Introduction to clock system of ATmega328p microcontroller

www.xanthium.in

Rahul.Sreedharan

In this tutorial we will configure an ATmega328P microcontroller to work with an external Quartz crystal operating at 11.0592MHz. ATmega328P is a widely used microcontroller developed by ATMEL Corporation(Now Microchip).It is used in the popular Arduino platform (Arduino UNO).

This tutorial deals with configuring ATmega328P for Embedded C development using AVR-GCC as the Cross compiler and AVRDUDE as the programming interface. We do not deal with Arduino Platform directly.

Basic Clock System of ATmega328P

ATmega328P can be clocked by 6 options as shown below.

Table 13-1. Device Clocking Options Select In Fuse Low Byte

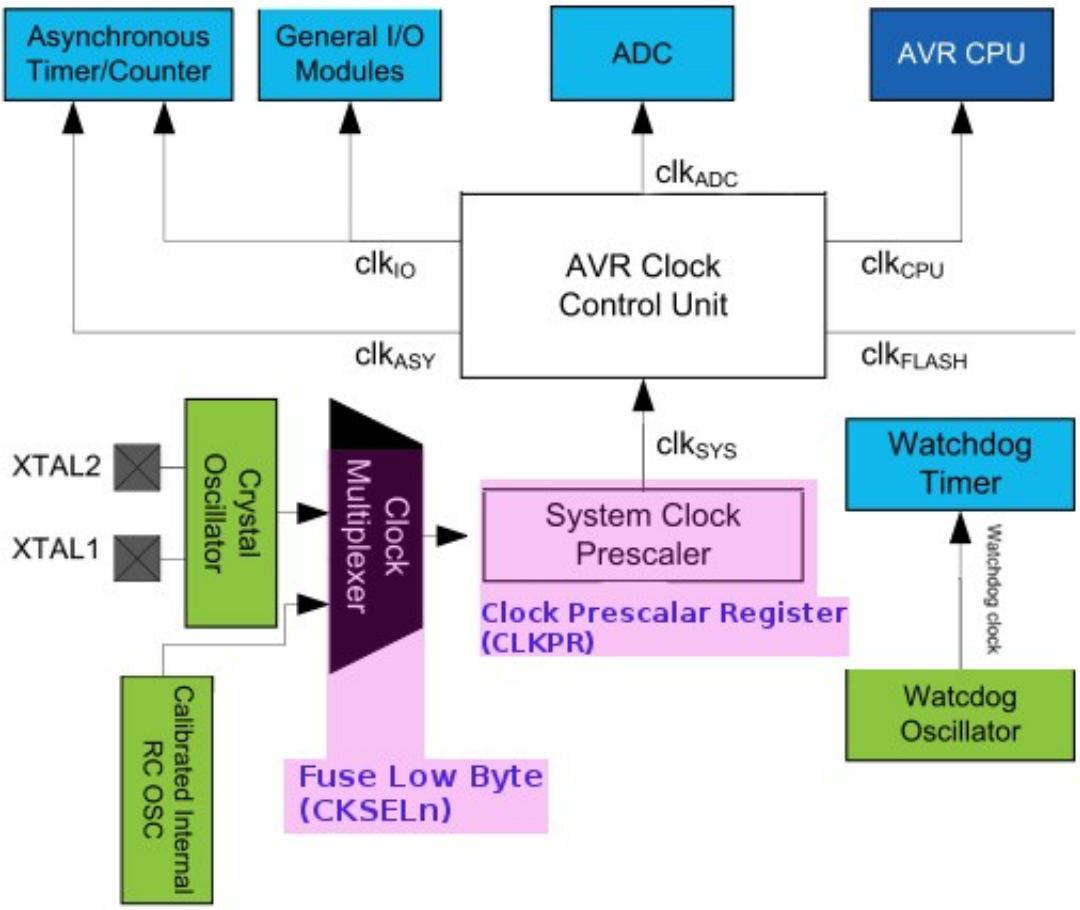
Device Clocking Option	CKSEL[3:0]
Low Power Crystal Oscillator	1111 - 1000
Full Swing Crystal Oscillator	0111 - 0110
Low Frequency Crystal Oscillator	0101 - 0100
Internal 128kHz RC Oscillator	0011
Calibrated Internal RC Oscillator	0010
External Clock	0000
Reserved	0001

Note: For all fuses, '1' means unprogrammed while '0' means programmed.

