SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

PIC16F877-Timers/Counters

Dr.G.Arthy
Assistant Professor
Department of EEE
SNS College of Engineering

TIMERS

- Used to **measure** the time or generate an accurate time delay.

- Timer is a simple **binary counter** that can be configured to **count clock pulses** (Internal/External).

- Once it reaches the max value, it will roll back to zero, setting up an **OverFlow** flag and generates the interrupt if enabled.

CAN’T A MICROCONTROLLER DO THIS?

WHY TIMER IS REQUIRED?

- The microcontroller can also generate/measure the required time delays by running loops.

- But the timer relieves the CPU from that redundant and repetitive task, allowing it to allocate maximum processing time for other tasks.

TYPES OF TIMERS

PIC16F877a has three timers.

- Timer0 (8-bit timer)
- Timer1 (16-bit timer)→ Good resolution
- Timer2 (8-bit timer)

All Timers can act as a timer or counter or PWM Generation.

PRESCALER

- **Prescaler** is a block that presents inside the timer module and it is used to divide the clock frequency by a constant.

- It allows the timer to be clocked at the rate a user desires.

TIMER INTERRUPT

- As the timer increments and when it reaches its maximum value of 255 (for 8-bit timers) or 65536 (for 16-bit timers), it will trigger an interrupt and initialize itself to 0 back again. This interrupt is called as the Timer Interrupt.

FOSC

- The **FOSC stands for Frequency of the Oscillator**, it is the frequency of the Crystal used. The time taken for the Timer register depends on the value of Prescaler and the value of the FOSC.

TIMER 0

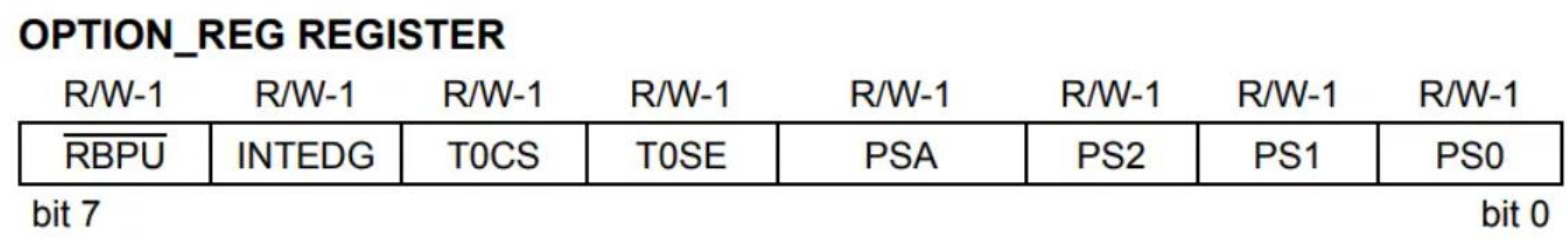
The main features of Timer 0 is given below:

- 8-bit timer/counter with prescaler
- Readable and Writable
- Internal or external Clock set
- Build 8-bit software programmable prescaler
- Edge select for external clock
- Interrupt on overflow from 0XFF to 0X00

