. Каждая из областей памяти данных (оперативная память и EEPROM) также расположена в своем адресном пространстве.

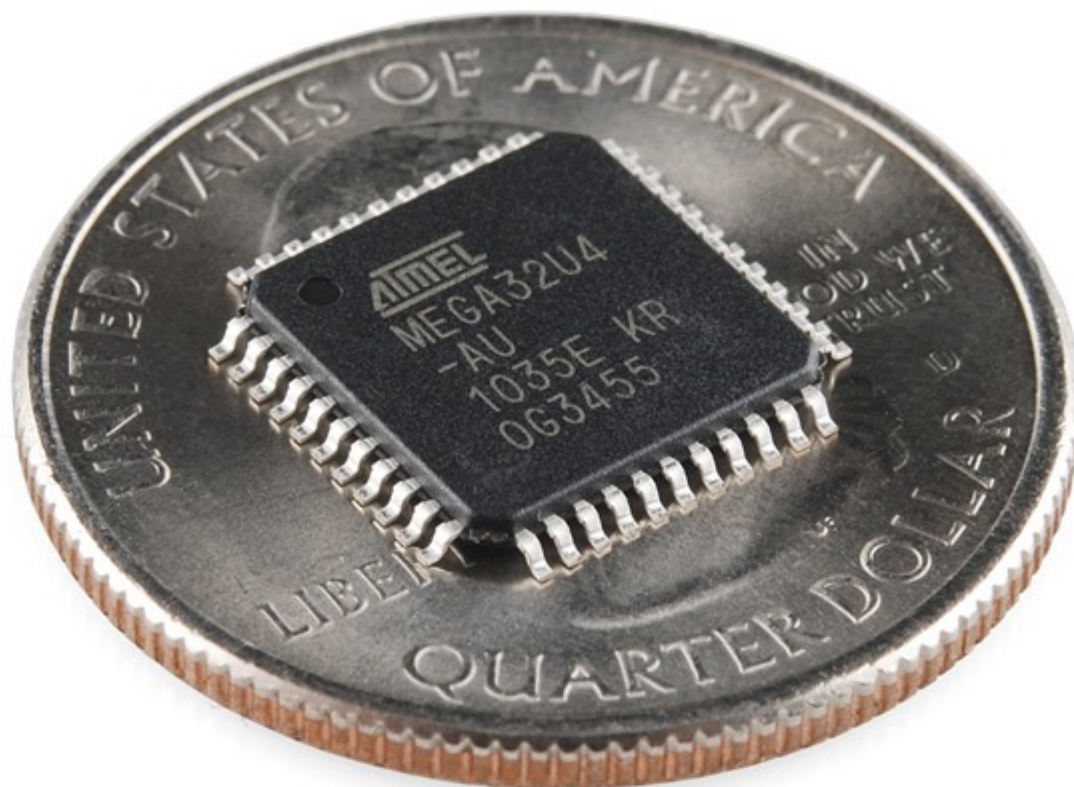
# Форм-факторы

- PDIP - Plastic Dual In-Line Package
- SOIC - Small Outline Integrated Circuit
- PLCC - Plastic Lead Chip Carrier
- TQFP - Thin Profile Quad Flat Package
- MLF - Micro Lead Frame Package

## PDIP - Plastic Dual In-Line Package



# TQFP





# Характеристики линейки AVR 8 bit tiny/mega

## 64-pin

- mega256
- mega128
- mega1281
- mega64
- mega649
- mega645
- mega641
- mega329
- mega325
- mega169
- mega165
- 90CAN128

