

В прошлых статьях я советовал тебе не лезть к этим битам. И на это были свои основания, так как неправильно выставив эти биты ты можешь наглухо заблокировать контроллер для дальнейшей перепрошивки или вообще какого либо использования.

Но без знания этой особенности контроллера далеко не уедешь. Так что распишу все по порядку. У разных версий контроллеров число FUSES разное, какие то могут отсутствовать, но основные есть всегда. Вот по ним и пройдемся.

Конфигурационные биты находятся в особой области памяти и могут быть изменены только с помощью программатора при записи контроллера. Есть старший байт и младший байт. Младший байт обычно отвечает за частоту, а старший за всякие фенечки.

Итак, главное:

В Atmel AVR принята следующая нотация: сброшенный в ноль fuse bit считается активным, т.е. включенным.

Пример Бит RSTDSBL, как можно догадаться из названия, это RESET DISABLE. Включил эту опцию и у тебя нога RESET превращается в порт ввода-вывода, но за это ты теряешь возможность перешить контроллер через ISP.

Так вот, чтобы выключить RESET (и получить большое западло с прошивкой в обмен на мелкую подачку в виде дополнительной ножки) в этот бит надо записать 0.

С одной стороны нелогично и криво. Как бы во всем мире принята нотация, что ноль это выключено, а тут, понимаешь, наоборот. С другой стороны, это их контроллер, что хотят то и делают. Один раз запомнить и все. Да и вообще, в электронике часто за сигнал берут ноль.

Однако контроллеры делают электронщики, а прошивающие программы — программисты. Как бы логично. И вот эти программисты взяли и заварили адскую путаницу с галочками. Нет бы им раз и навсегда принять за стандарт, что галочка это 1, а не ВКЛЮЧЕНО (что, напомню, является нулем). И поэтому в одних прошивающих программах галочка означает, что опция включена (в FUSE бит записывается 0), в других, обычно написанных электронщиками, галочка означает единицу. Т.е. с точностью до наоборот.

А что будет если перепутать? А будет ОЧЕНЬ плохо. Контроллер войдет в неправильный режим и может заблокируется наглухо. Т.е. раз прошил и все. Приехал.

Нет, спасти его можно, но для этого тебе потребуются дополнительные ухищрения в виде высоковольтного программатора, JTAG адаптера или генератора тактов. Все зависит от

того в какой режим ты загонишь контроллер своими неправильными настройками.

Новичку, обычно, бывает проще сходить и купить новый МК, чем оживить заблокированный. Но не спеши отправлять его в помойку. Пометь и отложи на будущее, разберешься оживишь.

Конфигурация тактового сигнала

По умолчанию все контроллеры AVR (кроме старых серий AT90S2313, AT90S8535 итд) сконфигурированы так, чтобы работать от внутреннего источника тактов. Т.е. стоит подать на них питание и они начинают работать. Ничего больше и не нужно.

За источник тактов отвечают биты **CKSEL**

Выставив их правильным образом можно выбрать частоту работы контроллера, а также источник тактового сигнала.

- $CKSEL3...0 = 0000$ — Внешний источник сигнала.

Т.е. на вход XTAL1 подаются прямоугольные импульсы. Такое иногда делают в синхронных системах, когда несколько контроллеров работают от одного генератора.

Техническое отступление

В этот режим часто попадают, когда пытаются выставить контроллер на работу от внешнего кварца ($CKSEL=1111$), но либо путают нотацию, либо из-за прикола с обратной нотацией битов во всяких извратских прошивающих программах. Раз и кристалл заблокировался. Но, на самом деле, наглухо, с помощью CKSEL, заблокировать кристалл нельзя. Обычно все решается напайкой кварца и запуском от этого кварца. Худшее же что может случиться — потребуются внешний генератор тактов. Который бы оживил кристалл. Делается он за пять минут из любой микросхемы TTL логики, например из K155ЛА3 — схем в инете навалом. Или на таймере 555, либо можно взять второй МК и на нем написать простую программку, дрыгающую ножкой. А если есть осциллограф, то с него можно поймать сигнал контрольного генератора — его клемма должна быть на любом осциле. Землю осцила на землю контроллера, а выход генератора на XTAL1.

