



# TM57 Series

## TM57ME20

DEMO CODE

FOR BASIC FUNCTIONS

## Application Note

**Tenx reserves the right to change or  
discontinue this product without notice.**

**tenx technology inc.**

---

# CONTENTS

<b>PRODUCT NAME</b> .....	<b>3</b>
<b>TM57 series IC</b> .....	<b>3</b>
<b>TITLE</b> .....	<b>3</b>
<b>TM57PE12_ DEMO CODE FOR BASIC FUNCTIONS</b> .....	<b>3</b>
<b>APPLICATION NOTE</b> .....	<b>3</b>
<b>01. Instruction for using the analog comparator (with external input reference voltage)</b> .....	<b>3</b>
1. To get the detail of the DEMO program, please refer to <b>TM57ME20_CMP_eRef.ASM</b> .....	<b>3</b>
2. Setup (4MHz internal RC clock) .....	<b>3</b>
3. Circuit Diagram .....	<b>5</b>
<b>02. Instruction for using the analog comparator (with internal input reference voltage)</b> .....	<b>6</b>
1. To get the detail of the DEMO program, please refer to <b>TM57ME20_CMP_iRef.ASM</b> .....	<b>6</b>
2. Setup (4MHz internal RC clock) .....	<b>6</b>
3. Internal Input Circuit Diagram .....	<b>8</b>
<b>03. Instruction for using dual clocks (switching between the fast and slow clocks)</b> .....	<b>9</b>
1. To get the detail of the DEMO program, please refer to <b>TM57ME20_Dual_clk.ASM</b> .....	<b>9</b>
2. Setting up (4MHz internal RC as the fast clock, 32.768KHz external crystal oscillator as the slow clock) .....	<b>9</b>
3. Circuit Diagram .....	<b>11</b>
<b>04. Instruction for enabling external interrupts (INT0 and INT1)</b> .....	<b>12</b>
1. To get the detail of the DEMO program, please refer to <b>TM57ME20_extIWK.ASM</b> .....	<b>12</b>
2. Setup (4MHz internal RC clock) .....	<b>12</b>
3. Circuit Diagram .....	<b>13</b>
<b>05. Instruction for using PWM function (PWM0 and PWM1)</b> .....	<b>14</b>
1. To get the detail of the DEMO program, please refer to <b>TM57ME20_PWM.ASM</b> .....	<b>14</b>
2. Setup (4MHz internal RC clock) .....	<b>14</b>
3. Circuit Diagram .....	<b>16</b>
<b>06. Instruction for using TIMER0 function</b> .....	<b>17</b>

- 1. To get the detail of the DEMO program, please refer to  
TM57ME20\_TM0.ASM .....17
- 2. Setup (4MHz internal RC clock) .....17
- 3. Circuit Diagram.....18
- 07. Instruction for using TIMER1 function .....19
  - 1. To get the detail of the DEMO program, please refer  
to TM57ME20\_TM1.ASM .....19
  - 2. Setup (4MHz internal RC clock) .....19
  - 3. Circuit Diagram.....20
- 08. Instruction for using TIMER2 function .....21
  - 1. To get the detail of the DEMO program, please refer to  
TM57ME20\_TM2.ASM .....21
  - 2. Setup (4MHz internal RC clock) .....21
  - 3. Circuit Diagram.....22
- 09. Instruction for using the function of waking up clock.....23
  - 1. To get the detail of the DEMO program, please refer to  
TM57ME20\_WKT.ASM .....23
  - 2. Setup (4MHz internal RC clock) .....23
  - 3. Circuit Diagram.....25
- 10. Instruction for using port B port function.....26
  - 1. To get the detail of the DEMO program, please refer to  
TM57ME20\_WKUP.ASM .....26
  - 2. Setup (4MHz internal RC clock) .....26
  - 3. Circuit Diagram.....27

**PRODUCT NAME**

TM57 series IC

**TITLE**

TM57PE12\_ DEMO CODE FOR BASIC FUNCTIONS

**APPLICATION NOTE**

**01.** Instruction for using the analog comparator ( with external input reference voltage )

1. To get the detail of the DEMO program, please refer to

TM57ME20\_CMP\_eRef.ASM

2. Setup (4MHz internal RC clock)

1) Set PC0/CP0Vi, PC1/CP0Vr, PC3/CP1Vi and PC4/CP1Vr to Analog input. For example:

```

movlw    CP1Vr_ana    ;B'00000000'
        +CP1Vi_ana    ;B'00000000'
        +CP0Vr_ana    ;B'00000000'
        +CP0Vi_ana    ;B'00000000'
movwr    PIE

```

The address of PIE is R PLANE 12H.