## 100-pin

- mega2560
- mega1280
- mega6410
- mega6490
- mega6450
- mega3290
- mega3250

## 28/32-pin

- mega168
- mega88
- mega8
- mega48
- tiny28

## 8-pin

- tiny45
- tiny25
- 90S2323
- 90S2343
- tiny15
- tiny13
- tiny12
- tiny11

## 40/44-pin XRAM (8051-pinout)

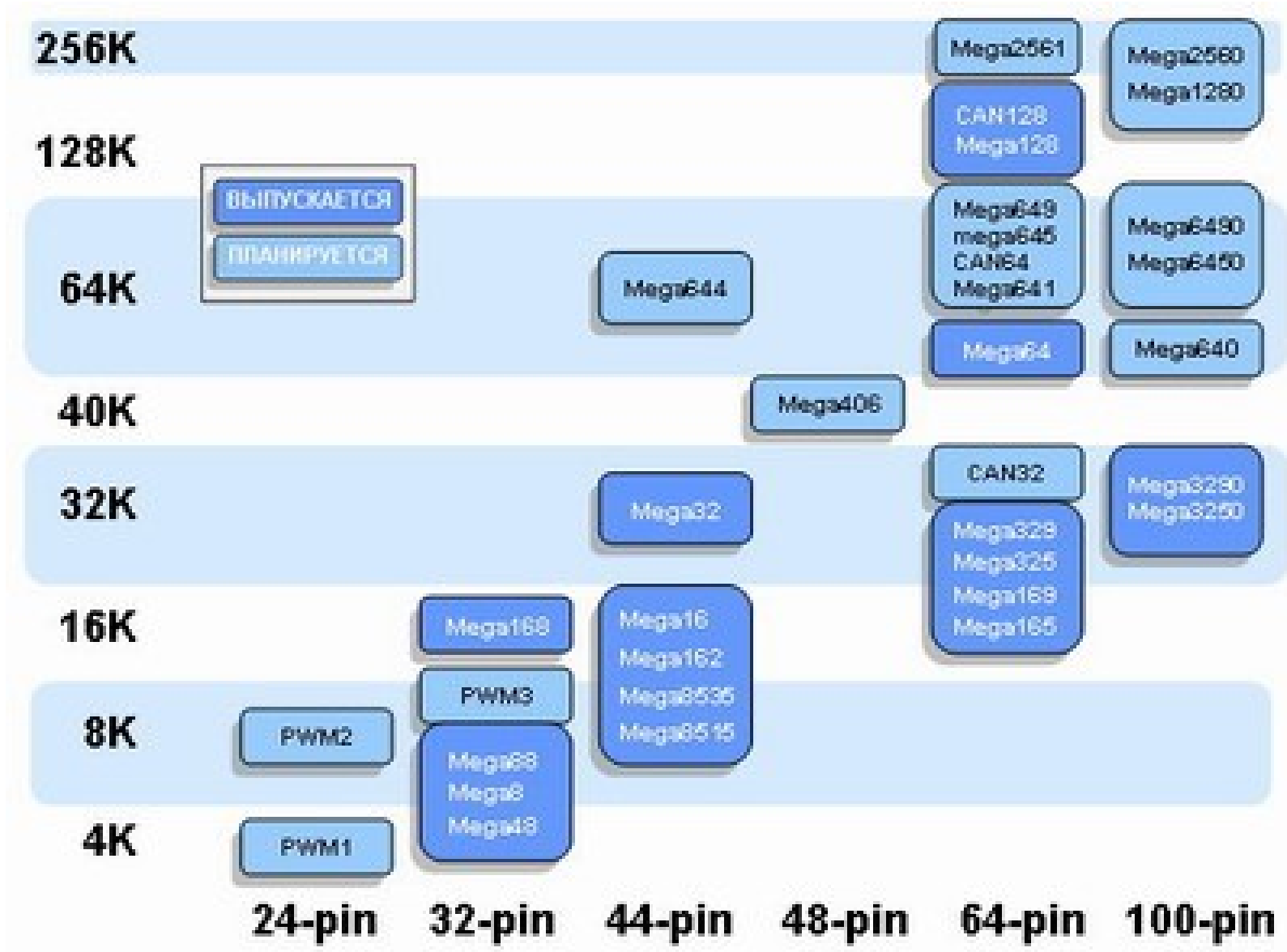
- mega162
- mega8515

## 40/44-pin ADC

- mega32
- mega16
- mega8535

## 20-pin

- 90S2313
- 90S1200
- tiny2313



Параметр	ATmega8535	ATmega8535L	ATmega16	ATmega16L	ATmega32	ATmega32L	ATmega128	ATmega128L	ATmega103	AT90CAN128
Объем flash-памяти, Кбайт	8	8	16	16	32	32	128	128	128	128
Объем EEPROM-памяти, Кбайт	0,5	0,5	0,5	0,5	1	1	4	4	4	4
Объем SRAM-памяти, Кбайт	0,5	0,5	1	1	2	2	4	4	4	4
Количество входов/выходов, шт.	32	32	32	32	32	32	53	53	40	53
Количество команд, шт.	130	130	130	130	131	131	133	133	121	133
Производительность, MIPS	16	8	16	8	16	8	16	8	6	16
Количество 8-разрядных таймеров, шт.	2	2	2	2	2	2	2	2	2	2
Количество 16-разрядных таймеров, шт.	1	1	1	1	1	1	2	2	1	2
Разрядность АЦП, бит	10	10	10	10	10	10	10	10	10	10
Количество каналов АЦП, шт.	8	8	8	8	8	8	8	8	8	8
8-разрядный ШИМ-генератор, шт.	4	4	4	4	4	4	2	2	–	–
16-разрядный ШИМ-генератор, шт.	–	–	–	–	–	–	6	6	–	–
Интерфейс I2C	+	+	+	+	+	+	+	+	+	+
Интерфейс USART	1	1	1	1	1	1	2	2	1	2
Интерфейс SPI	+	+	+	+	+	+	+	+	+	+
Встроенный CAN 2.0A и 2.0B контроллер	–	–	–	–	–	–	–	–	–	+
Тип корпуса	DIP40	DIP40	DIP40	DIP40	DIP40	DIP40	TQ64	TQ64	TQ64	TQ64
Рабочее напряжение питания 2,7... 5,5 В	–	+	–	+	–	+	–	+	–	–
Рабочее напряжение питания 4,5... 5,5 В	+	–	+	–	+	–	+	–	+	+
Макс. рабочая тактовая частота, МГц	16	8	16	8	16	8	16	8	6	16

# Цены (розничные)

- AT-tiny от ~0.8 до 3 USD (зависит от типа контроллера)
- AT-Mega от ~2 до 10 USD

# Программирование микроконтроллера

Программа заносится во Flash-память AVR как с помощью обычного программатора, так и с помощью SPI-интерфейса, в том числе непосредственно на собранной плате. Возможностью внутрисхемного программирования (функция ISP) через коммуникационный интерфейс SPI обладают все микроконтроллеры AVR, кроме Tiny11 и Tiny28.

# Самопрограммирование

Все микроконтроллеры семейства Mega имеют возможность

**самопрограммирования**, т. е.

самостоятельного изменения содержимого своей памяти программ. Эта особенность

позволяет создавать на их основе очень гибкие системы, алгоритм работы которых будет меняться самим микроконтроллером в зависимости от каких-либо внутренних условий или внешних событий.

# ARDUINO

=

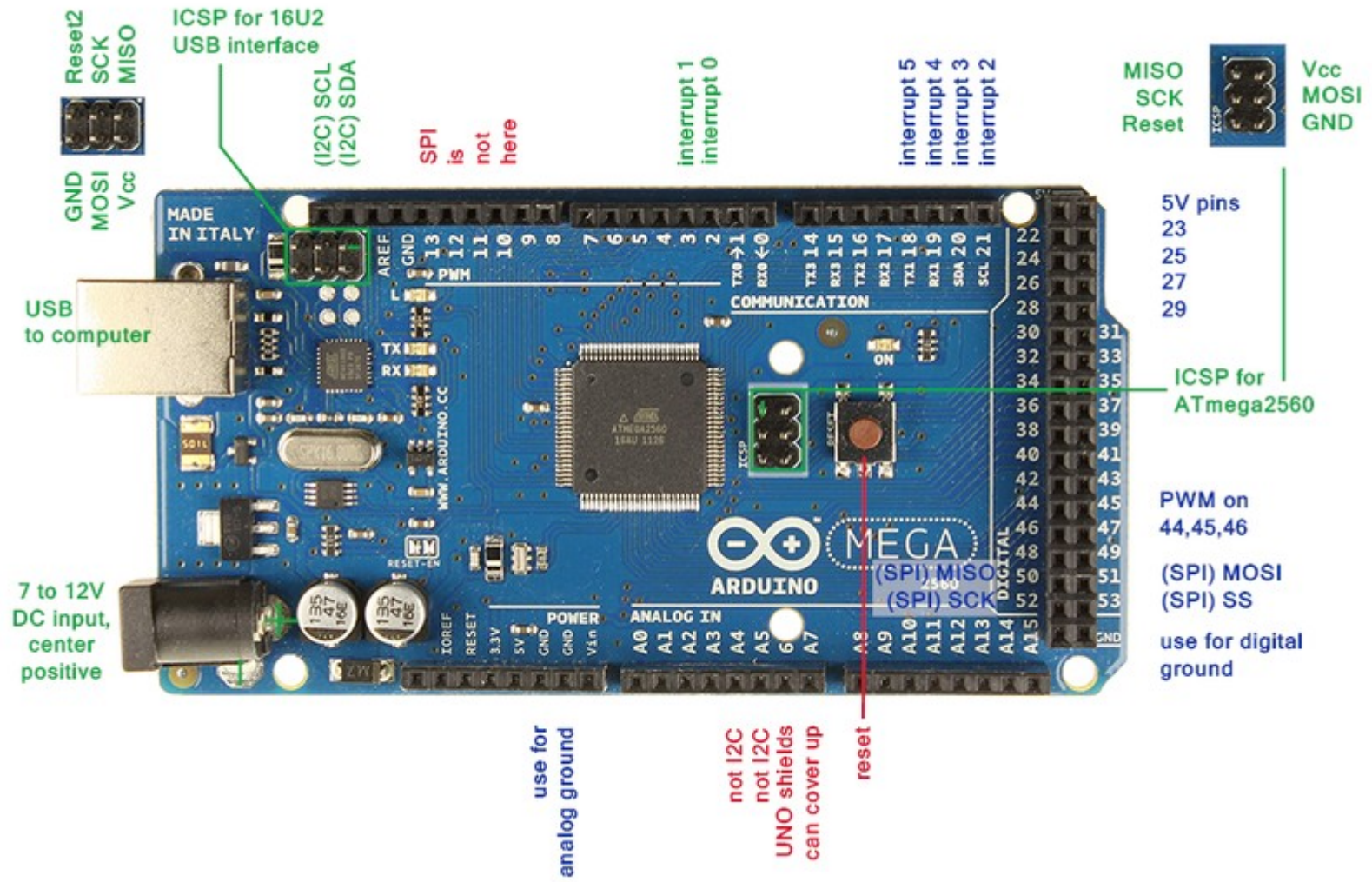
Обычный микроконтроллер (ATMega168/328 или AT Mega 2560)

