ATmega8 Advanced Guide: 8-bit Timers – Tutorial #10

T.K. HAREENDRAN

AVR tutorial

Share this:

Share More

Until now we learned how to connect an LED as an output device, and a button switch as an input device. Besides, we already studied the basic facts about AVR fuses, and the coding procedure to a greater extent. As described earlier, ATmega8 microcontroller has 23 programmable input/output (I/O) pins which can be used for interfacing with external world. It is possible to configure them as input or output by setting a particular register value through coding.

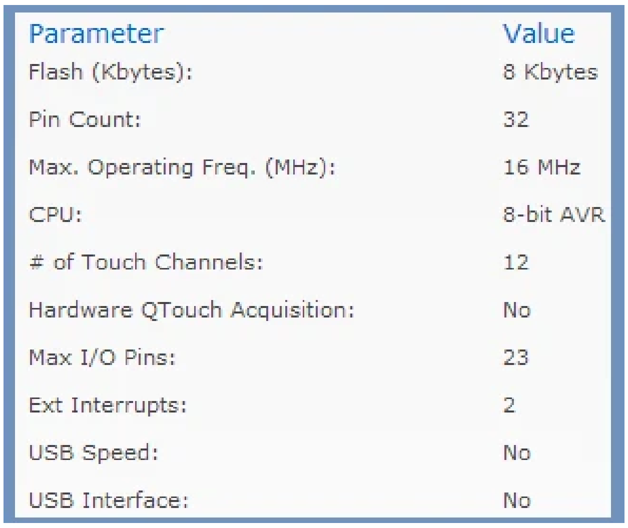
Upcoming parts of this tutorial will cover many advanced tips to interface ATmega8 microcontroller with widely used components and devices like sensors, motors, displays etc. The advanced features of AVR ATmega8 can be used for developing more complex projects than simple blinkers and beepers!

So let us start to learn adavanced features of ATmega8 one by one.

- Timers/Counters
- Pulse Width Modulation (PWM)
- Analog To Digital Conversion (ADC)

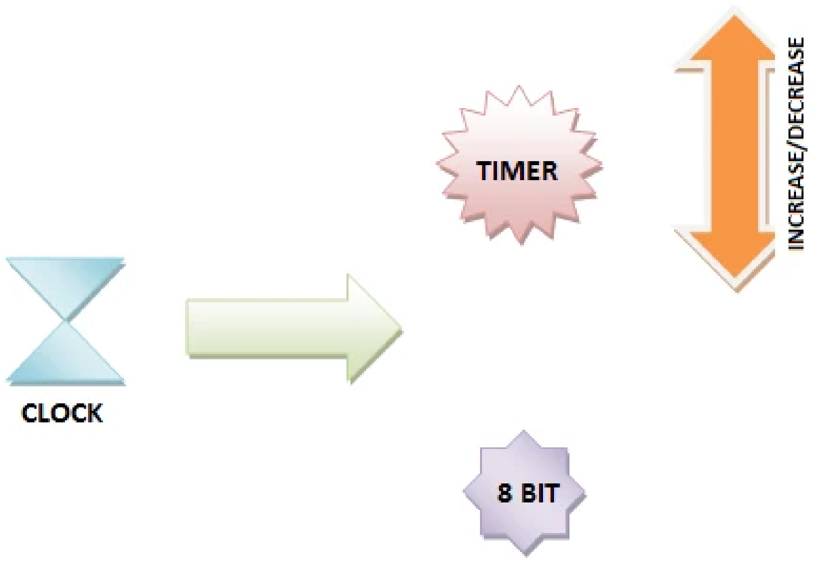
Timers/Counters

The low-power Atmel 8-bit AVR RISC-based microcontroller combines 8KB of programmable flash memory, 1KB of SRAM, 512K EEPROM, and a 6 or 8 channel 10-bit A/D converter. The device supports throughout of 16 MIPS at 16 MHz and operates between 2.7-5.5 volts. Key parameters are:

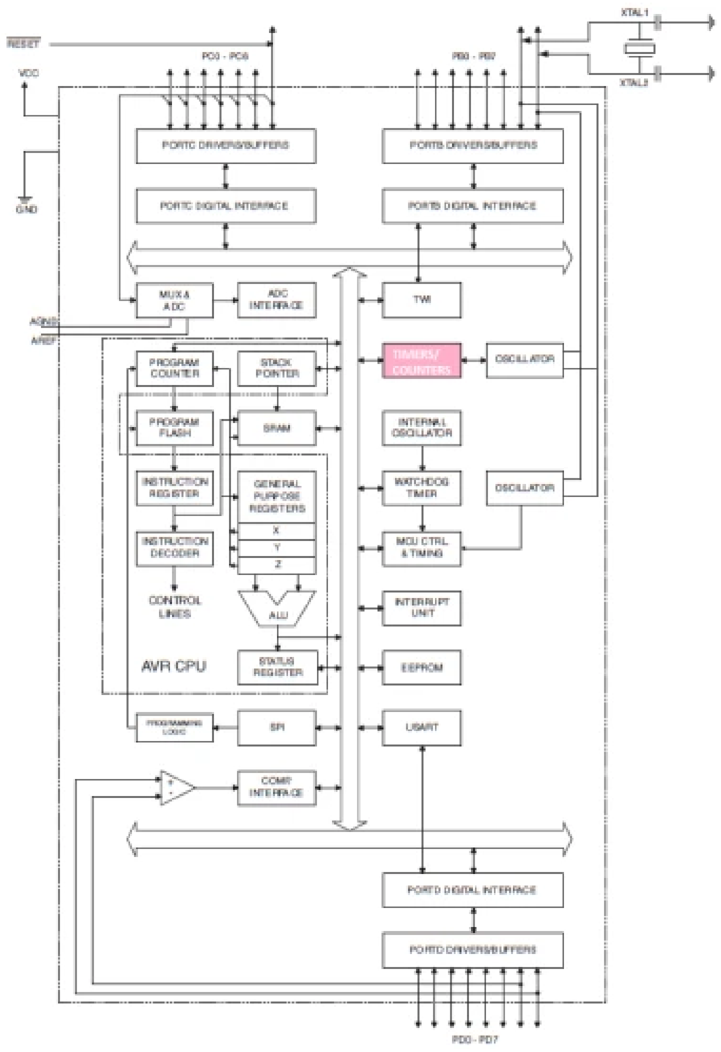


Parameter	Value
Flash (Kbytes):	8 Kbytes
Pin Count:	32
Max. Operating Freq. (MHz):	16 MHz
CPU:	8-bit AVR
# of Touch Channels:	12
Hardware QTouch Acquisition:	No
Max I/O Pins:	23
Ext Interrupts:	2
USB Speed:	No
USB Interface:	No

AVR has powerful and multifunctional timers. Timers are standard features available in almost all microcontrollers. A timer is simply a register of 8 or 16 bit size. This is known as the resolution of the timer (ie 8 bit timer, 16 bit timer).



In an 8 bit timer, the register is 8 bits long and thus can store a number from 0 to 255. Likewise, a 16 bit timer can hold a value between 0 to 65535. This register has a property of increasing or decreasing its value without any intervention by CPU at a rate frequently defined by the user. This frequency (at which the timer register increases/decreases) is known as the Timer Frequency. Note that Timer Frequency can be less than or equal to the CPU clock frequency. Internal block diagram of ATmega8 is shown here for your quick reference.