Но что делать если зуд нестерпимый, контроллер залочен, а никакой микросхемы для реанимации под рукой нету? Тут иногда прокатывает метод пальца. Прикол в том, что на тело человека наводится весьма нефиговая наводка частотой примерно 50Гц. Всякий кто хватался за щупы осциллографа руками помнит какие шняги тут же возникают на экране — вот это оно! А почему бы эту наводку не заюзать как тактовый сигнал? Так что припаиваешь к выводу XTAL1 провод, хватаешься за него рукой, и жмешь на чтение или запись контроллера :) Предупреждаю сразу, метод работает через жопу, далеко не с первого раза, читает долго и порой с ошибками, но на перезапись FUSE битов в нужную сторону должно хватить. Пару раз у меня такой фокус получался.

$CKSEL3...0 = 0100$ – 8 MHz от внутреннего генератора(обычно по умолчанию стоят такие)

Для большинства AVR такая конфигурация CKSEL означает тактовку от внутреннего

генератора на 8МГц, но тут могут быть варианты. Так что в этом случае втыкай внимательно в даташит. В табличку Internal Calibrated RC Oscillator Operating Modes

Иногда нужно иметь внешний тактовый генератор, например, чтобы его можно было подстраивать без вмешательства в прошивку. Для этого можно подключить RC цепочку, как показано на схеме и подсчитать частоту по формуле $f = 1/3RC$, где f будет частотой в герцах, а R и C соответственно сопротивлением резистора и емкостью конденсатора, в омах и фарадах.

- CKSEL3...0 = 0101 – для частот ниже 0.9 MHz
- CKSEL3...0 = 0110 – от 0.9 до 3 MHz
- CKSEL3...0 = 0111 – от 3 до 8 MHz
- CKSEL3...0 = 1000 – от 8 до 12 MHz

Данная табличка справедлива только для ATmega16 у других МК может отличаться. Уточняй в даташите!

Проблема у внутреннего генератора и внешних RC цепочек обычно в нестабильности частоты, а значит если сделать на ней часы, то они будут врать, не сильно, но будут. Поэтому иногда полезно запустить контроллер на кварце, кроме того, только на кварце можно выдать максимум частоты, а значит и производительности проца.

- CKSEL3...0 = 1001 — низкочастотный «часовой» кварц.

На несколько десятков килогерц.

Используется в низкоскоростных устройствах, особенно когда требуется точная работа и низкое потребление энергии.

Для обычных кварцев ситуация несколько иная. Тут максимальная частота кварца зависит также и от бита **СКОРТ** когда СКОРТ = 1 то:

- CKSEL3...0 = 1010 или 1011 — от 0,4 до 0.9 MHz
- CKSEL3...0 = 1100 или 1101 — от 0,9 до 3 MHz
- CKSEL3...0 = 1110 или 1111 – от 3 до 8 MHz (либо от 1 до 16МГц при СКОРТ=0)

А если **СКОРТ** равен 0 то при тех же значения CKSEL можно поставить кварц от 1 до 16MHz.

Разумеется, кварц на 16MHz можно поставить только на Мегу без индекса "L". (Хотя, как показывает практика, Lку тоже можно неслабо разогнать. У меня ATmega8535L заработала на 16МГц, но были странные эффекты в работе. Поэтому я не стал так извращаться и разгон снял). Опять же, все выше сказанное в точности соответствует только Меге 16, у других может незначительно отличаться.

Бит **СКОРТ** задает размах тактового сигнала. Т.е. амплитуду колебаний на выходе с кварца. Когда СКОРТ = 1 то размах маленький, за счет этого достигается меньшее

энергопотребление, но снижается устойчиво к помехам, особенно на высоких скоростях (а предельной, судя по таблице выше, вообще достичь нельзя. Точнее запуститься то он может запустится, но вот надежность никто не гарантирует). А вот если СКОРТ активизировать, записать в него 0, то размах сигнала сразу же станет от 0 до питания. Что увеличит энергопотребление, но повысит стойкость к помехам, а значит и предельную скорость. При оверклокинге МК тем более надо устанавливать СКОРТ в 0.

Также стоит упомянуть бит SCKDIV8 которого нет в Atmega16, но который часто встречается в других контроллерах AVR. Это делитель тактовой частоты. Когда он установлен, т.е. в нуле, то частота выставленная в битах CKSEL0...3 делится на 8, на чем в свое время прилично застрял Длинный, долго пытаюсь понять чего это у него запахло не работает. Вся прелесть в том, что этот делитель можно отключить программно, записав в регистр CLKPR нужный коэффициент деления, например один. Весь прикол в том, что SCKDIV8 активен по дефолту! Так что внимательней!