+ стандартизованная HW обвязка для соединения с PC по USB

+ сервисная программа/загрузчик, реализующая функцию самопрограммирования

# Формула Arduino- используем на ЛР

- Arduino= AVR MC+ загрузчик+стандартизованная плата

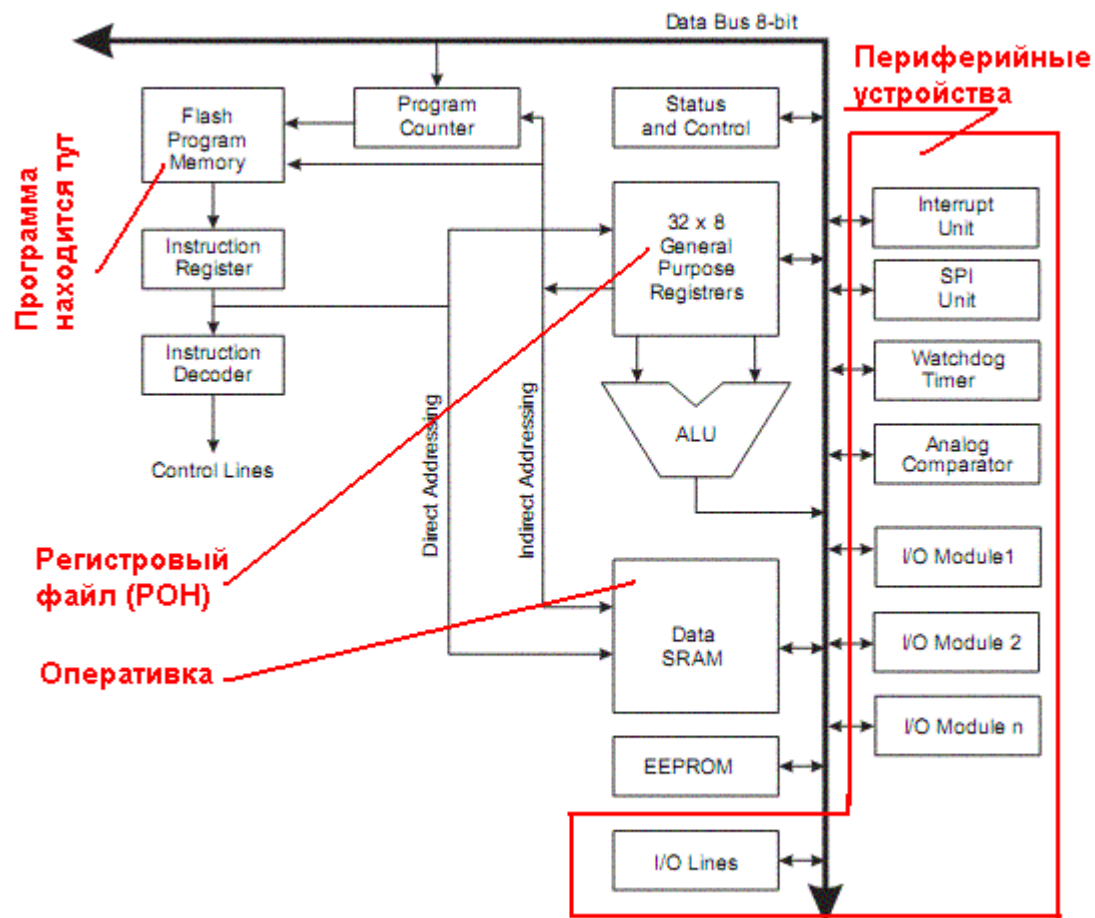




# ARDUINO

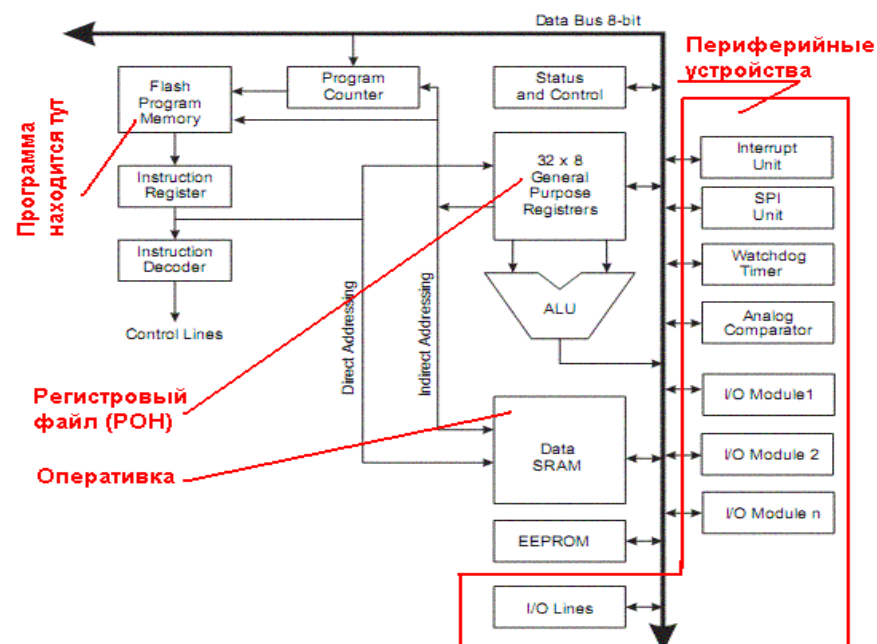
- Удобная для нас платформа для прототипирования
- Возможно использовать GCC,ASM
- Примеры
- Можно использовать библиотеки Arduino(нужно ли-решать вам), можно работать на низком уровне ( как в AVR Studio и в AVR Studio)

# Структурная схема контроллера AVR



# Память программ

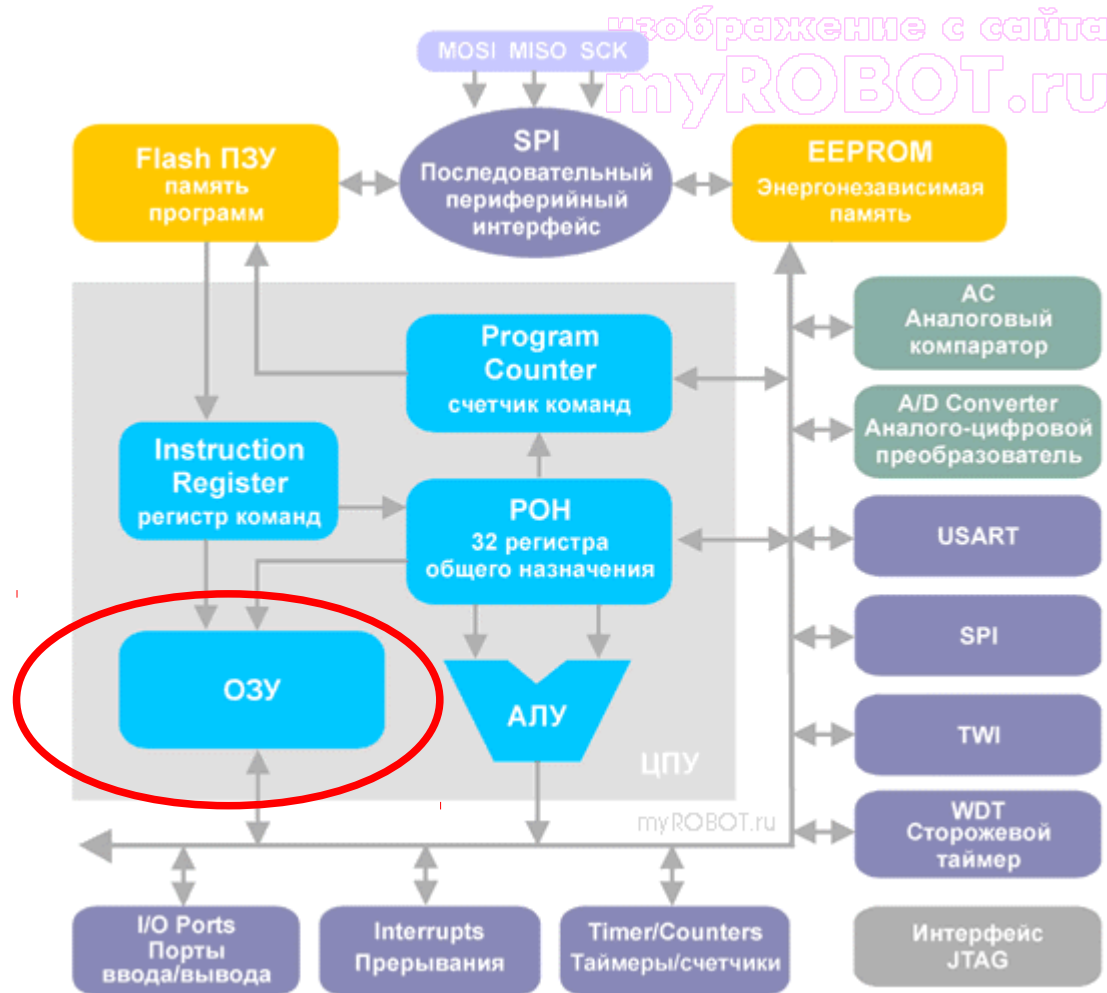
Память программ предназначена для хранения последовательности команд, управляющих функционированием микроконтроллера, и имеет 16-ти битную организацию. Все AVR имеют Flash-память программ, которая может быть различного размера - от 1 до 256 КБайт. Ее главное достоинство в том, что она построена на принципе электрической перепрограммируемости, т. е. допускает многократное стирание и запись информации.



