Experiment – 12: Familiarization with Fuse Bits of ATmega328 under ArduinoUNO Platform

1. Definition

(1) A Fuse Bit, in the context of ATmega328 Microcontroller, can be conceptually visualized as a 'Jumper Wire' which has two conditions:
- **(a)** Unbroken (Unprogrammed; Logic value 1)
- **(b)** Broken (Programmed; Logic value 0)

(2) A Fuse Bit establishes a definite 'Functional Mode' of the microcontroller (MCU). For example: the fuse bit named CKDIV8, when it is programmed, defines that the system clock (clkSYS = clkCPU) will be equal to 'Oscillator Frequency/8', and it is shown in Fig-12. In Fig-12.1, the LL (Logic Low = programmed) value of CKDIV8 closes K2 and opens K1.

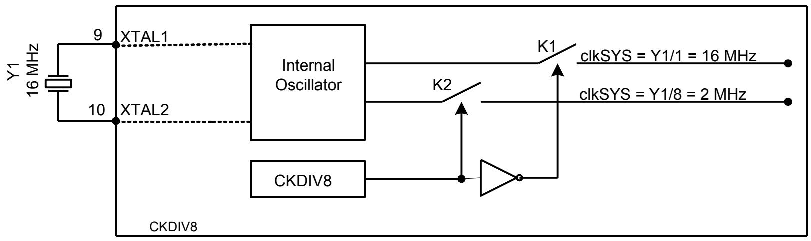

Figure-12.1 *Role of fuse bit CKDIV8 in the determination of system clock frequency*

(3) There are 24 fuse bits inside ATmega328 of which only 19 are active. The value of a fuse bit remains unchanged even in 'Chip Erase' process during Programming Session. These 24 fuse bits are organized as:
(a) Extended Fuse Byte, **(b)** High Fuse Byte, **(c)** Low Fuse Byte

(4) The value of a fuse bit can be altered using Parallel Programmer like TOP2013 and the like. The value can also be changed using Serial Programmer; but, it is not recommended.

(5) The value of a fuse bit must be altered before 'Programming the Lock Bits' of the MCU. Once, the lock bits are programmed, the value of the fuse bit cannot be changed. The option that is left is to erase the chip (chip erase brings the lock bits back to their default values) and then set the fuse bits.

2. Extended Fuse Byte, EF (1 = Unprogrammed, 0 = Programmed)

Bit Position →	7	6	5	4	3	2	1	0
Symbolic Name →	-	-	-	-	-	BODLEVEL2	BODLEVEL1	BODLEVEL0
Default Value →	1	1	1	1	1	1	1	1
Arduino Value →	0	0	0	0	0	1	0	1

3. High Fuse Byte, HF (1 = Unprogrammed, 0 = Programmed)

Bit Position →	7	6	5	4	3	2	1	0
Symbolic Name →	RSTDISBL	DWEN	SPIEN	WDTON	EESAVE	BOOTSZ1	BOOTSZ0	BOOTRST
Default Value →	1	1	0	1	1	0	0	1
Arduino Value →	1	1	0	1	1	0	1	0

(HF7) **RSTDISABL** External Reset Disable

(a) When this fuse bit is not programmed (unprogrammed), the Pin-1 of ATmega328 works as a Reset Pin, and this pin can now receive signal from external hardware device to reset the MCU. Fig-12.2 explains it.

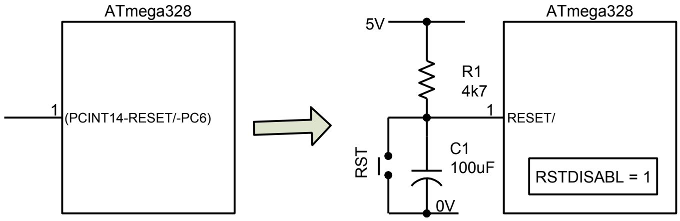

Figure-12.2: *Role of RSTDISABL bit to configure Pin-1 of ATmega328 as Reset/-pin*

(b) If we want to use Pin-1 as a 1-bit digital IO port (PC6) or an interrupt pin (PCINT14), the RSTDISABL bit must be programmed (logic value 0). The choice of option between PC6 and PCINT14 is made by program instruction.

;---

(HF6) **DWEN** debugWire Enable

;---

(HF5) **SPIEN** Enable Serial Program and Data Downloading

(1) The program code/data are written into Flash and EEPROM memory using commercial expensive Parallel ROPM Programmer.

(2) The ATmega328 has built-in circuitry called (In System Programming Interface, ISP or SPI?) by which we can write program code/data into the Flash/EEPROM of the MCU. The setup and procedures are elaborately discussed in the data sheets.