2) Set PC0, PC1, PC3 and PC4 as input, and PC2 and PC5 output. For example:

```

movlw    b'11100100'
movwr    PCE
movlw    b'00011011'
movwf    PCD

```

The address of PCE is R PLANE 07H, and the address of PCD is F PLANE 07H.

- 3) After setting up as above, the following actions also need to be performed:
  - a) Set the trigger edge of the interrupt event to the falling edge
  - b) Disable access to the internal reference voltage (VREN)
  - c) Enable CMP1
  - d) Enable CMP0
  - e) Enable CMP1 output
  - f) Enable CMP0 output
  - g) Use external input as CMP1 reference voltage
  - h) Use external input as CMP0 reference voltage

The instructions to execute the procedure are shown as follows:

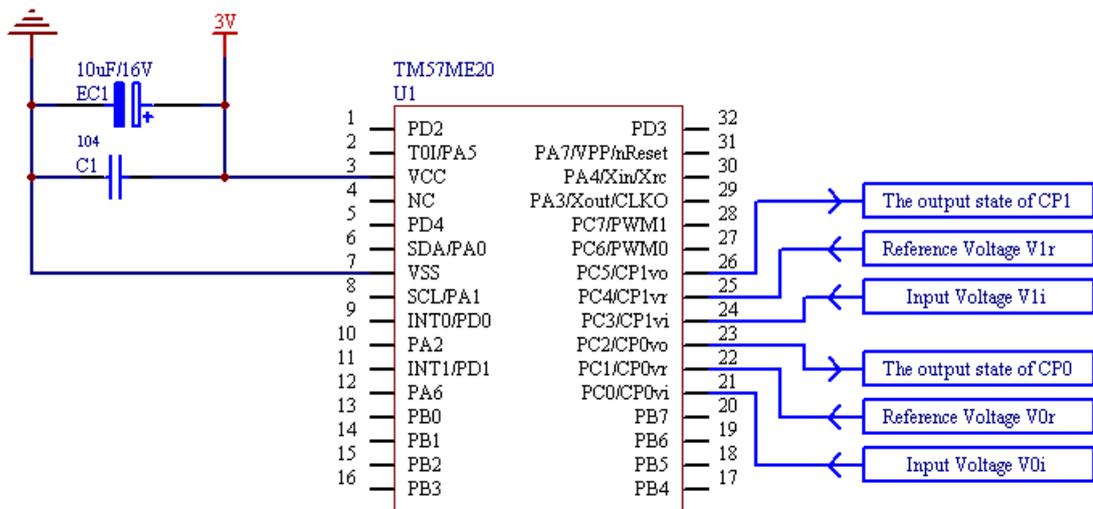
```

Movlw    CPEDGE_f_e      ;B'00000000'
         +VREN_dis       ;B'00000000'
         +CP1_EN_en      ;B'00100000'
         +CP0_EN_en      ;B'00010000'
         +CP1_OE_en      ;B'00001000'
         +CP0_OE_en      ;B'00000100'
         +CIC1_Ext       ;B'00000010'
         +CIC0_Ext       ;B'00000001'
movwf    CMPCTRL
  
```

The address of CMPCTRL is F PLANE 12H.

- 4) After applying power to the DEMO board, connecting PC1/CP0vr and PC4/CP1vr pins to the corresponding reference voltage, and connecting PC0/CP0vi and PC3/CP1vi pins to the corresponding input voltage, the corresponding comparison results can be read from PC2/CP0vo and PC5/CP1vo pins.
- 5) Additionally, the user can also obtain the comparison results through reading the states of CPST, CP1ST (F PLANE 14H, 1) and CPST, CP0ST (F PLANE 14H, 0).

3. Circuit Diagram



## 02. Instruction for using the analog comparator (with internal input reference voltage)

1. To get the detail of the DEMO program, please refer to  
TM57ME20\_CMP\_iRef.ASM

2. Setup (4MHz internal RC clock)

1) Set PC0/CP0Vi, PC1/CP0Vr, PC3/CP1Vi and PC4/CP1Vr to Analog input. For example:

```
movlw    CP1Vr_dig    ;B'00001000'
          +CP1Vi_ana  ;B'00000000'
          +CP0Vr_dig  ;B'00000010'
          +CP0Vi_ana  ;B'00000000'
movwr    PIE
```

The address of PIE is R PLANE 12H.