REGISTERS IN TIMER 0

- ✓ OPTION_REG
- ✓ TMR0
- ✓ INTCON

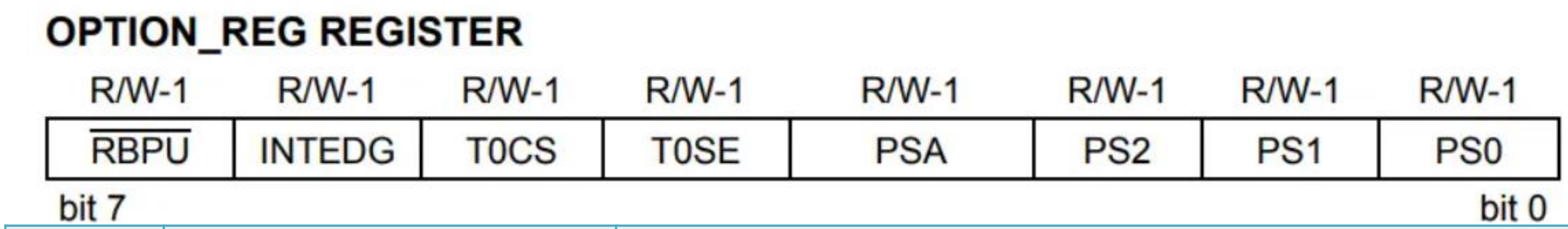
OPTION_REG



OPTION_REG REGISTER

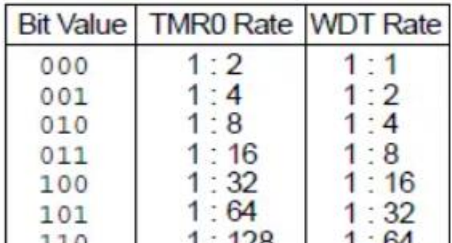
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
-n = Value at POR ‘1’ = Bit is set ‘0’ = Bit is cleared x = Bit is unknown



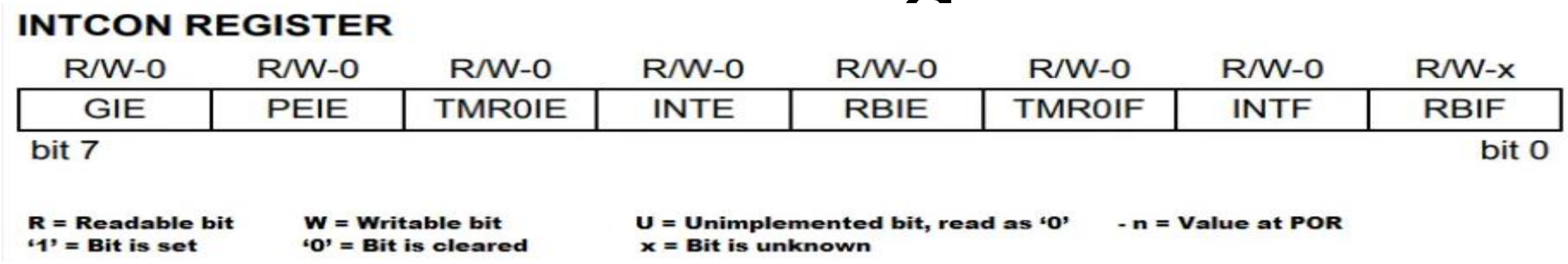
OPTION_REG REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

RBPU	PORTB Pull-up Enable bit	1 = PORTB pull-ups are disabled 0 = PORTB pull-ups are enabled by individual port latch values
INTEDG		
T0CS	TMR0 Clock Source Select bit	1 = Transition on T0CKI pin 0 = Internal instruction cycle clock (CLKO)
T0SE	TMR0 Source Edge Select bit	1 = Increment on high-to-low transition on T0CKI pin 0 = Increment on low-to-high transition on T0CKI pin
PSA	Prescaler Assignment bit	1 = Prescaler is assigned to the WDT 0 = Prescaler is assigned to the Timer0 module
PS2:PS0	Prescaler Rate Select bits	

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64

INTCON Register



INTCON REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

R = Readable bit   W = Writable bit   U = Unimplemented bit, read as '0'   - n = Value at POR
'1' = Bit is set   '0' = Bit is cleared   x = Bit is unknown

GIE	Global Interrupt Enable bit	1-Enables all unmasked interrupts 0-Disables all interrupts
PIE	Peripheral Interrupt Enable bit	1-Enables all unmasked peripheral interrupts 0-Disables all peripheral interrupts
TMR0IE	TMR0 Overflow Interrupt Enable bit	1-Enables the TMR0 interrupt 0-Disables the TMR0 interrupt
INTE	(RB0/INT External Interrupt Enable Bit) Not for Timers	
RBIE	(RB Port Change Interrupt Enable Bit) Not for Timers	1 = Prescaler is assigned to the WDT 0 = Prescaler is assigned to the Timer0 module
TMR0IF	TMR0 Overflow Interrupt Flag bit	1-TMR0 register has overflowed (must be cleared in software) 0-TMR0 register did not overflow
INTF & RBIF	(RB0/INT External Interrupt Flag Bit) (RB Port Change Interrupt Flag Bit)	Not for Timers

TMR0 Register

- This is the 8-bit register that holds the timer values.

