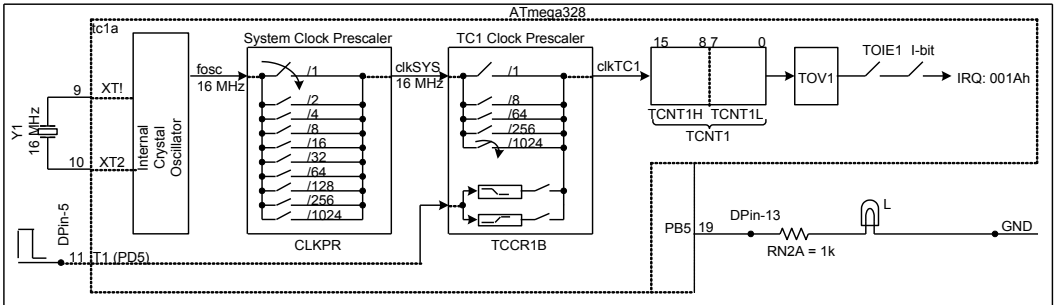


## 5.2 Programming of Time-1(T1) of ATmega328 to Generate 1-sec Time Delay



**Figure-5.2:** Circuit for the demonstration of 1-sec time delay using Timer-1(T1)

- (1) We need to make an arrangement so that the TOV1 flag assumes Logic-H state at every 1-sec interval. This requires that the T1 must count  $n$ -number of clock pulses in 1-sec and then triggers the roll-over event. Let us find the value of  $n$ .
- (2) We take (arbitrarily) division factor  $1/1024$  for the TC1 Clock Prescaler. This gives us:  $\text{clkTC1} = 16 \times 10^6 / 1024 = 15625 \text{ Hz}$ . If the  $\text{clkTC1}$  would turn out with a fractional value, we would go for another division factor and continue so until the  $\text{clkTC1}$  would be an integer value. This is to avoid inaccuracy in the 1-sec time delay.
- (3) From the result of Step-2, we can conclude that the T1 will take 1-sec time to count 15625  $\text{clkTC1}$  pulses.
- (4) Now, if T1 is allowed to count from 0000h, the roll-over will occur once it has finished counting  $\text{FFFFh} + 01\text{h} = 1000\ 0000\text{h}$  (65536) clock pulses. The time delay would appear as: 4.194304-sec.
- (5) But, we have wanted that the roll-over must occur after 1-sec having finished the counting of 15625 pulses. Therefore, the T1 must start counting from a pre-set value (say,  $nI$ ) so that  $nI + 15625 = 65536$ . The value of pre-set parameter ( $nI$ ) is 49911 (C2F7h), which we must load into TCNT1 before the T1 is put into operation.
- (6) Based on the analytical knowledge of Step-1 to 5, we may formulate the following algorithm for 1-sec time delay generation.
  - L1: Initialize everything as needed
  - L1A: Initialize others as needed
    1. Timer-1 mode operation of TC1
    2. Choose  $1/1024$  division factor for TC1 Clock Prescaler
    3. Calculate pre-set value and load it into TCNT1 Register
    4. Clear TOV1 flag
    5. Ensure T1 is OFF
  - ;
  - L1B:
    5. Set direction of PB5-line as output
    6. Ensure OFF state of LED (L)

```

;-----
L1C: 7. Disable Interrupt Structure
      8. Start T1
;-----
L2:  if (TOV1 != HIGH)
      Goto  L2          ; 1-sec has not elapsed
;-----
L3:  ; 1-sec has elapsed
      1. Reset TOV1 flag
      2. Reload pre-set value (there is no auto re-loading in Atmega328 like 8051) into TCNT1
      3. Toggle the state of L (if the LED was ON before, make it OFF now and vice versa)
;-----
L4:  Goto  L2

```

**(7) Assembly Codes for Step-6 (Target MCU: ATmega32 MCU and not ATmega328; External LED is connected at PB0-line.)**

(Assembly language Programming involves the manipulation of registers of the MCU as per instruction set.)

```

      .include  "m32def.inc"
      .org     0x0000
RESET: nop
      rjmp    0x0040

      .org     0x0040          ; beginning of application space
START: nop
L1:   ; initialize stack at 0x9000
      ldi     r16, 0x09
      out     sph, r16
      ldi     r16, 0x00
      out     spl, r16

L1A:  ldi     r16, 0x00
      out     TCCR1A, r16          ; Normal Up-counting Mode Operation of T1

      ldi     r16, 0xF7
      out     TCNT1L, r16
      ldi     r16, 0xC2
      out     TCNT1H, r16          ; pre-set value for 1-sec time delay is loaded into TCNT1

      in      r16, TIFR1
      ori     r16, 0x40
      out     TIFR1, r16          ; TOV1 is cleared

      in      r16, 0x00
      out     TCCR1B, r16          ; T1 is OFF (Stop Condition)

L1B:  sbi     DDRB, DDB0          ; PB0-line of ATmega32 is output
      cli     PORTB, PB0         ; LED is OFF

L1C:  cli
      ; All interrupts are disabled

```

```

        ldi    r16, 0x05
        out   TCCR1B, r16           ; T1 is started with internal clock of clkSYS/1024 = 15625

L2:    in    r16, TIFR1
        ror   r16
        ror   r16
        ror   r16
        ror   r16
        brcc  L2                   ; TOV1 is not LH; 1-sec has not elapsed

L3:    in    r16, TIFR1
        ori   r16, 0x40
        out   TIFR1, r16          ; TOV1 flag is cleared by writing LH at TOV1 flag position

        ldi   r16, 0xF7
        out   TCNT1L, r16
        ldi   r16, 0xC2
        out   TCNT1H, r16        ; pre-set value for 1-sec time delay is loaded into TCNT1

        in    r16, PORTB
        ldi   r17, 0x01          ; 0000 0001
        eor   r16, r17
        out   PORTB, r16        ; Toggle LED via PBO-line of ATmega32

L4:    rjmp  L2                   ; wait for next 1-sec

        .exit
;=====

```

### (8) MCU Registers involved in the Timer-1 Operation

- (a) SREG: Status Register
- (b) TCCR1A : TC1 Control Register – A
- (c) TCCR1B : TC1 Control Register – B
- (d) TCNT1H : TC1 High Byte
- (e) TCNT1L : Tc1 Low Byte
- (f) TIFR1 : TC1 Interrupt Flag Register

### P529 (9) Arduino IDE Codes for Step-6/7

To the extent of my knowledge, the Arduino IDE is not rich in dealing with Timer-1 operation using High Level Commands like the way it deals with LCD. Therefore, in Arduino IDE, we will activate the operations of T1 by manipulating the MCU registers through assignment statements.

```

void setup()
{
    TCCR1A = 0x00;           //L1A: normal mode of operation for Timer-1
    TCCR1B = 0x00;           // T1 is made OFF via TCCR1B register
    TCNT1 = 0xC2F7;         // preload for 1-sec Time Delay; Timer-1 clock = 15625 Hz

    pinMode (13, OUTPUT);    // PB5 as output

```

```

digitalWrite (13, LOW);          // L is OFF
cli();                          // all interrupts of ATmega328 are disabled
TCCR1B = 0x05;                  // T1 is started with internal clock = 16 MHz/1024 = 15625 Hz
}

void loop()
{
    while (bitRead(TIFR1, 2) != HIGH) // checking TOV1 flag for LH
        ;
    bitSet(TIFR1, 2);              // TOV1 flag is cleared
    TCNT1 = 0xC2F7;               // reloading the pre-set count

    digitalWrite(13, !(digitalRead(13))); // Toggle L
}

```