2) Set PC0 and PC3 as input, and PC2 and PC5 as output. For example:

```
movlw    b'11110110'
movwr    PCE
movlw    b'00011011'
movwf    PCD
```

The address of PCE is R PLANE 07H, and the address of PCD is F PLANE 07H.

3) After setting up as above, the following actions also need to be performed:

- a) Setting the trigger edge of the interrupt event to the falling edge
- b) Enable access to the internal reference voltage (VREN)
- c) Enable CMP1
- d) Enable CMP0
- e) Enable CMP1 output

- f) Enable CMP0 output
- g) Using internal input as CMP1 reference voltage
- h) Using internal input as CMP0 reference voltage

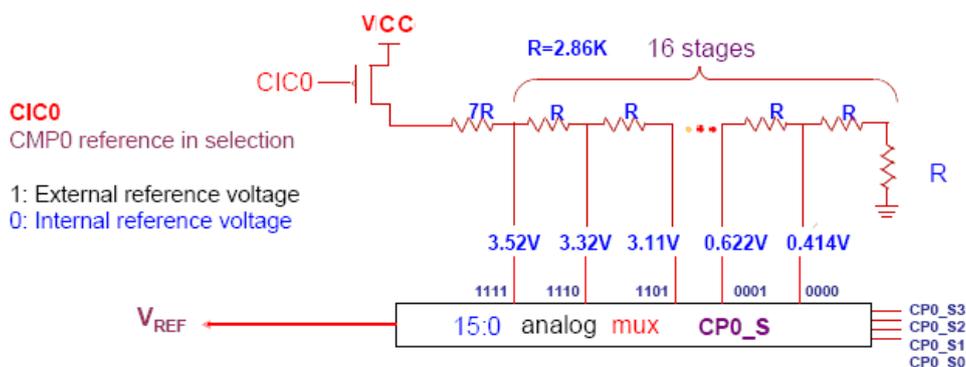
The instructions to execute the procedure are shown as follows:

```

movlw    CPEDGE_f_e    ;B'00000000'
        +VREN_en      ;B'01000000'
        +CP1_EN_en    ;B'00100000'
        +CP0_EN_en    ;B'00010000'
        +CP1_OE_en    ;B'00001000'
        +CP0_OE_en    ;B'00000100'
        +CIC1_Int     ;B'00000000'
        +CIC0_Int     ;B'00000000'
movwf    CMPCTRL
    
```

The address of CMPCTRL is F PLANE 12H.

- 4) The internal reference voltage setting is shown in the following figure:



The reference voltage is divided into 16 levels (0~15), the following formula is obtained based on the principle of voltage divider:

$$\begin{aligned}
 \text{Reference Voltage at level } n \quad V_n &= (VCC/24) \times (n+2) \quad n = 0,1,2,\dots,15) \\
 &= 0.04167 \times (VCC) \times (n+2)
 \end{aligned}$$

The user can set the appropriate voltage level according to the circuit design and store the reference voltage level data (n) corresponding to the level to CP\_S register. The instructions to be executed are as follows:

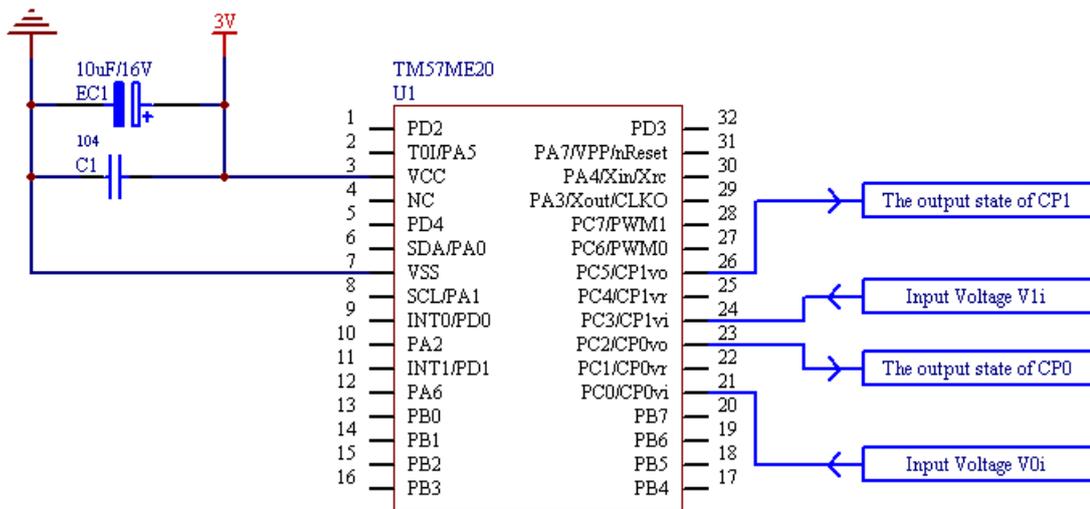
```

movlw    CP1_S_0.04167VCC3    ;B'00010000'
        +CP0_S_0.04167VCC3    ;B'00000001'
movwf    CP_S
    
```

