# Force Arduino to use 1Mhz clock

From the ATmega328 Datasheet

**9.2.1 Default Clock Source**

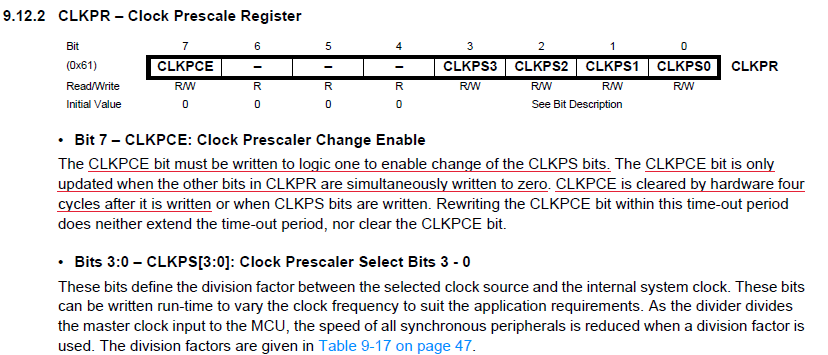
The device is shipped with internal RC oscillator at 8.0MHz and with the fuse CKDIV8 programmed, resulting in 1.0MHz system clock.

This appears to indicate that the Arduino will run at 1MHz, and we do see this in the Simulator. Each Arduino UNO R3 (or similar) may have the initial clock settings set differently, or the programmer used may alter this setting.

A typical scenario will use the RC oscillator at 8.0MHz, but without the divide-by-8, thus each cycle will take 1/8MHz or .125μ microseconds.

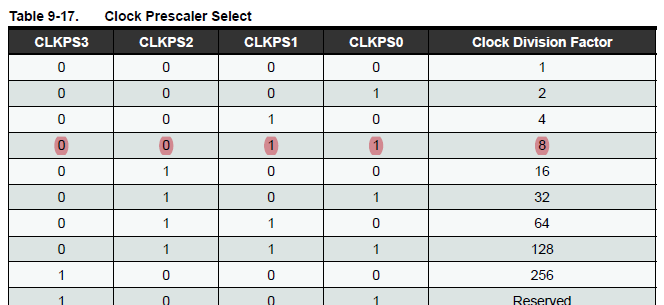
We can force the MCU to use a pre-scaler that will divide the clock by 8 at program startup using instructions that set the Clock Pre-scaler Register CLKPR. The clock pre-scaler is protected by what is called a fuse, and so we need to signal the processor that we want to enable a change prior to it looking for the value we want to set in CLKPR. We enable the change through the same register.

The details for this come from the ATmega328 Datasheet.



The CLKPR register is depicted at address 0x61 in Extended I/O memory. This means we cannot access it with IN/OUT, but rather must use LDS/STS.

Bit-7 is designated as the CLKPC: Clock Prescaler Change Enable. Bits 3 through 0 are used to create a code for one of the possible clock pre-scalers shown in Table 9-17



The highlighted row shows that binary bits 0011 setup a Clock Division Factor of 8.

Putting everything together, we need two distinct operations to set the clock pre-scaler:

1. Enable Clock Pre-Scaler Change CLKPCE by setting CLKPR to 0b10000000
2. Set new Clock Pre-Scaler to DIV8 by setting CLKPR to 0b00000011

The underlined steps in the section on Bit-7 tell us to enable the CLKPCE by setting bit-7 to 1 and all other bits to 0. We set this value in the CLKPR register, and then update CLKPR to a mask matching one of the clock division factors from table 9-17 within the next 4 cycles.

Here is what that looks like:

; force clock to 1mhz

ldi r16,0b10000000 ; CLKPCE mask

sts CLKPR,r16 ; enable clock prescaler change

ldi r16,0b00000011 ; DIV8 mask

sts CLKPR,r16 ; set clock prescaler to DIV8

We setup the Clock Pre-Scaler Change Enable CLKPCE mask with bit-7 on and all other bits off and store this value into the CLKPR register. We have 4 cycles not to set our desired clock division factor. We setup a mask for division-factor 8 where bits 3-0 are 0011 (note bits 7-4 are 0’s). The LDI takes 1 cycle. We then store this value in the CLKPR register with a Store Direct to Data Space STS instruction which takes 2 cycles. The processer picks up the change on the 4th cycle, and we get an 8MHZ / 8 = 1MHZ clock cycle.