

Фьюзы – за что они отвечают?

CKSEL – выбор тактового генератора для микроконтроллера.

Для работы микроконтроллера (как и для любого процессора) нужны тактовые импульсы. Источником тактового сигнала может быть:

– внутренний RC генератор. Никаких дополнительных элементов не нужно. Удобно, но RC генератор имеет небольшую точность работы (вплоть до 10% погрешности) и, кроме того, «плавает» от температуры. Для некритичных по времени приложений вполне годится.

– внешний кварцевый (или керамический) резонатор. Нужен сам резонатор, плюс два конденсатора на 15-30пФ. Соответственно, будут заняты две ножки микроконтроллера – XLAT1 и XLAT2. Применяется там, где нужны точные замеры времени или частота работы микроконтроллера выше, чем может дать внутренний RC генератор.

– еще можно тактировать микроконтроллер от внешнего источника тактового сигнала. Это может быть другой микроконтроллер (для синхронизации работы) или внешняя схема, дающая нужный сигнал. Тактовый сигнал подается на ножку XLAT1.

Источник тактового сигнала для микроконтроллера задается комбинацией битов CKSEL3...0.

Это может быть (для ATtiny2313, выборочно):

CKSEL3...0 = 0000 – Внешний тактовый сигнал;

CKSEL3...0 = 0010 – Внутренний тактовый генератор – частота 4 МГц;

CKSEL3...0 = 0100 – Внутренний тактовый генератор – частота 8 МГц;

CKSEL3...0 = 1101 – Внешний тактовый генератор – кварц частотой от 3 до 8 МГц;

CKSEL3...0 = 1111 – Внешний тактовый генератор – кварц частотой больше 8 МГц.

CKOUT – разрешает вывод тактовой частоты на ножку CLKO микроконтроллера (для тактирования других устройств).

CKOUT = 1 – ножка микроконтроллера работает как обычный порт ввода-вывода;

CKOUT = 0 – на ножку микроконтроллера выдается сигнал тактового генератора.

CKOPT – задает размах тактового сигнала на внешнем генераторе.

CKOPT = 1 – размах небольшой – генератор работает в экономном режиме. Нормально генератор может работать лишь при небольших частотах и в условиях близким к идеальным. При значительных помехах, большой тактовой частоте, перепадах (скачках) напряжения питания, микроконтроллер может работать нестабильно;

СКОPT = 0 – задающий генератор работает на полную мощность, устойчив к помехам и может работать во всем диапазоне частот. Если нет особых требований к энергосбережению – советую всегда программировать этот бит.

SCKDIV8 – деление тактовой частоты на 8.

Тут все просто:

SCKDIV8= 1 – микроконтроллер работает на частоте задающего генератора;

SCKDIV8= 0 – микроконтроллер работает на частоте в 8 раз меньше частоты задающего генератора;

SUT – задает скорость запуска микроконтроллера.

После снятия «сброса» (или подачи питания) программа, записанная в микроконтроллер, начинает работать не мгновенно. Микроконтроллер выжидает некоторое время, для того, чтобы нормально запустился тактовый генератор, установилось напряжение питания и т.д. Время ожидания до запуска программы и задают биты **SUT1...0**. Чаще всего нам не критична скорость запуска, поэтому советую ставить на максимум.

SUT1..0 = 11 – максимальное время запуска (чуть больше 65 мS).

На время запуска еще влияет **CKSEL0**, но это уже детали ...

RSTDISBL –разрешает использовать ножку **Reset** как еще один порт ввода-вывода.

Иногда нужная вещь, но нужно знать -

после программирования **RSTDISBL** микроконтроллер уже нельзя будет прошить последовательным программатором! Поэтому без особой надобности не трогайте его.

RSTDISBL = 1 – ножка сброса работает как сброс;

RSTDISBL = 0 – ножка сброса работает как еще один порт ввода-вывода, последовательное программирование отключено.

SPIEN – разрешение на последовательное программирование.