The address of CP\_S is F PLANE 13H. The value of CP1\_S\_0.04167VCC3 is B'00010000', which means the internal reference voltage level of CP1 is 1; The value of CP0\_S\_0.04167VCC3 is B'00000001', which means the internal reference voltage level of CP0 is 1.

- 5) After applying power to the DEMO board, connecting PC0/CP0vi and PC3/CP1vi pins to the corresponding input voltage, the corresponding comparison results can be read from PC2/CP0vo and PC5/CP1vo pins.
- 6) Additionally, the user can also obtain the comparison results through reading the state of CPST, CP1ST (F PLANE 14H, 1) and CPST, CP0ST (F PLANE 14H, 0).

### 3. Internal Input Circuit Diagram



### 03. Instruction for using dual clocks ( switching between the fast and slow clocks )

1. To get the detail of the DEMO program, please refer to  
TM57ME20\_Dual\_clk.ASM
2. Setting up (4MHz internal RC as the fast clock, 32.768KHz external crystal oscillator as the slow clock)

#### 1) Switching from the fast clock to the slow clock

- a) Select the slow type: determined by the Bit[1:0] of TM2SUBTYP (address R PLANE 15H) . There are three possible selections:

0:SXT, 1: ILRC , 2: XRC. Here, low speed external crystal oscillator clock is selected. The instructions to be executed are as follows:

```
movlw    SUBTYP_SXT    ;SUBTYP_SXT=0
movwr    TM2SUBTYP
```

- b) Because an external oscillator clock is used here, the CPU instruction clock output pin must prohibit the output of CPU instruction clock; otherwise, the oscillator can not start-up. The instructions to be executed are as follows

```
movlw    CLK2PIN_no    ;B'00000000'
movwr    HWAUTO
```

The address of HWAUTO is R PLANE 0BH.

Also, the Xin/Xout pin voltage level corresponding to the external oscillator (here, PA3 and PA4) must be set to Input state. The instructions to be executed are as follows:

```
movlw    b'00011000'
movwf    PAD
movlw    b'01100111'
movwr    PAE
```

The address of PAE is R PLANE 05H, and the address of PCD is F PLANE 05H.

- c) When the fast clock is activated, execute the following instruction operations in the following order to switch to the slow clock mode:

```
bsf CLKCTRL,SUBE ; Enable sub-clock (slow clock)
bsf CLKCTRL,SELSUB ; Switch the system clock to the
                    slow speed
bsf CLKCTRL,STPFCK ; Stop the fast clock
```

The address of CLKCTRL is F PLANE 15H, and SUBE, SELSUB and STPFCK are Bit5, Bit7 and Bit6 of it respectively.

- d) Afterward, the CPU instruction clock switches to slow clock. The user can observe the change of the output waveform on Xout by an oscilloscope to observe the clock change.

## 2) Switching from the slow clock to the fast clock.

- a) When the slow is working, execute the following instruction operations in the following order:

```
bcf CLKCTRL,STPFCK ; Start the fast clock
bcf CLKCTRL,SELSUB ; Switch the system clock to the
                    fast speed
bcf CLKCTRL,SUBE ; Stop the slow clock
```

The address of CLKCTRL is F PLANE 15H, and SUBE, SELSUB and STPFCK are Bit5, Bit7 and Bit6 of it respectively.