# Число циклов перепрограммирования

Гарантированное число циклов перезаписи Flash-памяти у микроконтроллеров AVR второго поколения составляет не менее 10 тыс. циклов при типовом значении 100 тыс. циклов. (В официальной технической документации Atmel Corp. указывается значение 10 тыс. циклов.)

# Память данных



# Память данных

Память данных объединяет в единое адресное пространство три независимых банка: регистровая память, оперативная память (ОЗУ - оперативное запоминающее устройство или RAM) и энергонезависимая память (ЭСППЗУ или EEPROM).

# Карта памяти avr16

Регистровый файл

R0
R1
R2
⋮
R31

Память

\$0000
\$0001
\$0002
⋮
\$001F

Регистры ввода/вывода

\$00
\$01
⋮
\$3F

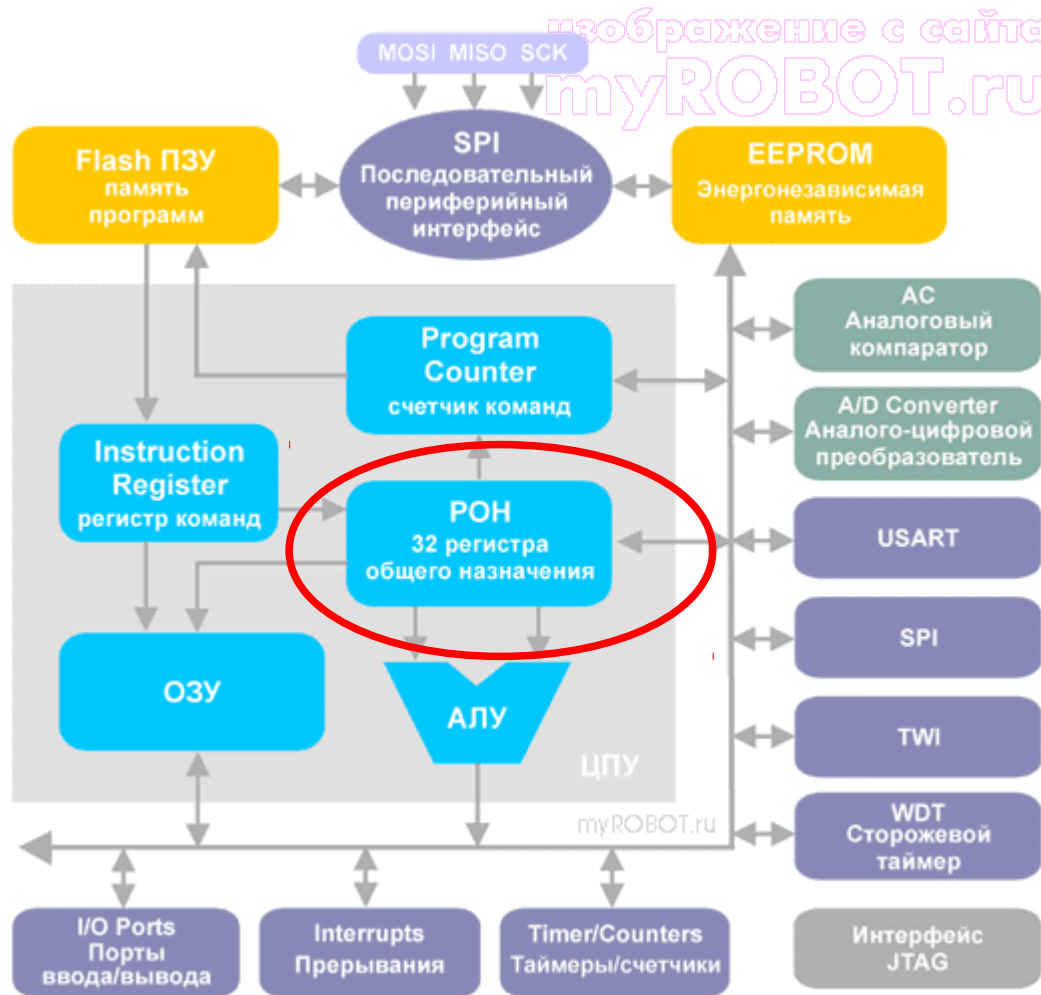
\$0020
\$0021
⋮
\$005F

EEPROM
--------

Внутреннее ОЗУ

\$C060
\$0061
⋮
\$015F

# Регистровая память





# Регистровая память (РОН и РВВ)

Регистровая память включает 32 регистра общего назначения (РОН или GPR), объединенных в файл, и служебные регистры ввода/вывода (РВВ). И те и другие расположены в адресном пространстве ОЗУ, но не являются его частью.

# Служебные регистры

В области регистров ввода/вывода расположены различные служебные регистры (регистры управления микроконтроллером, регистры состояния и т. п.), а также регистры управления периферийными устройствами, входящими в состав микроконтроллера. *По сути, управление микроконтроллером заключается в управлении этими регистрами.*

# Пример

IO Порт микроконтроллера может работать в нескольких режимах ( два режима работы как вход и как выход)

чтобы сконфигурировать порт B

`DDRD = 0xFF; // все пины порта D как выходы`

`PORTD = 0x0f; // все пины порта D в 1`

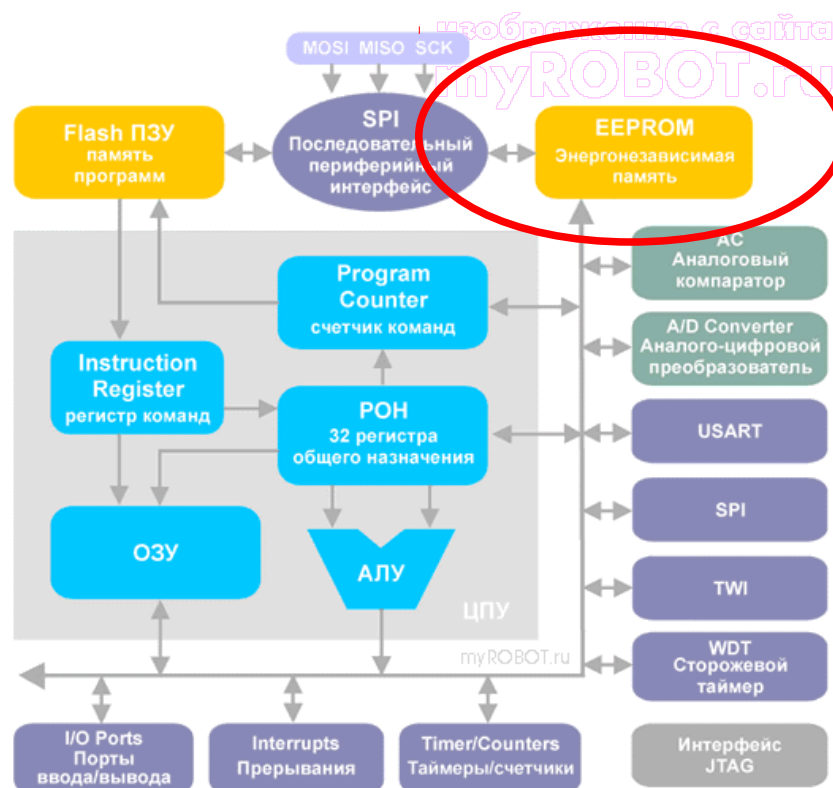
# Энергонезависимая память данных (EEPROM)

Для долговременного хранения различной информации, которая может изменяться в процессе функционирования микроконтроллерной системы, используется EEPROM-память.