По умолчанию запрограммирован (0) – разрешено последовательное программирование.

SPIEN = 0 – разрешено последовательное программирование;

SPIEN = 1 – запрещено последовательное программирование.

WDTON – включает Watch Dog Timer.

Для ответственных приложений, там, где недопустимо зависание программы (будь то ошибка программы или злостная помеха), применяют Watch Dog Timer. Это внутренний таймер микроконтроллера, работающий от своего независимого генератора. При переполнении этого таймера микроконтроллер сбрасывается и начинает выполнять программу с начала. Программист должен в тесте программы (обычно в главном цикле) вставить специальную команду обнуления этого таймера (WDR). Команда периодически выполняется и обнуляет таймер, не давая ему переполниться. Если микроконтроллер «повис» перестают выполняться команды обнуления, таймер переполняется и сбрасывает микроконтроллер.

WDTON = 1 – Watch Dog Timer – отключен (можно включить программно);

WDTON = 0 – Watch Dog Timer – включен (программно выключить нельзя).

В обычных приложениях не нужен.

BODLEVEL и BODEN – контроль напряжения питания микроконтроллера (Brown-out Detector).

Если питание микроконтроллера опустится к минимально допустимому или чуть ниже, то работа микроконтроллера будет нестабильной. Возможны ошибочные действия, потеря данных, случайное стирание EEPROM. Микроконтроллер умеет следить за уровнем своего питания (BODEN=0) и когда оно достигает уровня, который задается битами BODLEVEL, сбрасывается и держится в ресете пока уровень не поднимется до рабочего уровня. В некритических приложениях можно не использовать.

JTAGEN – разрешает интерфейс JTAG (внутрисхемный отладчик).

При активации некоторые линии микроконтроллера отдаются под интерфейс. Но зато можно подключать JTAG отладчик и с его помощью легко отладить любую программу прямо в схеме – удобно.

JTAGEN = 1 – запрещен JTAG;

JTAGEN = 0 – разрешен JTAG.

DWEN – бит, разрешающий работу DebugWire

– еще одного отладочного интерфейса. DebugWire однопроводный отладочный интерфейс работающий через ножку сброса, поэтому «не отнимает» у микроконтроллера ножки портов ввода-вывода.

DWEN= 1 – запрещен DebugWire ;

DWEN= 0 – разрешен DebugWire .

AVR микроконтроллеры могут во время своей работы изменять содержимое области программ (программировать сам себя).

SELFPRGEN – бит, разрешающей программе производить запись в память программ.

SELFPRGEN = 1 – изменение области программ запрещено;

SELFPRGEN = 0 – разрешено изменение области программ.

EESAVE – защита EEPROM от стирания.

При подаче команды полного стирания микроконтроллера (обычно осуществляется при каждом программировании кристалла) стирается и EEPROM. Если Вы хотите чтобы EEPROM оставалось нетронутым – активируйте этот фьюз. Это актуально если в EEPROM хранятся важные данные.

EESAVE = 1 – стирать EEPROM вместе с Flash;

EESAVE = 0 – оставлять EEPROM при очистке нетронутым.

AVR микроконтроллеры могут иметь бутлоадер – это область в конце памяти, в которой можно разместить загрузчик, который предназначен для загрузки и запуска основной программы.

BOOTRST – как раз и заставляет микроконтроллер запускаться с области бутлоадера.

BOOTRST = 1 – микроконтроллер запускает программу с нулевого адреса;

BOOTRST = 0 – микроконтроллер запускает программу с бутлоадера.

BOOTSZ0..1 – задает размер бут сектора (области памяти программ для бутлоадера).

Lock Bits – Это отдельный фьюз байт который предназначен для защиты области программ и/или EEPROM от копирования. Полное стирание восстанавливает эти биты в исходное состояние.

Еще раз повторяюсь, это не полный перечень фьюз бит, для каждого конкретного микроконтроллера смотрите даташит.