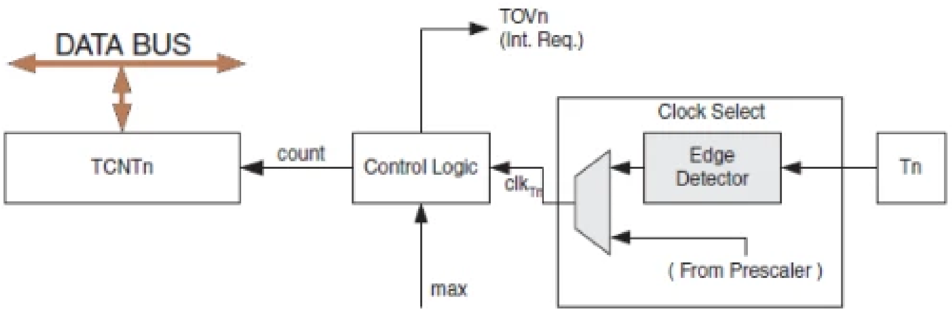
8 Bit Timer

ATmega8 have 3 different timers, of which the simplest one is TIMER0, with an 8 bit (0-255) resolution. The Atmega controllers provide hardware counters. Those counters are registers that are incremented normally by a signal from the oscillator which also drives the Atmega. The oscillator is not connected directly but via a variable prescaler to run slower. To every counter there is at least one compare register . When the counter value equals the value of the compare register, a certain action can be triggered. Actions can also be triggered when the timer value reaches an overflow.

Prescaler: We already marked that the Timer frequency can be less than or equal to the CPU clock frequency. OK, then how we decide or set the timer frequency? Here the "Prescaler" comes in! It is the method of generating the clock frequency for the TIMER from the CPU clock by dividing it with a suitable number.

Overflow: The timer in some conditions automatically take an action or informs the CPU to suspend the current execution, by an Interrupt signal. One example of this is the Timer Overflow, ie an 8 bit timer has counted upto its maximum value (255) and revert to 0.Here, the timer sends a signal or interrupt to the CPU to break its current execution and execute the ISR (interrupt service routine). ISR is a function that the CPU should execute whenever an interrupt occurs. The programmer must write into the ISR to handle the interrupt.

The main part of the 8-bit Timer/Counter is the programmable counter unit. Next figure shows the block diagram of the counter and its surroundings. Just note that the Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the clock select logic which is controlled by the clock select bits located in the Timer/Counter Control Register.



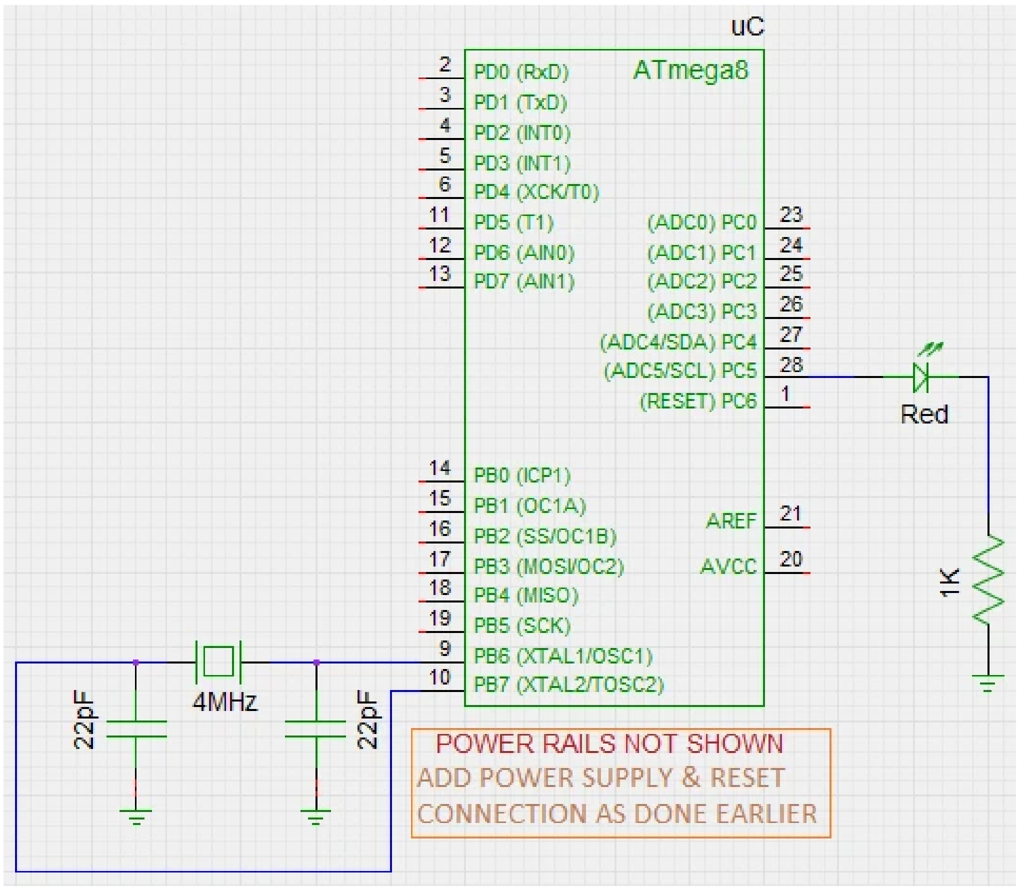
The Timer0 in ATmega8 has some registers, and few of them are introduced here in a simple manner.