Размер EEPROM от 64 Байт до 4 КБайт. Этот тип памяти, доступный программе микроконтроллера непосредственно в ходе ее выполнения, удобен для хранения промежуточных данных, различных констант, коэффициентов, серийных номеров, ключей и т.п. EEPROM может быть загружена извне как через SPI интерфейс, так и с помощью обычного программатора.

Число циклов стирание/запись - не менее 100 тыс

Данные после выключения питания не будут утеряны.



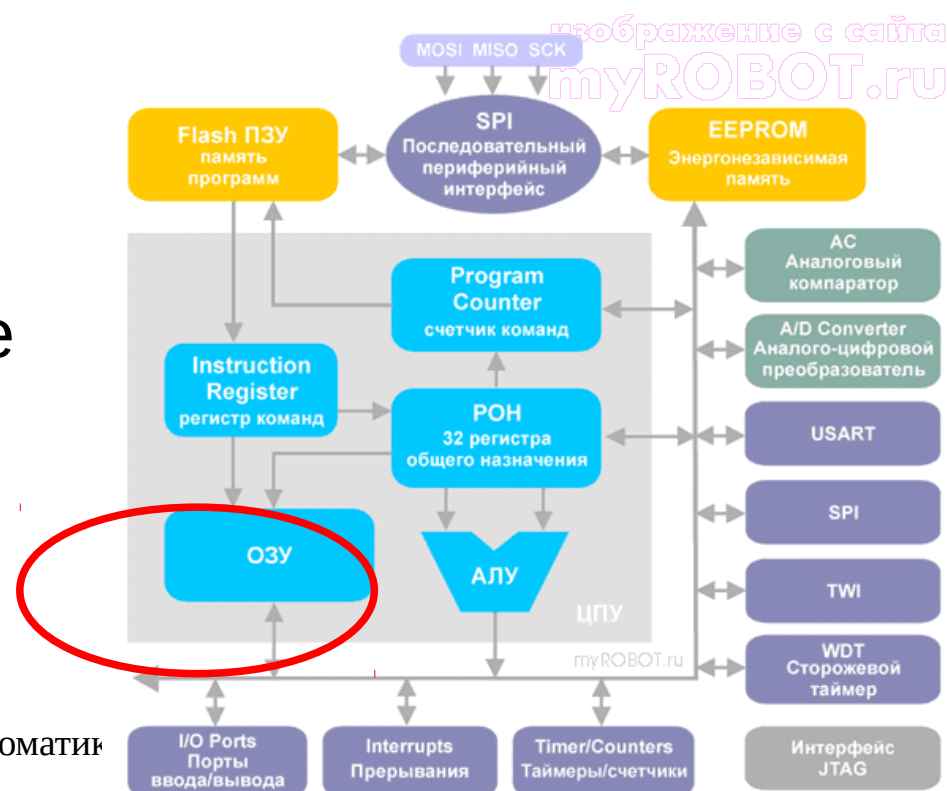
# Внутренняя память (SRAM)

Внутренняя оперативная статическая память Static RAM (SRAM) имеет байтовый формат и используется для оперативного хранения данных.

Число циклов перезаписи неограниченно

Память не является энергонезависимой и после отключения питания все данные будут утеряны.

Для некоторых МК доступно подключение внешней памяти до 64К



# Периферия контроллера

- Порты цифровые Digital Pins (от 3 до 48)
- Порты ввода/вывода
- Порты прерываний
- Порты таймеров/счетчиков
- сторожевой таймер
- Аналоговые входы (АЦП 10 разрядов)
- Аналоговый компаратор ( подключаемый к прерыванию)
- UART ( от 1 до 4х)
- JTAG
- SPI
- Супервайзер питания
- ШИМ -выходы

# Токовая нагрузка на цифровые порты

Порты ввода/вывода AVR имеют число независимых линий "вход/выход" от 3 до 53. Каждая линия порта может быть запрограммирована на вход или на выход. Мощные выходные драйверы обеспечивают токовую нагрузочную способность

**20 мА на линию порта (втекающий ток) при максимальном значении 40 мА,**

что позволяет, например, непосредственно подключать к микроконтроллеру светодиоды и биполярные транзисторы. Общая токовая нагрузка на

**все линии одного порта не более 80 мА**

(все значения приведены для напряжения питания 5 В).

# Прерывания

Система прерываний - одна из важнейших частей микроконтроллера. Все микроконтроллеры AVR имеют многоуровневую систему прерываний. Прерывание прекращает нормальный ход программы для выполнения приоритетной задачи, определяемой внутренним или внешним событием.



# Таймеры/счетчики (TIMER/COUNTERS)

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.

# Использование таймеров-счётчиков

- Точное формирования временных интервалов
- Подсчет импульсов на выводах микроконтроллер
- Формирование последовательности импульсов
- Тактирования приемопередатчика последовательного канала связи

# Таймер-счетчик в режиме ШИМ

В режиме ШИМ (PWM) таймер/счетчик может представлять собой широтно-импульсный модулятор и используется для генерирования сигнала с программируемыми частотой и скважностью

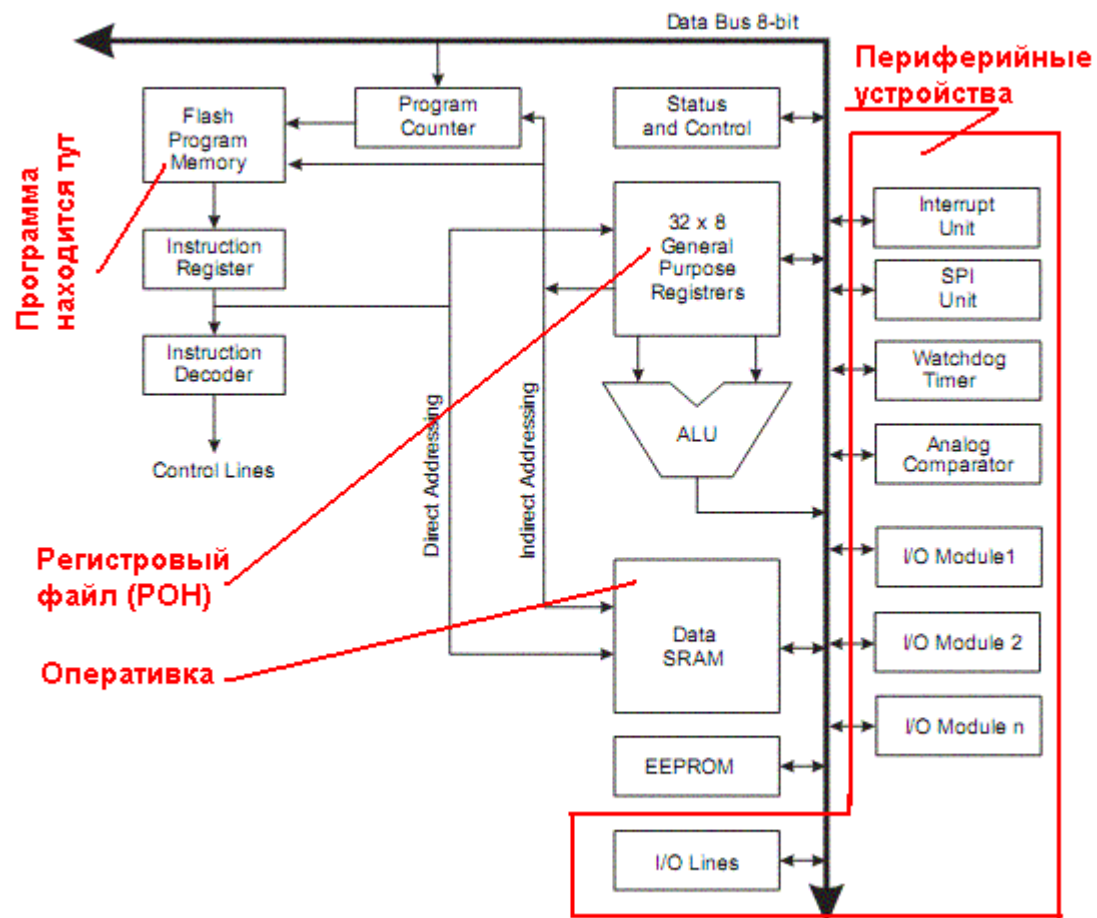
# ADC (АЦП)