Биты SUT задают скорость старта МК после снятия RESET или подачи питания. Величина там меняется от 4ms до 65ms. Мне, за всю практику, пока не довелось эту опцию использовать — незачем. Так что ставлю на максимум 65ms — надежней будет.

Бит **RSTDISBL** способен превратить линию **Reset** в одну из ножек порта, что порой очень нужно когда на какой-нибудь крошечной Типу не хватает ножек на все задачи, но надо помнить, что если отрубить Reset то автоматически отваливается возможность прошивать контроллер по пяти проводкам. И для перешивки потребуется высоковольтный параллельный программатор, который стоит несколько тысяч и на коленке сделать его проблематично, хотя и возможно.

Второй заподьянский бит это **SPIEN** если его поставить в 1, то у тебя тоже мгновенно отваливается возможность прошивать по простому пути и опять будет нужен параллельный программатор. Впрочем, успокаивает то, что сбросить его через SPI невозможно, по крайней мере в новых AVR (в старых, в AT90S*** было можно)

WDTON отвечает за Собачий таймер, он же Watch Dog. Этот таймер перезагружает процессор если его периодически не сбрасывать – профилактика зависаний. Если WDTON поставить в 0, то собаку нельзя будет выключить вообще.

BODLEVEL и **BODEN** — это режим контроля за напряжением. Дело в том, что при определенном пороге напряжения, ниже критического уровня, контроллер может начать сильно глючить. Самопроизвольно может запорточить, например, EEPROM или еще что откосять. Ну, а ты как думал, не покорми тебя с пару недель — тоже глючить начнешь :)

Так вот, для решения этой проблемы есть у AVR встроенный супервизор питания. Он следит, чтобы напруга была не ниже адекватного уровня. И если напруги не хватает, то просто прижимает RESET и не дает контроллеру стартовать. Вот эти два фуза и рулят этой фицей. **BODEN** включает, а **BODLEVEL** позволяет выбрать критический уровень, один из двух. Какие? Не буду раскрывать, посмотри в даташите (раздел System Control and

Reset).

JTAGEN — Включить JTAG. По умолчанию активна. Т.е. JTAG включен. Из-за этого у MEGA16 (а также 32 и прочих, где есть JTAG) нельзя использовать вывода порта C, отвечающие за JTAG. Но зато можно подключать JTAG отладчик и с его помощью лезть контроллеру в мозги.

EESAVE — Защита EEPROM от стирания. Если эту штуку включить, то при полном сбросе МК не будет стерта зона EEPROM. Полезно, например, если в EEPROM записываются какие-либо ценные данные по ходу работы.

BOOTRST — перенос стартового вектора в область бутлоадера. Если эта галочка включена, то МК стартует не с адреса 00000, а с адреса бутсектора и вначале выполняет бутлоадер. Подробней про это было написано в статье про прошивку через лоадер.

BOOTSZ0..1 — группа битов определяющая размер бут сектора. Подробней смотри в даташите. От контроллера к контроллеру они отличаются.

Lock Bits

Это, собственно, и к фузам то отношения не имеет. Это биты защиты. Установка этих битов запрещает чтение из кристалла. Либо флеша, либо EEPROMA, либо и того и другого сразу. Нужно, только если ты продаешь свои устройства. Чтобы злые конкуренты не слили прошивку и не заказали в Китае более 9000 клонов твоего девайса, оставив тебя без штанов. Опасности не представляют. Если ты заблокируешь ими кристалл, то выполни полное стирание и нет проблемы.

Характерной особенностью установленных лок битов является считываемая прошивка — в ней байты идут по порядку. Т.е. 00,01, 02, 03, 04... FF, 00... Видел такую срань? Значит не судьба тебе спереть прошивку — защищена =)

Техника безопасности

И главное правило при работе с FUSE битами — ВНИМАНИЕ, ВНИМАНИЕ и ЕЩЕ РАЗ ВНИМАНИЕ! Не выставляйте никогда FUSE не сверившись с даташитом, даже если срисовываете их из проверенного источника.

Мало ли в какой нотации указал их автор, в прямой или инверсной. Так что если повторяете какую-либо конструкцию, то перед тем как ставить фузы, проверьте то ли вы вообще ставите!

Обязательно разберитесь что означает галочка в прошивающей программе. Ноль или единицу. Включено или выключено! Стандарта нет!!!

Если фуз биты задаются двумя числами — старший и младший биты, то выставляются они как в даташите. Где 0 это включено.