

Подключение микроконтроллера. Ликбез.

[AVR. Учебный курс 12 июля 2008 DI HALT 313 комментариев](#)

Казалось бы простая тема, а однако в комментах меня завалили вопросами как подключить микроконтроллер. Как подключить к нему светодиод, кнопку, питание. Что делать с **AGND** или **AREF**. Зачем нужен **AVCC** и все в таком духе. Итак, раз есть вопросы, значит тема не понятна и надо дать по возможности исчерпывающий ответ. Все описываю для контроллеров AVR, но для какихнибудь PIC все очень и очень похоже. Т.к. принципы тут едины.

[Чтобы понимать ряд терминов активно упоминающихся в этой статье, надо сначала прочитать статью про порты ввода-вывода.](#)

Питание

Для работы микроконтроллеру нужна энергия — электричество. Для этого на него естественно нужно завести питалово. Напряжение питания у МК **Atmel AVR** разнится от **1.8** до **5** вольт, в зависимости от серии и модели. Все **AVR** могут работать от 5 вольт (если есть чисто низковольтные серии, то просьба уточнить в комментах, т.к. я таких не встречал). Так что будем считать что напряжение питания контроллера у нас всегда 5 вольт или около того. Плюс напряжения питания обычно обозначается как **Vcc**. Нулевой вывод (а также Земля, Корпус, да как только его не называют) обозначают **GND**. Если взять за пример комповый блок питания. То черный провод это **GND** (кстати, земляной провод традиционно окрашивают в черный цвет), а красный это +5, будет нашим **Vcc**. Если ты собираешься запитать микроконтроллер от батареек, то минус батареек примем за **GND**, а плюс за **Vcc** (главное чтобы напряжение питания с батареек было в заданных пределах для данного МК, позырь в даташите. Параметр обычно написан на первой странице в общем описании фич:

- Operating Voltages

- 1.8 — 5.5V (ATtiny2313V)

- 2.7 — 5.5V (ATtiny2313)

- Speed Grades

- ATtiny2313V: 0 — 4 MHz @ 1.8 — 5.5V, 0 — 10 MHz @ 2.7 — 5.5V

- ATtiny2313: 0 — 10 MHz @ 2.7 — 5.5V, 0 — 20 MHz @ 4.5 — 5.5V

Обрати внимание, что есть особые низковольтные серии (например 2313V низковольтная) у которых нижняя граница напряжения питания сильно меньше. Также стоит обратить внимание на следующий пункт, про частоты. Тут показана зависимость максимальной частоты от напряжения питания. Видно, что на низком напряжении предельные частоты ниже. А низковольтные серии раза в два медленней своих высоковольтных коллег. Впрочем, разгону все процессоры покорны ;))))

Для работы контроллерам серии **AVR** достаточно только питания. На все входы **Vcc** надо подать наши 5 (или сколько там у тебя) вольт, а все входы **GND** надо посадить на землю. У микроконтроллера может быть много входов **Vcc** и много входов **GND** (особенно если он в квадратном **TQFP** корпусе. У которого питалово со всех сторон торчит). Много выводов сделано не для удобства монтажа, а с целью равномерной запитки кристалла со всех сторон, чтобы внутренние цепи питания не перегружались. А то представь, что подключил ты питалово только с одной стороны, а с другой стороны чипа навесил на каждую линию порта по светодиоду, да разом их зажег. Внутренняя тонкопленочная шина питания, офигев от такой токовой нагрузки, испарилась и проц

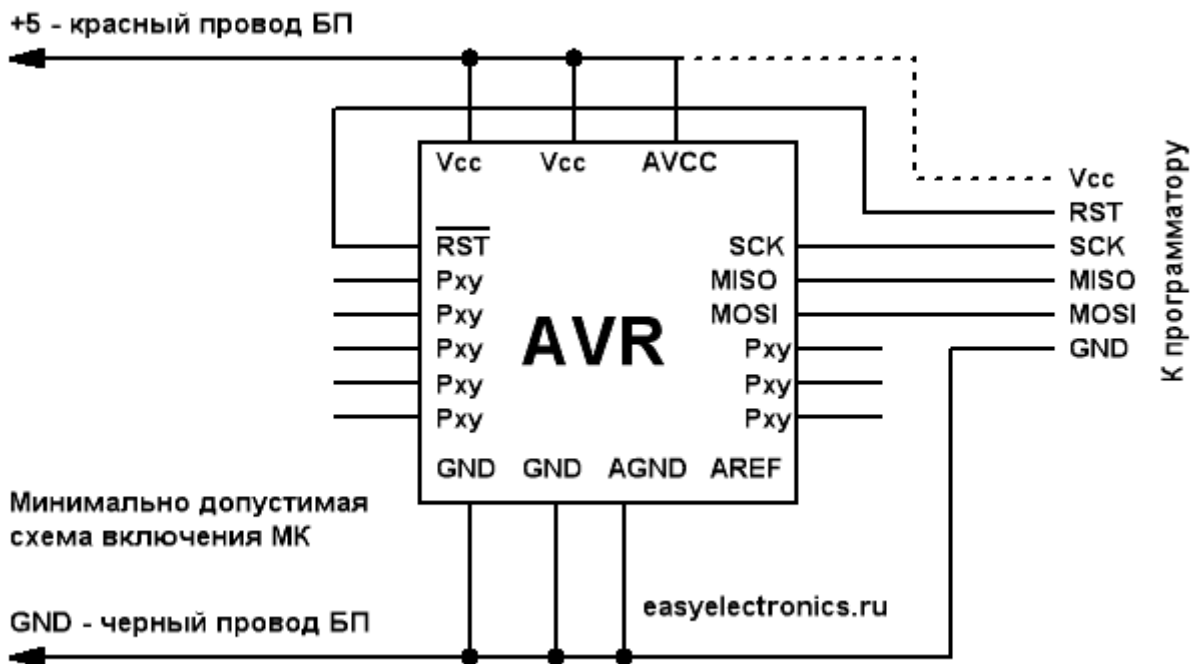
взял ВНЕЗАПНО и без видимых, казалось бы, причин отбросил копыта. Так что **ПОДКЛЮЧАТЬ НАДО ВСЕ ВЫВОДЫ Vcc и GND**. Соединить их соответственно и запитать.