- For example,
  - Initially, it will be 0. It will increment by one per one clock cycle. When it reaches 255, it will trigger the TMR0IF bit in INTCON Register. Then again starts from 0.

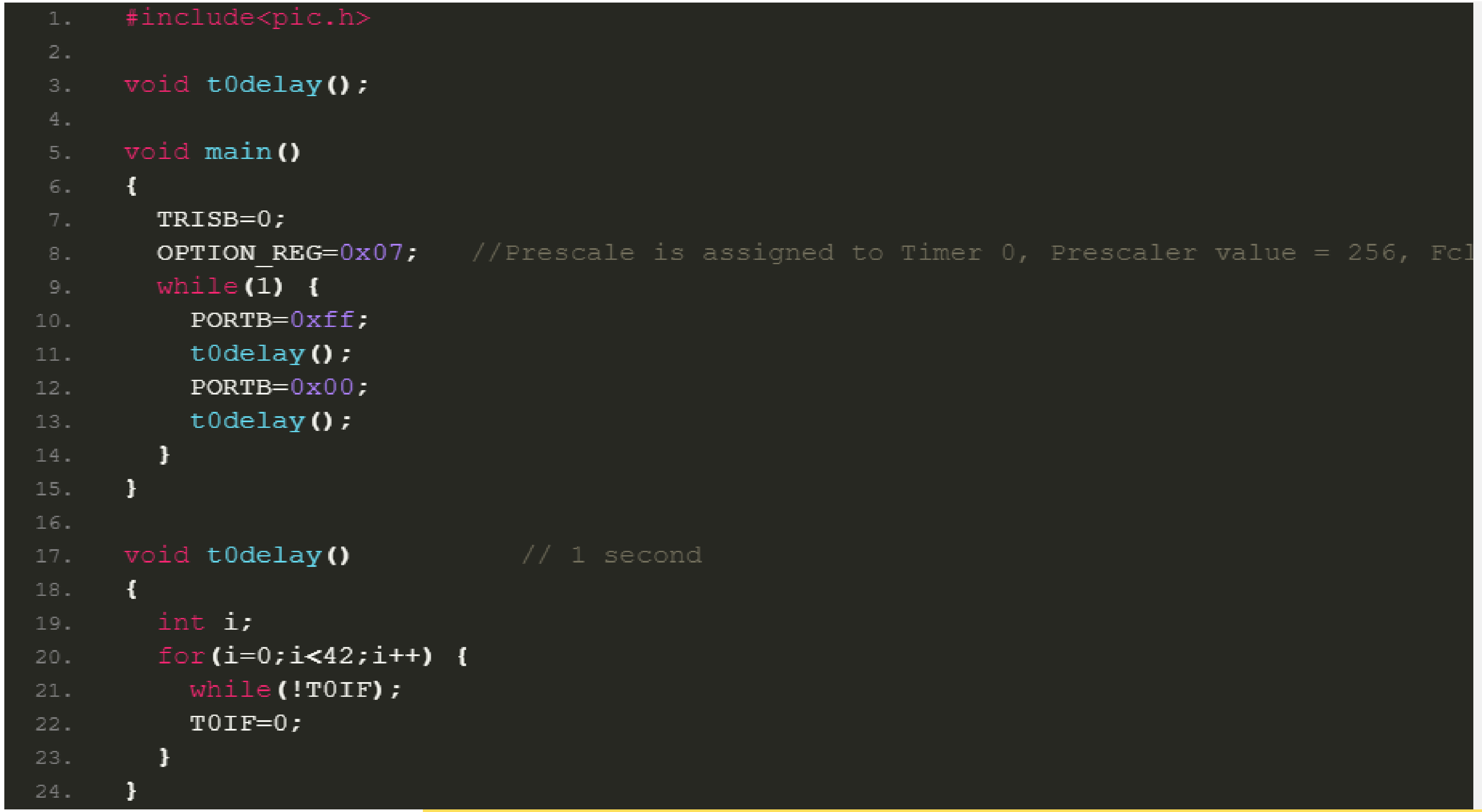
Delay Calculation for 1 second

$$\boxed{fout = \frac{fclk}{4 * \text{Prescaler} * (256 - \text{TMR0}) * \text{Count}}} \quad where \quad \boxed{Tout = \frac{1}{fout}}$$