(3) The fuse bit named SPIEN, when programmed (logic value 0) allows a user to engage the ISP Interface for the programming of ATmega328.

;---

(HF4)

(HF3)

;---

(HF2) **BOOTSZ1** Select Boot Size-1

(HF1) **BOOTSZ0** Select Boot Size-0

(HF0) **BOOTRST** Select Reset Vector

(a) After power up reset (after the reception of reset signal at Pin-1), the ATmega328 looks for a particular memory location in the flash memory from which it will begin program execution. This location is known as Boot Location. The size of Flash memory of ATmega328 MCU: 0000h – 3FFFh (16-bit word organized).

(b) The Boot Location is always 0000h if the BOOTRST fuse bit is not programmed (unprogrammed; logic value 1).

(c) If the BOOTRST fuse bit is programmed (logic value 0), then the combinations of BOOTSZ1 and BOOTSZ0 determine four different Boot Locations. These locations are shown below:

(i) (BOOTSZ1, BOOTSZ0, BOOTRST) = (0, 0, 0); Boot Location: 3800h

(ii) (BOOTSZ1, BOOTSZ0, BOOTRST) = (0, 1, 0); Boot Location: 3C00h

(iii) (BOOTSZ1, BOOTSZ0, BOOTRST) = (1, 0, 0); Boot Location: 3E00h *(iv)* (BOOTSZ1, BOOTSZ0, BOOTRST) = (1, 1, 0); Boot Location: 3F00h

;---

4. Low Fuse Byte, LF (1 = Unprogrammed, 0 = Programmed)

Bit Position →	7	6	5	4	3	2	1	0
Symbolic Name →	CKDIV8	CKOUT	SUT1	SUT0	CKSEL3	CKSEL2	CKSEL1	CKSEL0
Default Value →	0	1	1	0	0	0	1	0
Arduino Value →	1	1	1	1	1	1	1	1

(LF7) CKDIV8 Divide clock by 8

If the fuse bit named CKDIV8, when it is programmed, defines that the system clock (clkSYS = clkCPU) will be equal to 'Oscillator Frequency/8', and it is shown in Fig-12.3. In Fig-12.3, the LL (Logic Low = programmed) value of CKDIV8 closes K2 and opens K1.

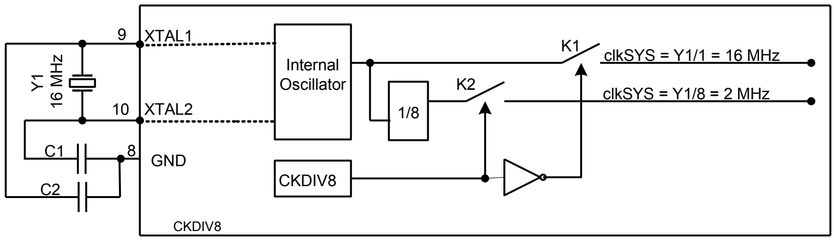

Figure-12.3 *Role of fuse bit CKDIV8 in the determination of system clock frequency*

;---

(LF6) CKOUT Clock output

;---

(LF5) SUT1 Select start-up time-1

(LF4) SUT0 Select start-up time-0

;---

(LF3) CKSEL3 Select Clock source-3

(LF2) CKSEL2 Select Clock source-2

(LF1) CKSEL1 Select Clock source-1

(LF0) CKSEL0 Select Clock source-0

The combinations of the fuse bits (LF3-LF0) determine the source of the system clock of the ATmega328 MCU followed by 'optional divisor 8'. The combinations are:

(a) [CKSEL3:CKSEL0] = [1 1 1 1] : External low power crystal (8 MHz – 16 MHz) as is shown in Fig-12.3. Recommended values for C1 and C2 : 12pF – 22pF.

(b) [CKSEL3:CKSEL0] = [0 0 1 0] : Calibrated internal 8 MHz oscillator (Fig-12.4).

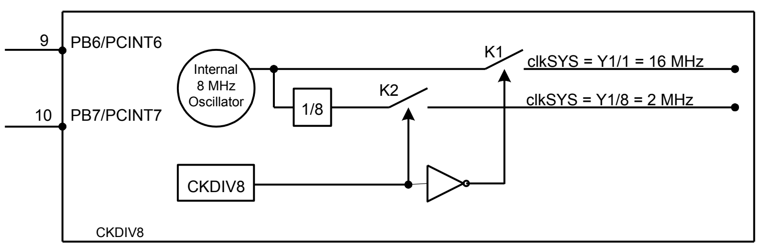

Figure-12.4 *Role of fuse bits (CKSEL3:CKSEL0) in the determination of internal oscillator for clkSYS*