We are interested in the following clocking options only.

1. Calibrated Internal RC Oscillator (Default Clock Source)
2. Low Power Crystal Oscillator (Quartz Crystal Source)

By default, ATmega328p is clocked by the Calibrated internal RC Oscillator which runs at 8MHz. The 8 MHz clock is divided by 8 due to the setting of the CKDIV8 bit in Fuse Low Byte. **So effective clock rate available is 1 MHz**.



To Configure the Micro controller to use the External Crystal we should make changes to the Fuse Low byte settings. Note that the fuses are read as logical zero, “0”, if they are programmed.

ATmega328 has three Fuse Bytes (8bits)

1. Extended Fuse Byte
2. Fuse High byte
3. **Fuse Low Byte**

Table 31-7. Fuse Low Byte

Low Fuse Byte	Bit No.	Description	Default Value
CKDIV8(4)	7	Divide clock by 8	0 (programmed)
CKOUT(3)	6	Clock output	1 (unprogrammed)
SUT1	5	Select start-up time	1 (unprogrammed)(1)
SUT0	4	Select start-up time	0 (programmed)(1)
CKSEL3	3	Select Clock source	0 (programmed)(2)
CKSEL2	2	Select Clock source	0 (programmed)(2)
CKSEL1	1	Select Clock source	1 (unprogrammed)(2)
CKSEL0	0	Select Clock source	0 (programmed)(2)

Our specification in this case is an 11.0592MHz Crystal with 22pF capacitors for Slow rising power.

Table 13-3. Low Power Crystal Oscillator Operating Modes

Frequency Range [MHz]	CKSEL[3:1](2)	Range for Capacitors C1 and C2 [pF]
0.4 - 0.9	100(3)	–
0.9 - 3.0	101	12 - 22
3.0 - 8.0	110	12 - 22
8.0 - 16.0	111	12 - 22

The above table gives the CKSEL[3:1] vales as 111 but CKSEL0 value is missing. The other values can be found in the Crystal start up times table.

Table 13-4. Start-up Times for the Low Power Crystal Oscillator Clock Selection

Oscillator Source / Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	CKSEL0	SUT[1:0]
Ceramic resonator, fast rising power	258 CK	14CK + 4.1ms(1)	0	00
Ceramic resonator, slowly rising power	258 CK	14CK + 65ms(1)	0	01
Ceramic resonator, BOD enabled	1K CK	14CK(2)	0	10
Ceramic resonator, fast rising power	1K CK	14CK + 4.1ms(2)	0	11
Ceramic resonator, slowly rising power	1K CK	14CK + 65ms(2)	1	00
Crystal Oscillator, BOD enabled	16K CK	14CK	1	01
Crystal Oscillator, fast rising power	16K CK	14CK + 4.1ms	1	10
Crystal Oscillator, slowly rising power	16K CK	14CK + 65ms	1	11

From which the values of **CKSEL0 = 1 and SUT[1:0] =11.** So Low Fuse byte value equals 0xFF.

Low Fuse Byte – Atmega328p

7	6	5	4	3	2	1	0
CKDIV8	CKOUT	SUT1	SUT0	CKSEL3	CKSEL2	CKSEL1	CKSEL0
1	1	1	1	1	1	1	1
No Divison/8	No Clock out	Startup Time		Frequency Range = 8.0 - 16.0 MHz Crystal Capacitor Values = 12 – 22pF			
		14 Clock Cycles + 65mS					
0xFF							

After you have figured out the Low Fuse Byte value as 0xFF,You can program the Fuse bits of ATmega328p using AVRDUDE and a compatible programmer like USBasp or USBtinyISP.

Here we are using USBasp so programming command is as follows.