Here, My fclk = 11.0592MHz (You can put your board’s fclk)
Prescaler = 256 (It is based on PS0 – PS2 bits in OPTION_REG)
TMR0 = 0. (My TMR0’s value will be 0)
Desire Delay (Tout = 1 second) So Fout = 1 (Tout = 1/Fout)
Apply these values to the above formula.
Count = 11059200 / (4*256*256*1)
Count = 42.1875 (approximately 42).

Timer0 Code

In this code, a LED is connected to Port B. Those LEDs are blinking every 1 second.



```c
#include<pic.h>

void t0delay();

void main()
{
  TRISB=0;
  OPTION_REG=0x07;   //Prescale is assigned to Timer 0, Prescaler value = 256, Fcl
  while(1) {
    PORTB=0xff;
    t0delay();
    PORTB=0x00;
    t0delay();
  }
}

void t0delay()           // 1 second
{
  int i;
  for(i=0;i<42;i++) {
    while(!T0IF);
    T0IF=0;
  }
}
```

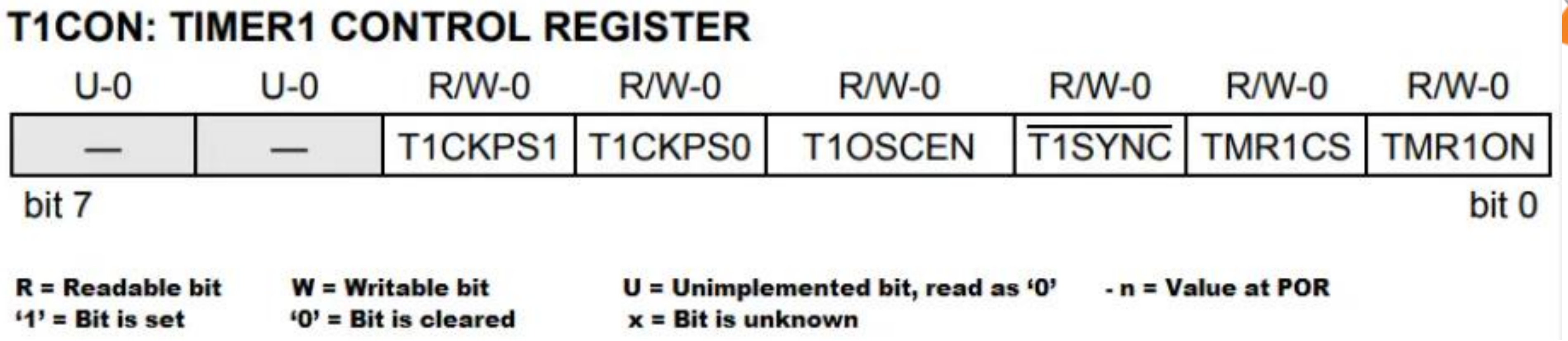
Timer 1

The timer TMR1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter with two 8-Bit registers TMR1H/TMR1L
- Readable and writable
- software programmable Prescaler up to 1:8
- Internal or external clock select
- Interrupt on overflow from FFFFh to 00h
- Edge select for external clock

Registers used for Timer1

- T1CON
- TMR1 (TMRIH, TMRIL)
- PIR1



T1CON: TIMER1 CONTROL REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

R = Readable bit **W = Writable bit** **U = Unimplemented bit, read as ‘0’** **- n = Value at POR**
‘1’ = Bit is set **‘0’ = Bit is cleared** **x = Bit is unknown**

T1CKPS1:T1CKPS0: Timer1 Input Clock Prescale Select bits
T1OSCEN: Timer1 Oscillator Enable Control bit
T1SYNC: Timer1 External Clock Input Synchronization Control bit
TMR1CS: Timer1 Clock Source Select bit
TMR1ON: Timer1 On bit

TMR1 Register

- Timer1 has a register called the TMR1 register, which is 16 bits in size. Actually, the TMR1 consists of two 8-bits registers:

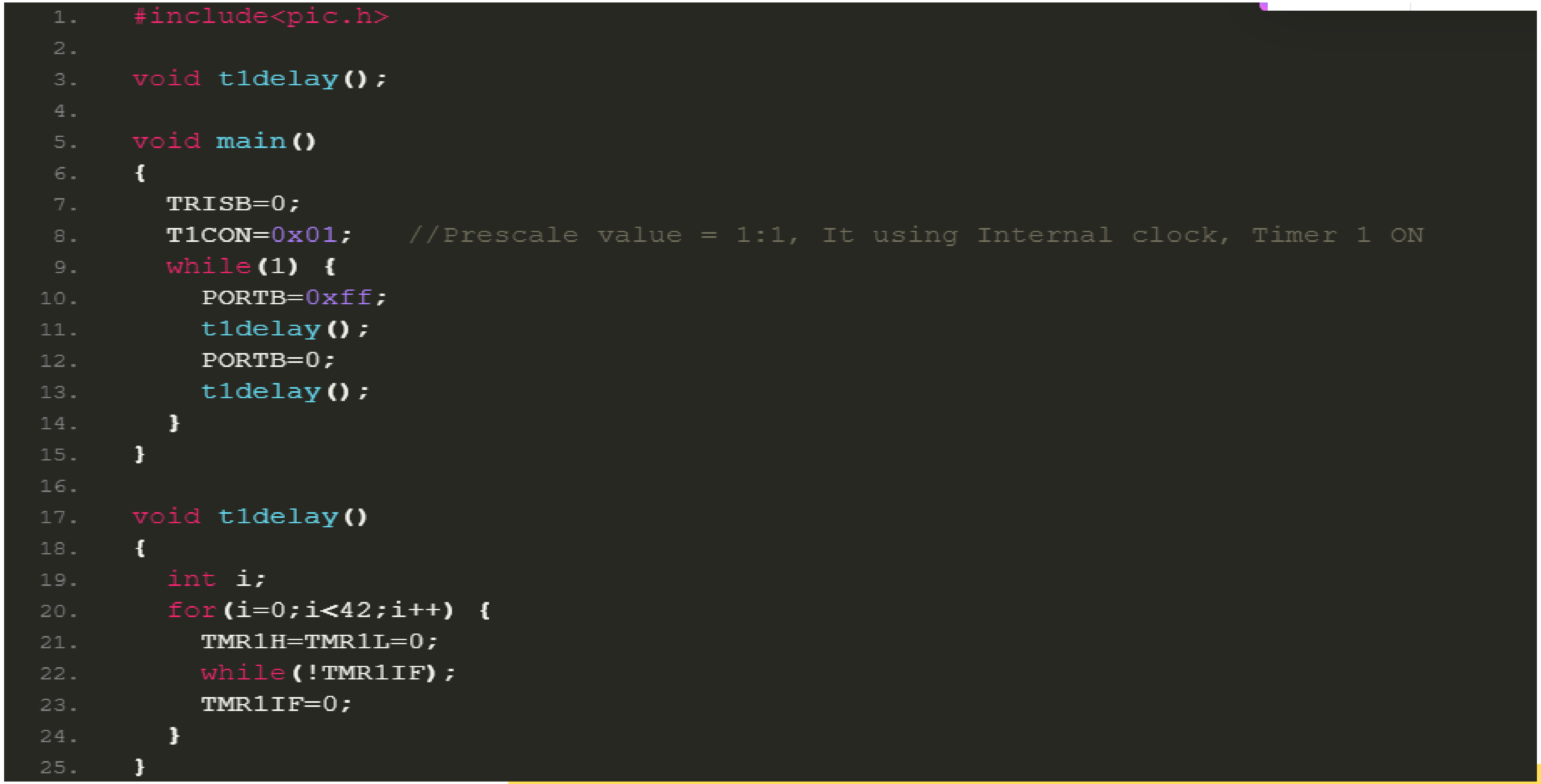
- TMR1H
- TMR1L

PIR1 Register

- This flag marks the end of ONE cycle count. The flag needs to be reset in the software if you want to do another cycle count.