Отдельные вопросы вызывают **AGND** и **AVCC** — это аналоговая земля и питание для Аналого-Цифрового Преобразователя. АЦП это очень точный измеритель напряжения, поэтому его желательно запитать через дополнительные фильтры, чтобы помехи, которые не редки в обычной питающей цепи, не влияли на качество измерения. С этой целью в точных схемах проводят разделение земли на цифровую и аналоговую (они соединены должны быть только в одной точке), а на **AVCC** подается напряжение через фильтрующий дроссель. Если ты не планируешь использовать АЦП или не собираешься делать точные измерения, то вполне допустимо на **AVCC** подать те же 5 вольт, что и на **Vcc**, а **AGND** посадить на ту же землю что и все. **Но подключать их надо обязательно!!!** ЕМНИП от AVCC питается также порт А.

Warning!!!

В чипе Mega8 похоже есть ошибка на уровне топологии чипа — Vcc и AVcc связаны между собой внутри кристалла. Между ними сопротивление около (!!!) 50м Для сравнения, в ATmega16 и ATmega168 между Vcc и AVcc сопротивление в десятки МЕГА ом! В даташите на этот счет никаких указаний нет до сих пор, но в [одном из топиков за 2004 год на AVRfreaks](#) сказано, что люди бодались с цифровым шумом АЦП, потом написали в поддержку Atmel мол WTF??? А те, дескать, да в чипе есть бага и Vcc и AVcc соединены внутри кристалла. В свете этой инфы, думаю что ставить дроссель на AVcc для Mega8 практически бесполезно. Но AVcc запитывать надо в любом случае — кто знает насколько мощная эта внутренняя связь?

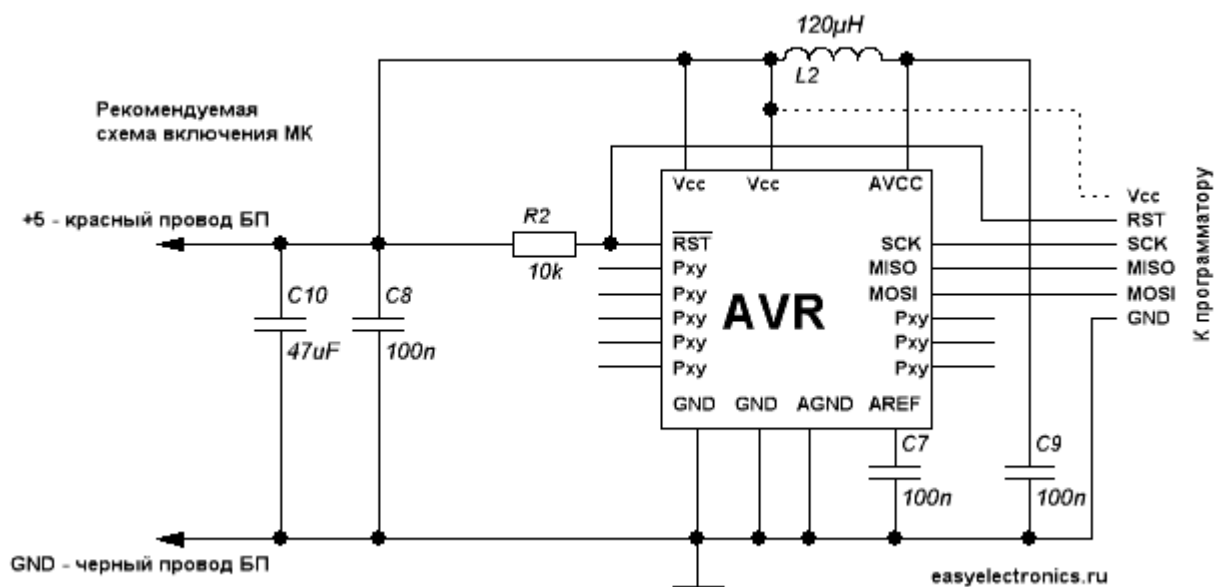
Простейшая схема подключения Микроконтроллера AVR приведена ниже:



Это необходимый минимум чтобы контроллер запустился. Провод Vcc до программатора показан пунктиром поскольку он не обязателен. Если ты собираешься питать МК от внешнего источника, то он не нужен. Но я все же рекомендую для начала питать всю систему (МК+программатор) от одного источника — больше вероятность успешной прошивки :) Для учебной цели, диодиком помогать,

сойдет и так.

Но настолько все упрощать я не рекомендую. Лучше сразу добавить парочку навесных внешних элементов. Правильней будет. Чтобы было вот так:



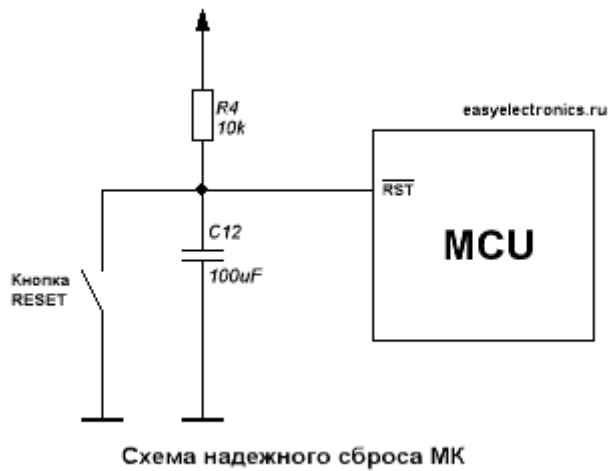
Как видишь, добавился дроссель в цепь питания **AVCC**, а также конденсаторы. Хорошим тоном является ставить керамический конденсатор на сотню нанофард между **Vcc** и **GND** у каждой микросхемы (а если у микрухи много вход питания и земель, то между каждым питанием и каждой землей) как можно ближе к выводам питания — он сгладит краткие импульсные помехи в шине питания вызванные работой цифровых схем. Конденсатор на 47мКФ в цепи питания сгладит более глубокие броски напряжения. Кондесатор между **AVcc** и **GND** дополнительно успокоит питание на **АЦП**.

Вход **AREF** это вход опорного напряжения **АЦП**. Туда вообще можно подать напряжение относительно которого будет считать **АЦП**, но обычно используется либо внутренний источник опорного напряжения на 2.56 вольт, либо напряжение на **AVCC**, поэтому на **AREF** рекомендуется вешать конденсатор, что немного улучшит качество опорного напряжения **АЦП** (а от качества опоры зависит адекватность показаний на выходе **АЦП**).

Схема сброса

Резистор на **RESET**. Вообще в **AVR** есть своя внутренняя схема сброса, а сигнал **RESET** изнутри уже подтянут резистором в 100кОм к **Vcc**. НО! Подтяжка это настолько дохлая, что микроконтроллер ловит сброс от каждого чиха. Например, от касания пальцем ножки **RST**, а то и просто от задевания пальцем за плату. Поэтому крайне рекомендуется **RST** подтянуть до питания резистором в 10к. Меньше не стоит, т.к. тогда есть вероятность, что внутрисхемный программатор не сможет эту подтяжку пересилить и прошить МК внутри схемы не удастся. 10к в самый раз.

Есть еще вот такая схема сброса:



Она замечательна тем — при включении схемы конденсатор разряжен и напряжение на **RST** близко к нулю — микроконтроллер не стартует, т.к. ему непрерывный сброс. Но со временем, через резистор, конденсатор зарядится и напряжение на **RST** достигнет лог1 — МК запустится. Ну, а кнопка позволяет принудительно сделать сброс если надо.

Задержка будет примерно $T=R \cdot C$ для данного примера — около секунды. Зачем эта задержка? Да хотя бы для того, чтобы МК не стартовал раньше чем все девайсы платы запитаются и выйдут на установившийся режим. В старых МК (**AT89C51**, например) без такой цепочки, обеспечивающей начальный сброс, МК мог вообще не стартовать.

В принципе, в **AVR** задержку старта, если нужно, можно сделать программно — потупить с пол секунды прежде чем приступить к активным действиям. Так что кондер можно выкинуть нафиг. А кнопку... как хочешь. Нужен тебе внешний **RESET**? Тогда оставь. Я обычно оставляю.

[....]

<http://easyelectronics.ru/podklyuchenie-mikrokontrollera-likbez.html>