- **TCCR0:** this Timer Counter Control Register is used to configure the timer
- **TCNT0:** Timer Counter0 register is the "real" counter in the TIMER0 counter
- **TIMSK:**Timer Interuupt Mask Register, used to activate/deactivate interrupts related to timer
- **TOIE0:** this bit when sets to "1" enables the OVERFLOW interrupt

Yes, today we just started learning all about AVR Timers/Counters. This part is only an introduction to AVR timers. Next will be focused more on 8-bit/16-bit timers and the practical use of timers.

Interested in connecting an external crystal with your Atmega8?

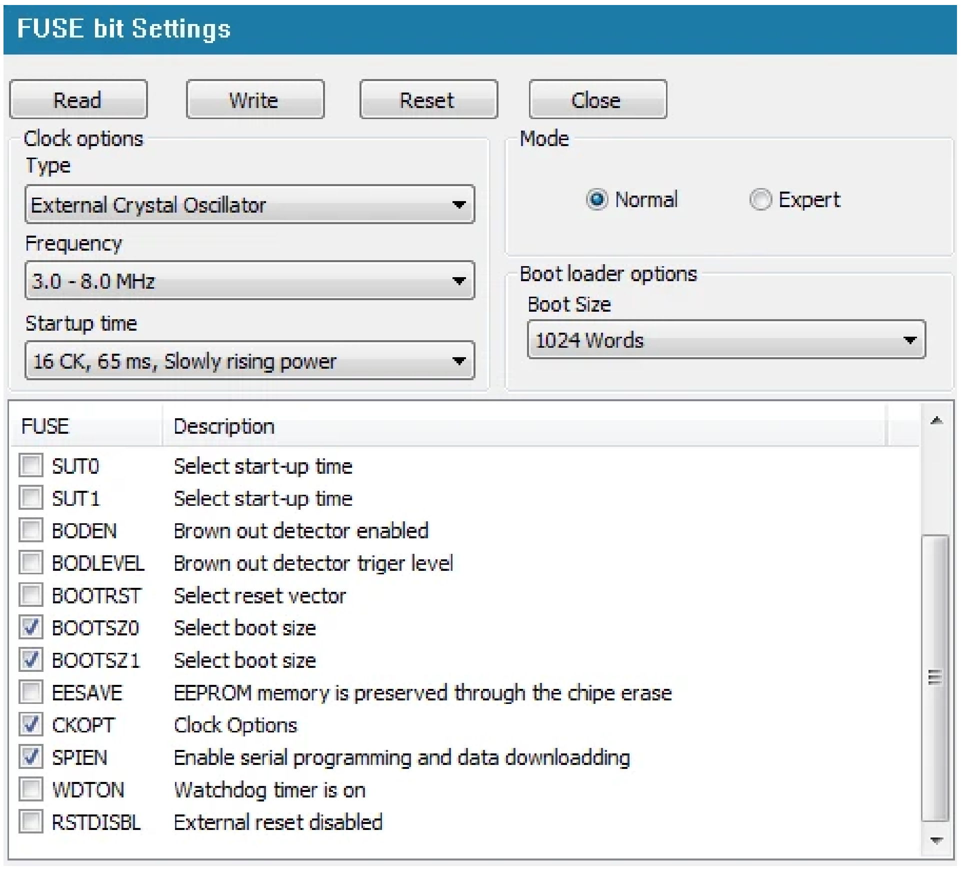
Carefully follow the circuit diagram shown here to complete the hardware setup