Timer1 Code

- In this code, a LED is connected to Port B.
- Those LEDs are blinking every 1 second.



```c
#include<pic.h>

void t1delay();

void main()
{
  TRISB=0;
  T1CON=0x01;   //Prescale value = 1:1, It using Internal clock, Timer 1 ON
  while(1) {
    PORTB=0xff;
    t1delay();
    PORTB=0;
    t1delay();
  }
}

void t1delay()
{
  int i;
  for(i=0;i<42;i++) {
    TMR1H=TMR1L=0;
    while(!TMR1IF);
    TMR1IF=0;
  }
}
```

Timer 2

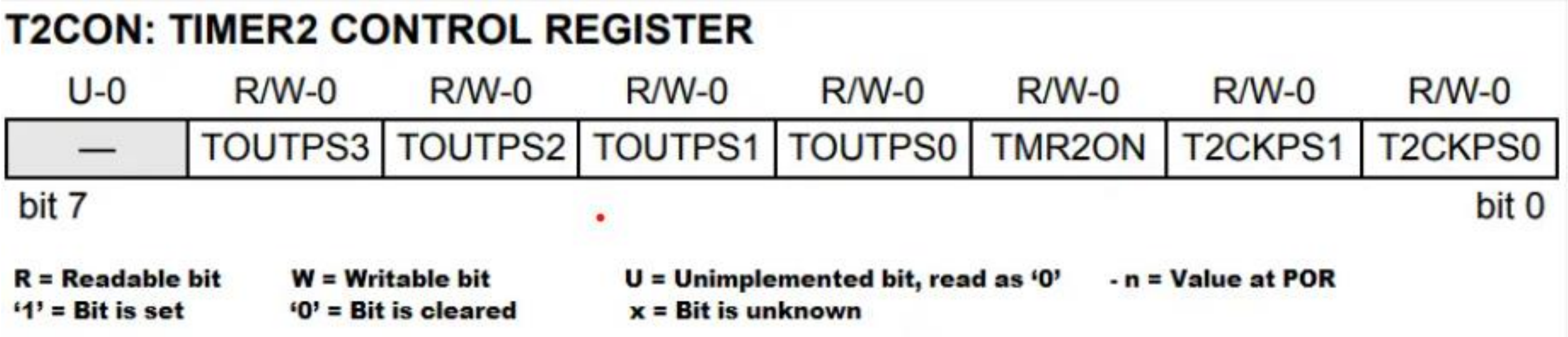
- The TImer2 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter
- Readable and writable
- Software programmable Prescaler/PostScaler up to 1:16
- Interrupt on overflow from FFh to 00h

Registers used for Timer2

- T2CON
- TMR2
- PIR2
- PR2

T2CON Register



T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

R = Readable bit **W = Writable bit** **U = Unimplemented bit, read as ‘0’** **- n = Value at POR**
‘1’ = Bit is set **‘0’ = Bit is cleared** **x = Bit is unknown**

