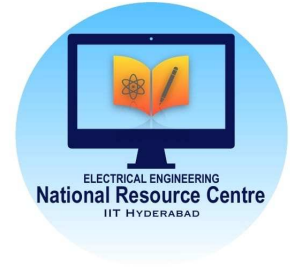




Blink LED through AVR-Assembly



G V V Sharma*

CONTENTS

1	Components	1
2	Blink	1

Abstract—This manual shows how to use the Atmega328p timer to blink the builtin led with a delay.

1 COMPONENTS

Component	Value	Quantity
Arduino	UNO	1

2 BLINK

- 1) Connect the Arduino to the computer and execute the following code

```
wget https://raw.githubusercontent.com/gadepall/arduino/master/assembly/timer/codes/timer.asm
```

- 2) Explain the following instruction

```
sbi DDRB, 5
```

- 3) What do the following instructions do?

```
ldi r16, 0b00000101
out TCCR0B, r16
```

Solution: The system clock (SYSCLK) frequency of the Atmega328p is 16 MHz.

TCCR0B is the Timer Counter Control Register. When

$$TCCR0B = 0b101 \quad (2.1)$$

$$\Rightarrow CLK = \frac{SYSCLK}{1024} \quad (2.2)$$

$$= \frac{16M}{1K} = 16kHz. \quad (2.3)$$

- 4) Explain the PAUSE routine.

```
ldi r19, 0b01000000 ;times to run the loop =
                        64 for 1 second delay
PAUSE: ;this is delay (function)
lp2: ;loop runs 64 times
      IN r16, TIFR0 ;tifr is timer
                        interrupt flag (8 bit timer
                        runs 256 times)
      ldi r17, 0b00000010
      AND r16, r17 ;need second
                        bit
      BREQ PAUSE
      OUT TIFR0, r17 ;set tifr
                        flag high

      dec r19
      brne lp2
      ret
```

Solution: TIFR0 is the timer interrupt flag and TIFR0=0bxxxxxx10 after every 256 cycles. PAUSE routine waits till TIFR0=0bxxxxxx10, this checking is done by the AND and BREQ instructions above.

- 5) Explain the lp2 routine.

Solution: R19 = 64 and is used as a count for lp2. The lp2 routine returns after 64 PAUSE routines.

- 6) What is the blinking delay?

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All content in this manual is released under GNU GPL. Free and open source.

Solution: The blinking delay is given by

$$delay = \frac{CLK}{lp2 \times PAUSE} seconds \quad (2.4)$$

$$= \frac{16 \times 1024}{64 \times 256} seconds = 1second \quad (2.5)$$