DESIGN NOTE #020

AUTHOR: AVRFREAKS

KEYWORDS: FUSES, LOCK BITS, ISP PROGRAMMING, PARALLELL PROGRAMMING

Understanding AVR Fuses and Lock bits

Introduction

When programming an AVR device you can, in addition to programming the Flash and EEPROM memories, program a number of Fuses and Lock bits to enable/disable features and/or different memory protection modes.

This document explains the difference between Fuses and Lock bits, and explain the consequence of programming them.

Overview

- Any AVR microcontroller will have two groups of programmable bits: Lock bits and Fuses.
- The unprogrammed state of a Fuse or a Lock bit is logic "high". Programming any Lock or Fuse bit will change its level to logic "low".
- The Lock bits and Fuses are not located in the normal flash or EEPROM space, nor are they accessible from the software, except for Lock bits related to the Boot Loader in devices with the Self-Programming feature.
- Fuses and Lock bits can be accessed in programming mode, using a Parallell/Serial Programming tool.

Lock Bits

All AVR devices contain two Lock bits named LB1 and LB2. Programming these (“0”) will add protection to the contents written to Flash and EEPROM memories according to Table 1 below. The level of protection is divided in three modes, where mode 1 offers no protection and mode 3 offers maximum protection. It is possible to move to a higher mode of protection simply by reprogramming the Lock bits. The AVR allows changing "high" bits to "low", but not the other way around. It is not possible to change a "low" Lock bit to a "high", thus lowering the level of protection is not possible. To clear the Lock bits, a complete Chip Erase is required, which erase the Flash memory.

Table 1. Memory Lock Bits

Memory Lock Bits			
Mode	**LB1**	**LB2**	**Protection Type**
1	1	1	Unprogrammed, no protection enabled
2	0	1	Further Programming disabled, Read back possible
3	0	0	Further programming and read back is disabled

In addition to LB1 and LB2, AVR devices with Self-Programming capabilities will have four additional Lock bits: BLB01, BLB02, BLB11, and BLB12. These bits control the restrictions on the use of the LPM and SPM instructions. For more information on the additional protection modes, please use the applicable datasheet.

Fuses

The Lock bits are available in all programming modes. When it comes to fuses, this is no longer the case. Most fuses are available in all programming modes, but some may only be available in certain modes.

Fuse settings are not affected by a Chip Erase. To erase a programmed fuse, you need only program a logic "high" at the correct fuse location.

A changed fuse will take effect after the next Power-on Reset.

Note that setting Lock mode 2 or higher will disable writing to the fuses. Always program Fuses before setting Lock bits.

Table 2 below contains information on which Fuses and Lock bits are available in the different AVR devices, and which programming mode they are accessible in (see legend below the table). For devices not listed, refer to the data sheet for information.

Table 2. Fuse Bits Availability

Device	RCEN	SPIEN	RSTDISBL	FSTRT	BODEN	BODLEVEL	CKSEL n..0	EESAVE	SUT n..0	BOOTRST	BOOTSZ n..0	INTCAP
AT90S1200	P	P	–	–	–	–	–	–	–	–	–	–
AT90S2313	–	P	–	P	–	–	–	–	–	–	–	–
AT90S/LS2323	–	P	–	P/I	–	–	–	–	–	–	–	–
AT90S/LS2343	P/I	P	–	–	–	–	–	–	–	–	–	–
AT90S/LS2333	–	P	–	–	P/I	P/I	P/I	–	–	–	–	–
AT90S/LS4433	–	P	–	–	P/I	P/I	P/I	–	–	–	–	–
AT90S8515	–	P	–	P	–	–	–	–	–	–	–	–
AT90S/LS8535	–	P	–	P/I	–	–	–	–	–	–	–	–
ATtiny11	–	–	H/I	H/I	–	–	H/I	–	–	–	–	–
ATtiny12	–	H/I(1)	H/I(2)	–	H/I	H/I	H/I	–	–	–	–	–
ATtiny15	–	H/I(1)	H/I(2)	–	H/I	H/I	H/I	–	–	–	–	–
ATtiny28	–	–	–	–	–	–	P	–	–	–	–	P
ATmega103	–	P	–	–	–	–	–	P/I	P/I	–	–	–
ATmega161	–	P	–	–	P/I	P/I	P/I	–	–	P/I	–	–
ATmega163	–	P	–	–	P/I	P/I	P/I	–	–	P/I	P/I	–

Notes: 1. Unprogramming this fuse, will disable further ISP programming
2. Programming this fuse will disable further ISP programming
3. Legend:
P – Parallel Programming mode supported.
H – High Voltage Serial Programming (HVSP) mode supported.
I – In-System Programming (ISP) supported.

Table 3. Fuse Description

Fuse	Description
RCEN	By programming this fuse, an internal RC Oscillator is used as the main Master Clock for the MCU. Some AVRs are shipped with this fuse preprogrammed, others not. Check the datasheet to determine the default state of this fuse.
SPIEN	This fuse bit controls the In-System Programming interface. If the fuse is programmed, SPI is allowed. If unprogrammed, the SPI interface is disabled. Note: Even though the ISP functionality is disabled, this will not affect the SPI (Serial Peripheral Interface) interface.
RSTDISBL	By programming this fuse the RESET pin is turned into a general IO pin. On some devices the pin is tuned into an IO pin, and on others into an Output only pin. Please read the applicable datasheet carefully to find the details on a given device.
FSTRT	The Fast Start Fuse controls the startup time for the MCU. If you use a ceramic resonator, or a fast-start Oscillator/clock system, you can program this fuse. This will allow the MCU to start running code quicker.
BODEN	By programming this fuse, the internal Brown-out Protection Circuitry inside the AVR is enabled. This circuitry will monitor the voltage and put the AVR in reset state if brownouts occur.
BODLEVEL	If BOD is enabled, the BODLEVEL Fuse will change the Brown-out voltage and Start-up times. This is device dependent. See the applicable datasheet for details.
CKSEL n..0	The CKSEL Fuses decide the delay period and the clock system that are used. The available clocking modes and Start-up times are device dependent. See the applicable datasheet for details.
EESAVE	By programming this fuse, the content of the EEPROM is not erased during a normal Chip Erase cycle. To erase the EEPROM, you will first need to unprogram this fuse, then do a Chip Erase.
SUT n..0	The SUT Fuses decide the delay period from the External Reset is released (not active anymore) until the Internal Reset is released; the Start-up time. This period should be selected on the basis of what kind of system clock you are using. If you are using an external clock source, a short Start-up time should be sufficient, but if you are using a crystal that needs a long time to be stable, a longer Start-up time is required.
BOOTRST	The Boot Reset Fuse decides if the Reset Vector is placed in the first location of the Application section, or in the Boot section. Programming this fuse will place the Reset Vector in the first location of the boot section, selected by the BOOTSZ Fuses. Other Interrupts are executed from the Application section. Some devices have the possibility of placing the Interrupt Vector table in both Application and Boot sections. The IVSEL bit in the GICR Register decides the location.
BOOTSZ n..0	The Boot Size Fuses select the size and start address of the Boot section. Boot section is only available in devices with SPM instruction.
INTCAP	When this fuse is programmed, the need for external crystal decoupling capacitors is eliminated, thus reducing total system cost. See the datasheet for typical capacitor values.