Prepare the code. Don’t forget to refer Part 3& 4of this tutorial

```
#define F_CPU 4000000UL
#include
#include
int main(void) {
    DDRC = 0b00100000;
    while(1) {
        PORTC ^= 1 << PORTC5;
        _delay_ms(1000);
    }
    return 0;
}
```

Fuse settings are mandatary now! So use the program (burner software) bundled with your usbasp burner to carry out the task. Refer this screenshot for better understanding of fuse bit settings. Finally burn the hex file into Atmega8. If you run into any difficulties, please inquire on the forum or comment below!

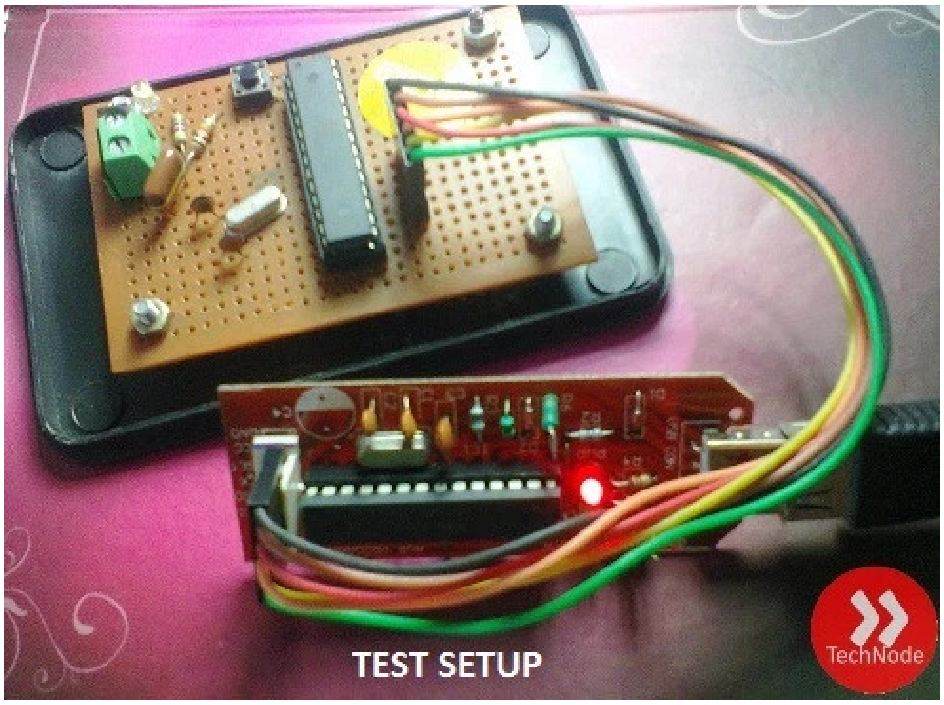

FUSE bit Settings

Clock options
Type: External Crystal Oscillator
Frequency: 3.0 - 8.0 MHz
Startup time: 16 CK, 65 ms, Slowly rising power

Mode: [✓ Normal] Expert

Boot loader options
Boot Size: 1024 Words

FUSE	Description
☐ SUT0	Select start-up time
☐ SUT1	Select start-up time
☐ BODEN	Brown out detector enabled
☐ BODLEVEL	Brown out detector triger level
☐ BOOTRST	Select reset vector
☑ BOOTSZ0	Select boot size
☑ BOOTSZ1	Select boot size
☐ EESAVE	EEPROM memory is preserved through the chipe erase
☑ CKOPT	Clock Options
☑ SPIEN	Enable serial programming and data downloadding
☐ WDTON	Watchdog timer is on
☐ RSTDISBL	External reset disabled



→ Part 11: **8/16 Bit Timers/Counters +**