- 10и разрядное АЦП ( от вообще нет у AT TinY до нескольких у MEGA)
- АЦП последовательного приближения
- Частота до 15 кГц
- Можно использовать внутренний или внешний источник опорного напряжения
- У старших AVR Mega есть возможность использовать АЦП в режиме дифференциального входа (например, когда один вход вычитается из другого плюс умножается на коэф) — полезно для измерительных мостов

# Управление АЦП

За АЦП отвечает регистр REFS1..0

- REFS1..0=00 — внешний ИОН к входу AREF
- REFS1..0 = 01 — VCC (Лучше AVCC+LC)
- REFD1..0 = 11 внутренний опорный 2.56 В
- Режимы: постоянно, от компаратора, от внешнего таймера, итд. , вернемся позднее



# Питание AVR

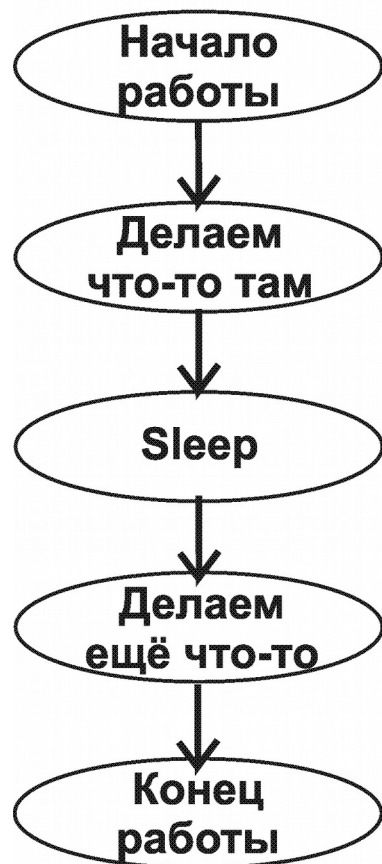
AVR функционируют при напряжениях питания от **1,8** до **6,0** Вольт. Ток потребления в активном режиме зависит от величины напряжения питания и частоты, на которой работает микроконтроллер, и составляет менее 1 мА для 500 кГц, 5 ... 6 мА для 5 МГц и 8 ... 9 мА для частоты 12 МГц.

# Режимы AVR

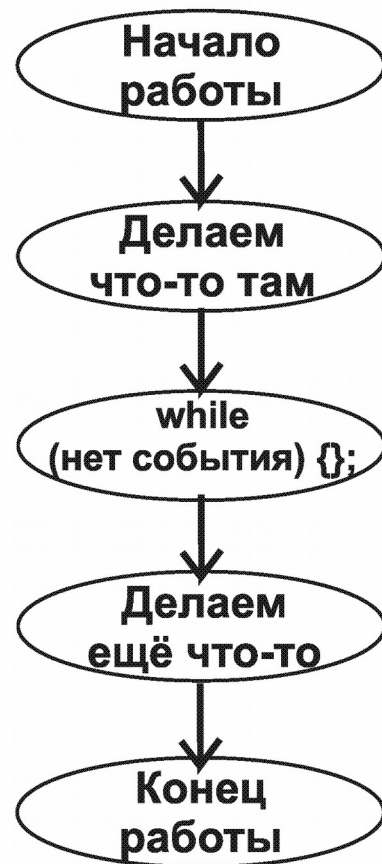
- При работе от аккумуляторных и батарейных источников питания в целях продолжения времени автономной работы применяются режимы энергосбережения.
- Режим энергосбережения выбирается программно.



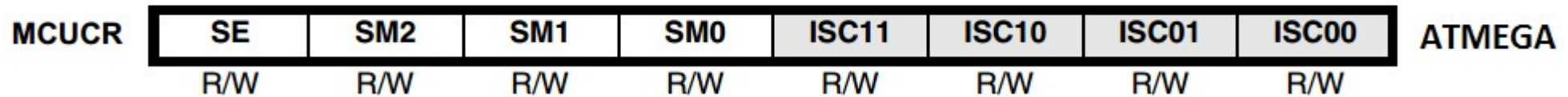
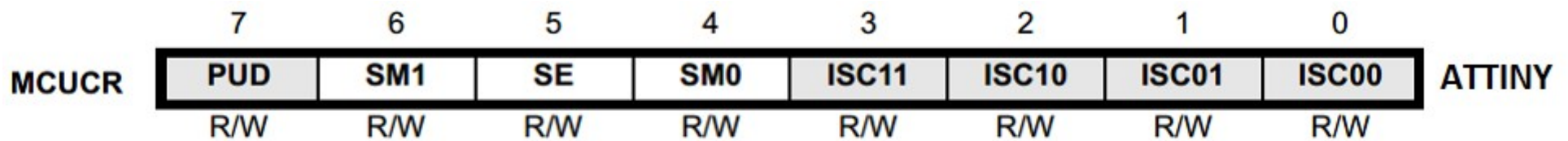
## Спящий режим



## Без экономии электропитания



# Управление энергосбережением



SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Standby

# Механизм конкуррирования энергосбережения

Шаг 1: выбираем режим, устанавливаем нужные биты регистра (SM2, SM1, SM0 )

Шаг 2: разрешаем конфигурацию-устанавливаем бит SE ->1

Усыпить микросхему asm sleep;

**Важно! Нужно продумать схему, кто ее разбудит!\***

\* еще раз вернемся к теме, когда будем говорить про прерывания.

# Режимы энергопотребления: IDLE

**Режим холостого хода (IDLE).** Прекращает работу только процессор и фиксируется содержимое памяти данных, а внутренний генератор синхросигналов, таймеры, система прерываний, SPI, UART и сторожевой таймер продолжают функционировать. Ток потребления не превышает 2,5 мА на частоте 12 МГц.

(Компаратор, сторожа, АЦП можно выключить)

# Стоповый режим (POWER DOWN)

Сохраняется содержимое регистрового файла, но останавливается внутренний генератор синхросигналов, и, следовательно, останавливаются все функции, пока не поступит сигнал внешнего прерывания или аппаратного сброса. При включенном сторожевом таймере ток потребления в этом режиме составляет около 80 мкА, а при выключенном - менее 1 мкА. (Все приведенные значения справедливы для напряжения питания 5 В).

**Важно: Заснет — все. Чип просыпается долго! Микросхема не узнает сколько она проспала. Микросхема не может разбудить себя сама.**

# Экономичный режим (POWER SAVE)

Продолжает работать только генератор таймера, что обеспечивает сохранность временной базы. Все остальные функции отключены.