TOUTPS3:TOUTPS0: Timer2 Output Postscale Select bits

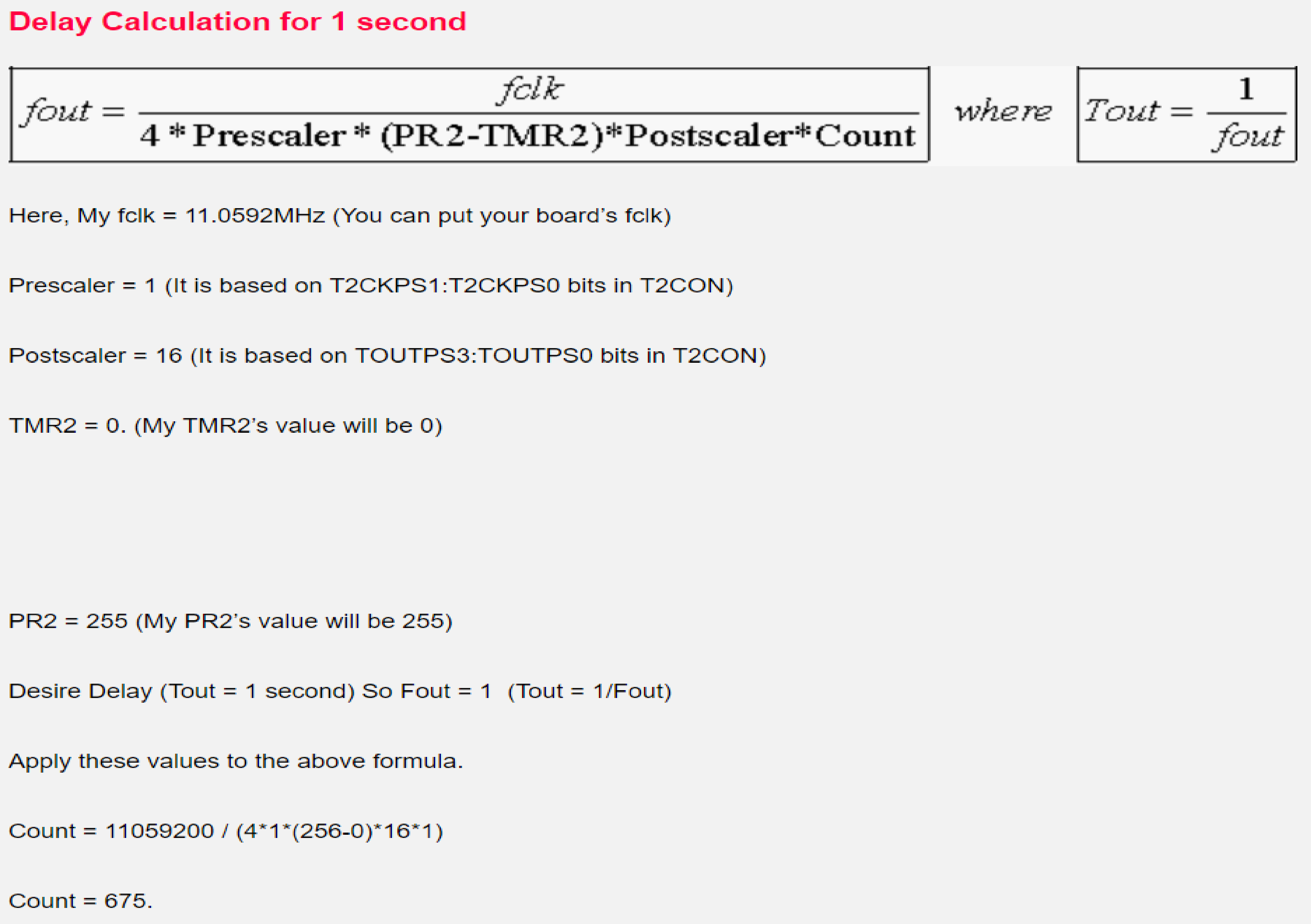
TMR2ON: Timer2 On bit

T2CKPS1:T2CKPS0: Timer2 Clock Prescale Select bits

TMR2 & PR2 Register

- **TMR2** – The register in which the “initial” count value is written.

- **PR2** – The register in which the final or the maximum count value is written.



Delay Calculation for 1 second

$$fout = \frac{fclk}{4 * \text{Prescaler} * (\text{PR2-TMR2}) * \text{Postscaler} * \text{Count}} \quad where \quad Tout = \frac{1}{fout}$$

Here, My fclk = 11.0592MHz (You can put your board’s fclk)

Prescaler = 1 (It is based on T2CKPS1:T2CKPS0 bits in T2CON)

Postscaler = 16 (It is based on TOUTPS3:TOUTPS0 bits in T2CON)

TMR2 = 0. (My TMR2’s value will be 0)

PR2 = 255 (My PR2’s value will be 255)