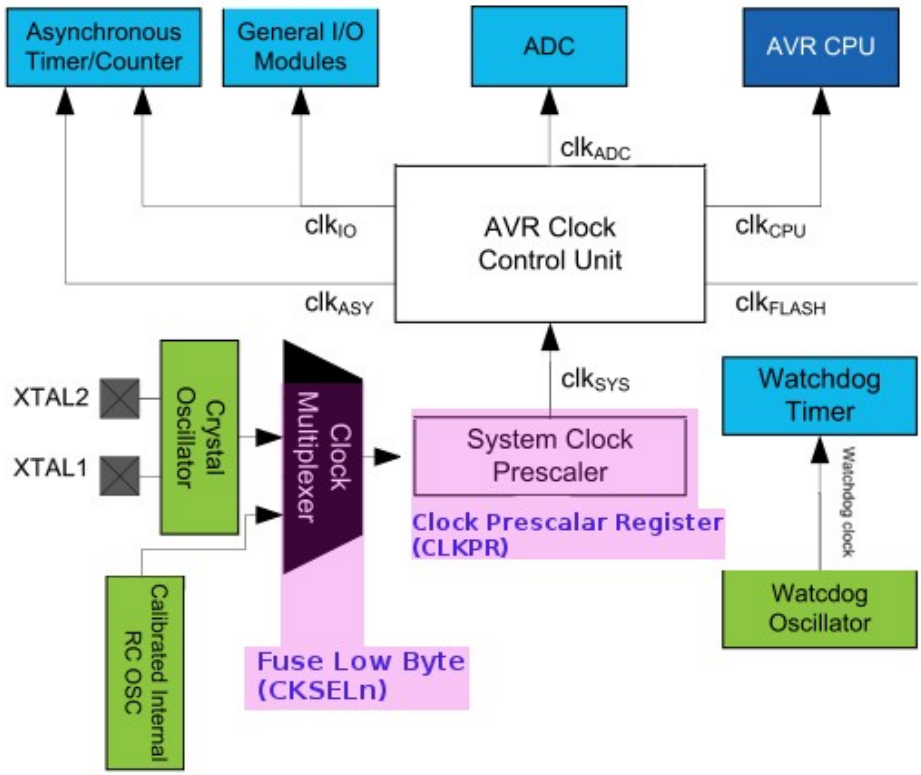
```
avrdude -c usbasp -p m328p -U lfuse:w:0xFF:m
```

System Clock Prescalar of ATmega328

ATmega328 has a System clock Prescalar which can be **used to scale down the clock frequency** by a factor of 2 to 256.

This feature can be used to decrease the system clock frequency and the power consumption when the requirement for processing power is low

It is situated **after the clock multiplexer block** controlled by the Fuse Low byte register as shown in the below figure.



So in our case, The System Clock Prescalar will be a fed a 11.0592MHz signal by the Clock Multiplexer which can be divided by a factor of 2 ,4,8,18,32 up to 256.

The Scaling factor is controlled by writing appropriate values to the **CLKPR** register bits.

13.12.2. Clock Prescaler Register

Name: CLKPR
Offset: 0x61
Reset: Refer to the bit description
Property: -

Bit	7	6	5	4	3	2	1	0
	CLKPCE				CLKPS3	CLKPS2	CLKPS1	CLKPS0
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				x	x	x	x

CLKPS[3:0] bits in the CLKPR control the division factor.

Table 13-17. Clock Prescaler Select

CLKPS[3:0]	Clock Division Factor
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	Reserved

These bits can be written at run time provided a special write procedure is followed.

To avoid unintentional changes of clock frequency, a special write procedure must be followed to change the CLKPS bits:

1. Write the Clock Prescaler Change Enable (CLKPCE) bit to '1' and all other bits in CLKPR to zero: CLKPR=0x80.
2. Within four cycles, write the desired value to CLKPS[3:0] while writing a zero to CLKPCE: CLKPR=0x0N

Interrupts must be disabled when changing prescaler setting to make sure the write procedure is not interrupted.

The clock from the Prescalar is fed to other peripherals like CPU, ADC,IO Modules, Timers etc, **adjusting the clock frequency will affect clocks to all other peripherals.**

In our case we are **dividing the clock by a factor of 1**,
So CPU ,ADC,Timers,USART etc will be clocked using a 11.0592Mhz clock.