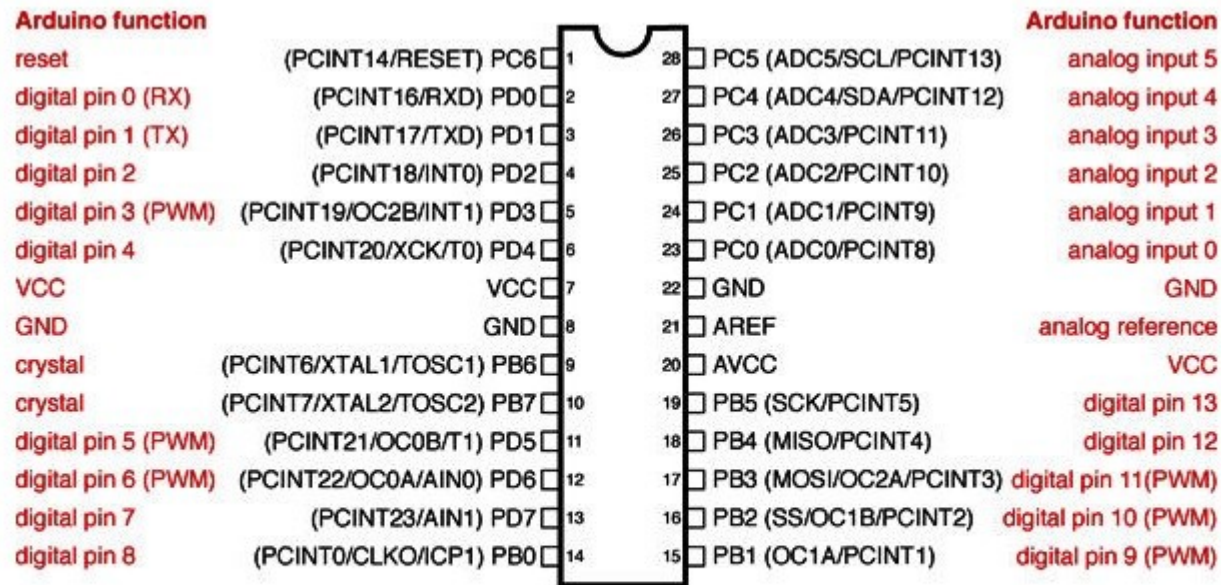
**Важно: Заснет — все. Чип просыпается долго! Микросхема знает сколько она находилась в режиме сна. Можем сами себя разбудить по таймеру.**

# ADC Noise Reduction

Этот режим останавливает процессор, но оставляет работать АЦП, систему внешних прерываний, интерфейс TWI, сторожевой таймер и таймер/счётчик2;

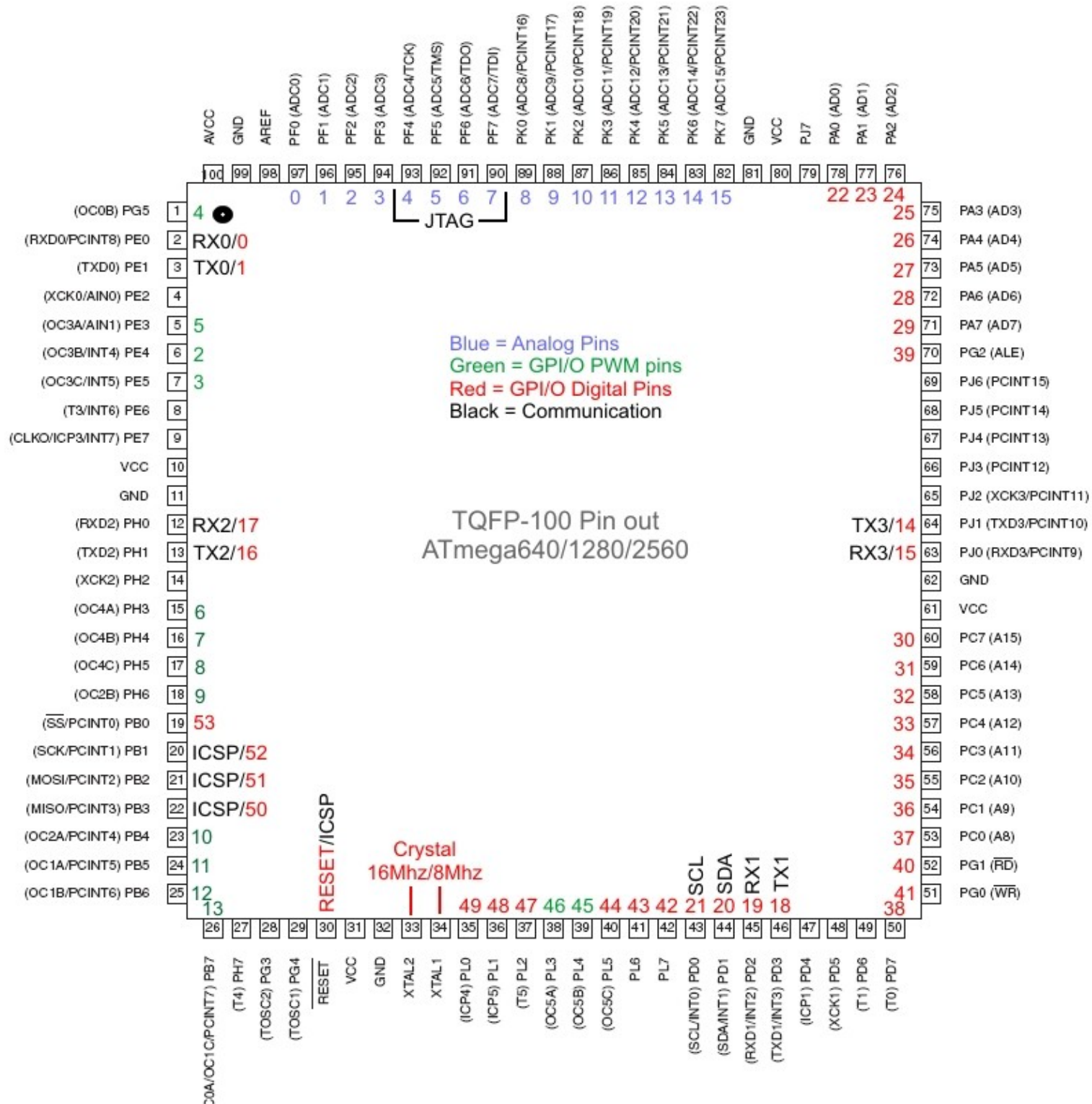
- Останавливаются: только часы I/O, CPU и FLASH.
- Используется для того, чтобы уменьшить внутренние помехи (возникающие при переключении транзисторов самой микросхемы) при работе АЦП - так получаются измерения более высокой точности (но зачем? Если нужна высокая точность, то АЦП-то так себе...)