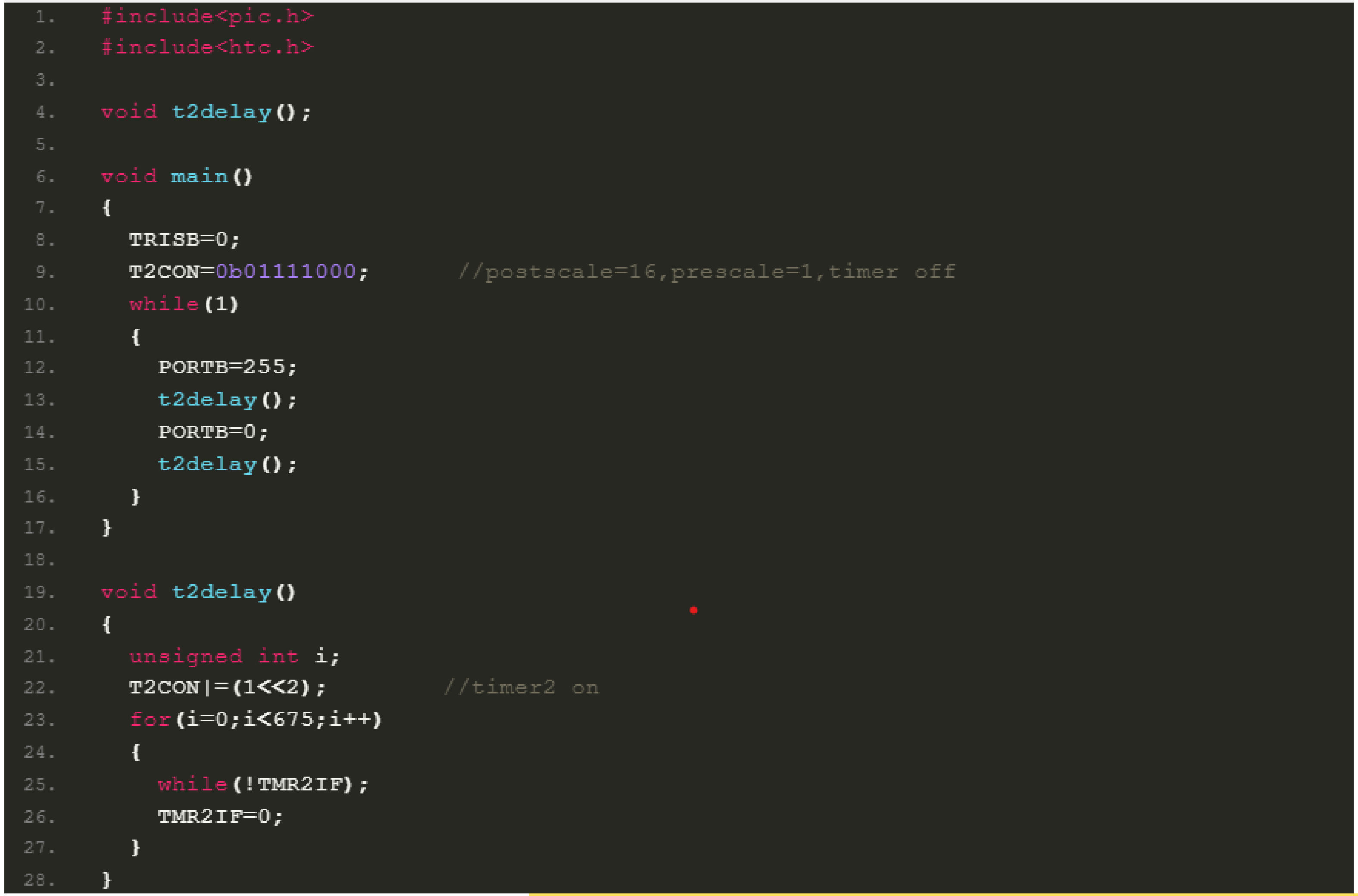
Desire Delay (Tout = 1 second) So Fout = 1 (Tout = 1/Fout)

Apply these values to the above formula.

Count = 11059200 / (4*1*(256-0)*16*1)

Count = 675.

Timer2 Code



```c
#include<pic.h>
#include<htc.h>

void t2delay();

void main()
{
  TRISB=0;
  T2CON=0b01111000;      //postscale=16,prescale=1,timer off
  while(1)
  {
    PORTB=255;
    t2delay();
    PORTB=0;
    t2delay();
  }
}

void t2delay()
{
  unsigned int i;
  T2CON|=(1<<2);         //timer2 on
  for(i=0;i<675;i++)
  {
    while(!TMR2IF);
    TMR2IF=0;
  }
}
```

Assessment

1. Mention the registers in Timer 2.

____

2. List the registers in Timer 0.

____