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 ELECTRONICS AND COMMUNICATION ENGINEERING

Sub: Microcontroller Programming And Interfacing
Subcode:23ECB202
Unit-IV

PIC PROGRAMMING IN C: TIMER, SERIAL PORT AND INTERRUPT/

Programming Timers 0,1, 2 and 3 in C

Programming Timers 0,1, 2 and 3 in C

- Measure the time or generate an accurate time delay. The microcontroller can also generate and measure the required time delays by running loops. Still, the timer relieves the CPU from that redundant and repetitive task, allowing it to allocate maximum processing time for other tasks.

- The timer is nothing but 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 generating the interrupt if enabled.

PIC16F877A Timers

- PIC16F877A has three timers.
  - Timer0 (8-bit timer)
  - Timer1 (16-bit timer)
  - Timer2 (8-bit timer)

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

- To start using a timer, we should understand **8-bit/16-bit timers, prescalers, Timer interrupts, and Focs.**
  - As said earlier, there are both the 8-bit and 16-bit Timers in our PIC16F877A. The main difference between them is that the 16-bit Timer has a much better Resolution than the 8-bit Timer.

Prescalers is a name for the part of a microcontroller that divides the oscillator clock before it reaches logic that increases the timer status.

- The range of the Prescalers ID is from 1 to 256, and the value of the Prescalers can be set using the OPTION Register

PIC16F877A Timers

To start using a timer we should understand some of the fancy terms like **8-bit/16-bit timer, Prescaler, Timer interrupts, and Focs.** Now, let us see what each one really means.

As said earlier, there are both the 8-bit and 16-bit Timers in our PIC16F877A. The main difference between them is that the 16-bit Timer has a much better Resolution than the 8-bit Timer.

Prescaler is a name for the part of a microcontroller that divides the oscillator clock before it reaches logic that increases timer status.

- The range of the Prescaler ID is from 1 to 256 and the value of the Prescaler can be set using the OPTION Register

Timer 0

- The Timer0 module timer/counter has the following features:
- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable Prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Registers used for Timer0

1. OPTION_REG

We perform all the necessary settings with OPTION_REG Register. The size of the register is 8 bits.

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

RBPU: PORTB Pull-up Enable bit (This bit is not used for timers)

1= PORTB pulls-ups are disabled

0= PORTB pulls-ups are enabled by individual port latch values

INTEDG (This bit is not used for timers)

T0CS: TMR0 Clock Source Select bit

1= Transition on T0CK1 pin

0= Internal instruction cycle clock (CLKO)

OPTION_REG

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
111	1 : 256	1 : 128

2. 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: Not for Timers

RBIE: Not for Timers

TMR0IF: TMR0 Overflow Interrupt Flag bit

1-TMR0 register has overflowed (must be cleared in software)
0-TMR0 register did not overflow

INTF: Not for Timers

RBIF: Not for Timers

3. 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**

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

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)

Delay Calculation for 1 second

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).

Timer 0 -Example Code

```
#include<pic.h>

void t0delay();

void main()
{
  TRISB=0;
  OPTION_REG=0x07;   //Prescale is assigned to Timer 0, Prescaler value = 256, Fclk = 11.0592MHz
  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 (TMR
- PIR1

T1CON

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
11 = 1:8 prescale value
10 = 1:4 prescale value
01 = 1:2 prescale value
00 = 1:1 prescale value
T1OSCEN: Timer1 Oscillator Enable Control bit
1-Oscillator is enabled
0-Oscillator is shut-off

- T1CON
- TMR1 (TMR
- PIR1

Registers used for Timer1

T1SYNC: Timer1 External Clock Input Synchronization Control bit

1-Does not synchronize external clock input
0-Synchronize external clock input

TMR1CS: Timer1 Clock Source Select bit

- 1-External clock from pin RC0/T1OSO/T1CKI (on the rising edge)
  0-Internal clock (FOSC/4)

TMR1ON: Timer1 On bit

1-Enables Timer1
0-Stops Timer1

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

  It increments from 0000h to the maximum value of 0xFFFFh (or 65,535 decimal). The TMR1 interrupt, if enabled, is generated on overflow which is latched in the interrupt flag bit, TMR1IF (PIR1<0>).

- This interrupt can be enabled/disabled by setting/clearing the TMR1 interrupt enable bit, TMR1IE (PIE1<0>). You can initialize the value of this register to whatever you want (not necessarily “0”).