# Цоколевка АТМмега328



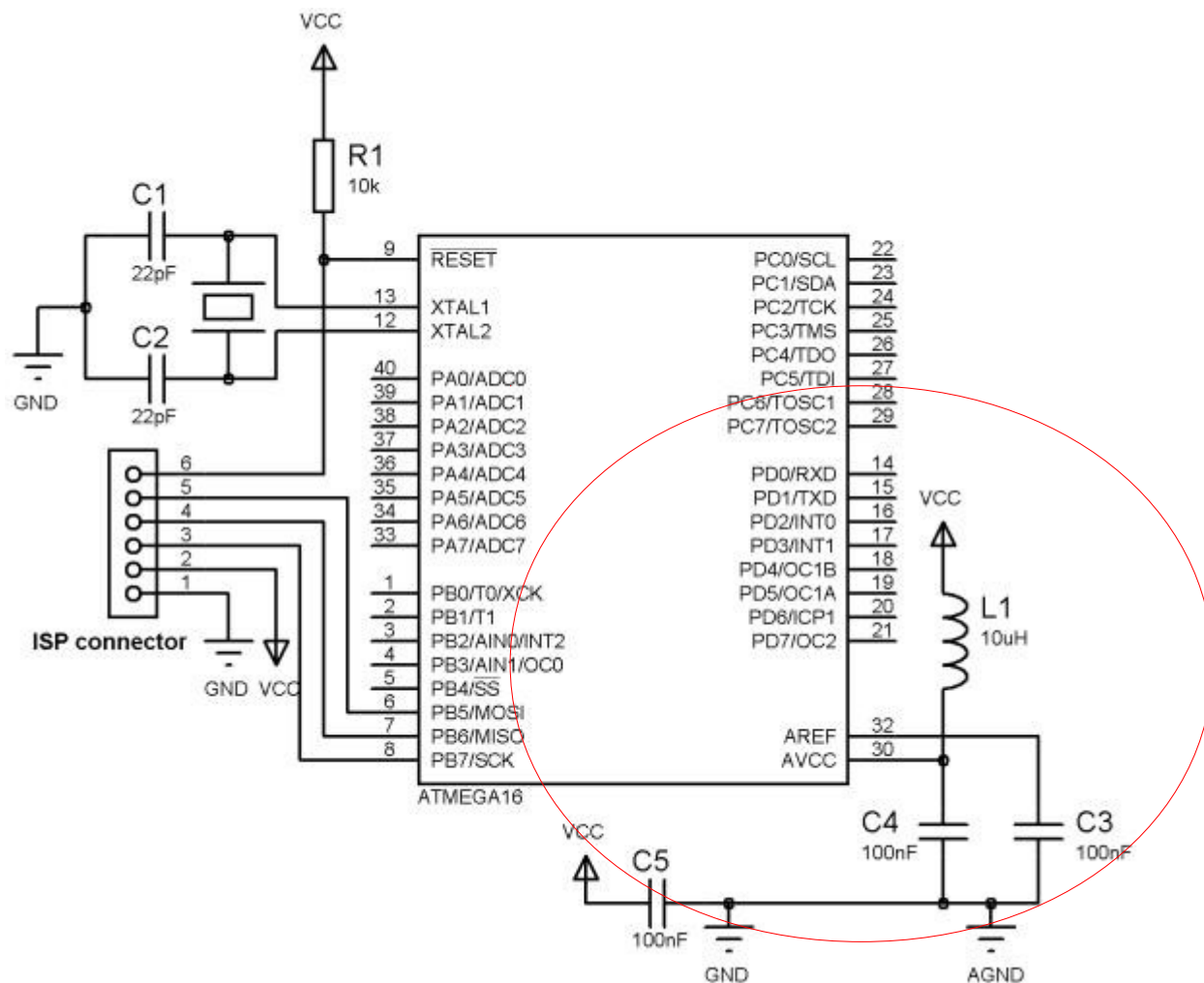
Digital Pins 11, 12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.





- VCC,GND питание
- XTAL1 XTAL2 — порты кварцевого резонатора (тактирующий сигнал)
- RESET
- AVCC, AGND, AREF — питание, опорное напряжение для аналоговой части
- P — порты ( [P][название А,В...][номер]

# Что нужно, чтобы контроллер заработал с полным функционалом?



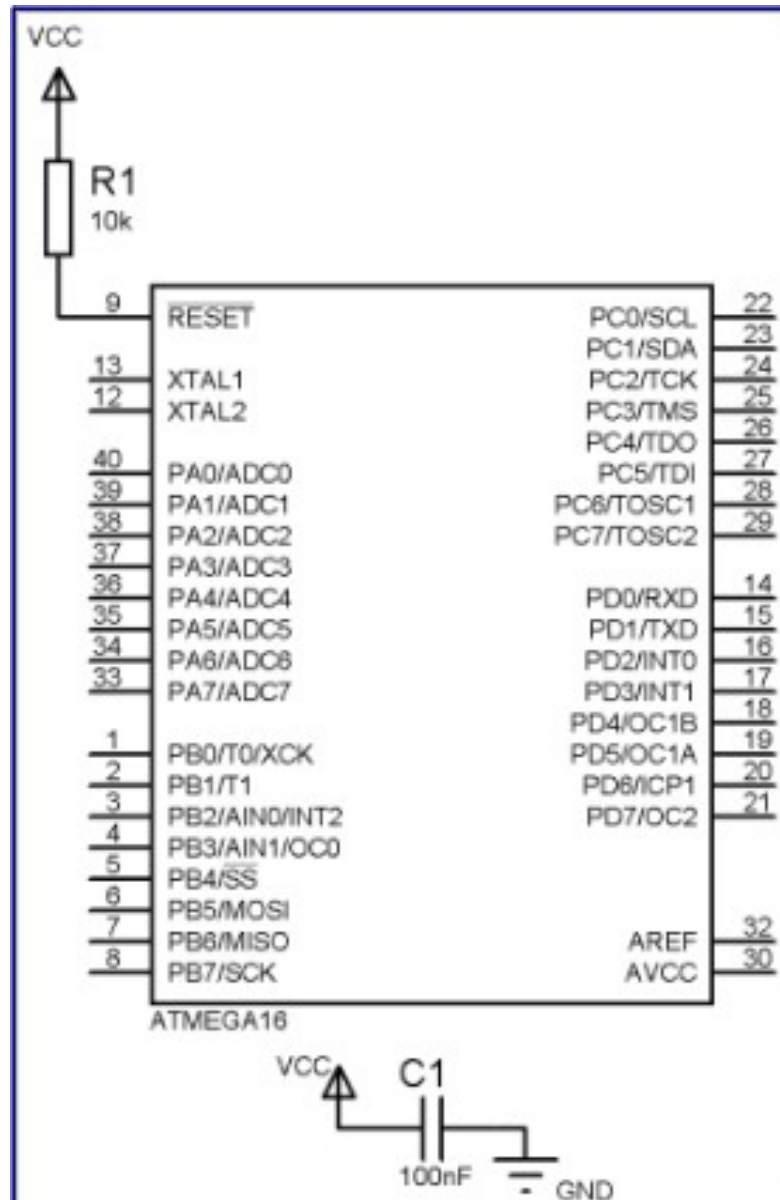
# Объяснения

- *Кварцевый резонатор* и конденсаторы *C1, C2* (по 22пФ) обеспечивают микроконтроллер и все его периферийные устройства качественным тактирующим сигналом (максимальная частота – 16.0МГц).
- *Резистор R1* (10К), обеспечивает высокий уровень на входе RESET, необходимый для стабильной работы микроконтроллера. Если, во время работы микроконтроллера, напряжение на этом пине упадет ниже определенного уровня, то произойдет сброс микроконтроллера и возможно нарушение работы задуманного алгоритма.
- - *ISP connector* используется для внутрисхемного программирования, то есть необходим для записи написанной вами программы в память микроконтроллера прямо на плате (не вынимая микроконтроллер из устройства).

# Необязательная часть

- *Дроссель L1 и конденсаторы C3, C4* обеспечивают напряжением питания аналоговые периферийные устройства а также некоторые регистры портов ввода/вывода. Если у микроконтроллера отсутствует аналоговая часть, соответственно, отсутствуют и пины аналогового питания, как результат – данные компоненты **не нужны**.

# Минималистические требования



VCC от  
1,8 до 6,0 Вольт

# Минималистические требования

- Тактирование осуществляется от встроенного тактового генератора (макс 8МГц), требуется установка Фьюз
- Аналоговая часть МК и некоторые регистры ввода/вывода лишены питания и неработоспособны
- Прошивка(программирование) контроллера выполняется в программаторе в колодке до монтажа устройства.