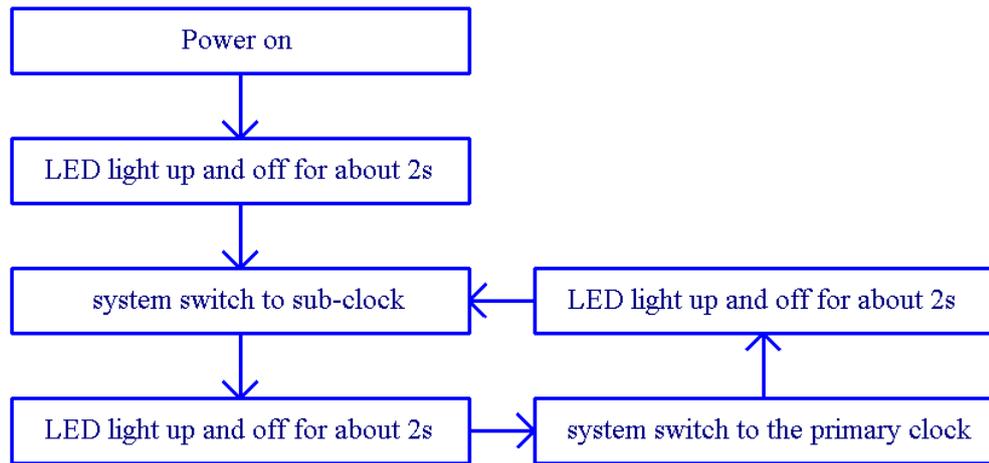
- b) Because the fast clock uses an internal RC circuit (4MHz) here, the CPU instruction clock can be outputted through the CLKO pin. The instructions to be executed are as follows:

```
movlw    CLK2PIN_out    ;B'00001000'
movwr    HWAUTO
```

The address of HWAUTO is R PLANE 0BH.

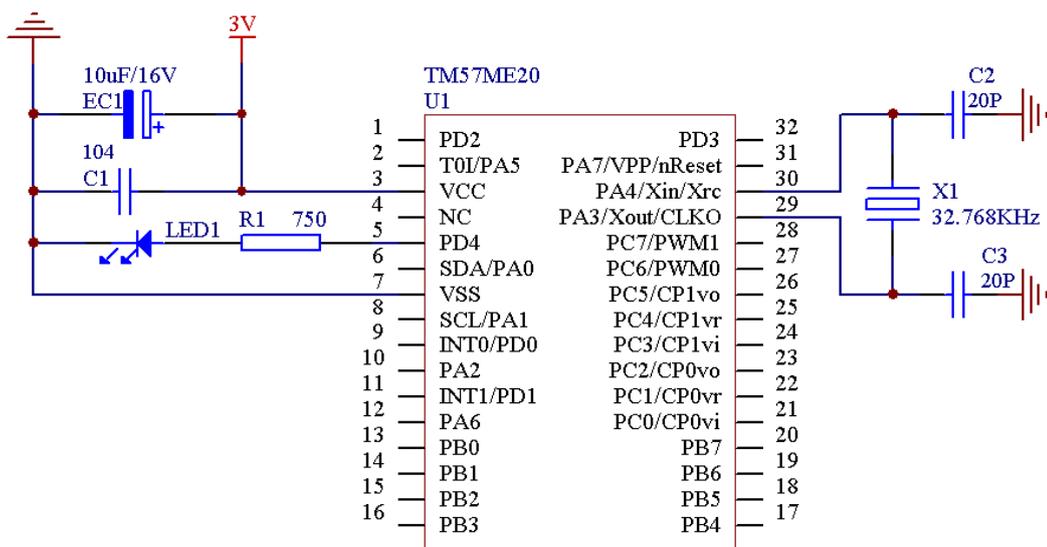
c) Afterward, the CPU instruction clock switches to fast clock. The user can inspect the pin voltage level of CLKO (same as the Xout pin) using an oscilloscope to observe the clock change.

3) The flow chart for this program is as follows:



The user can observe the waveform on CLKO (same as the Xout pin) using an oscilloscope to realize the system clock is in a fast mode or a slow mode.

### 3. Circuit Diagram



**04. Instruction for enabling external interrupts (INT0 and INT1)**

1. To get the detail of the DEMO program, please refer to

TM57ME20\_extIWK.ASM

2. Setup (4MHz internal RC clock)

- 1) Because interrupts are used, the system, here, also enable the automatic preserve current state mode (the hardware will automatic backup W and Status when entering interrupt and restore automatically when exiting interrupt). By executing the following instructions.

```
movlw    HWAUTO_en    ; B'10000000'
movwr    HWAUTO
```

The address of HWAUTO is R PLANE 0BH.

- 2) In considering the power consumption problem, it is recommended to set all the IO in a non-floating state before the system enters the sleep mode. If they are floating, set them to output 0 state.
- 3) Set PD0 / PD1 pins (shared with INT0 and INT1) to input mode and enable pull up. The instructions to be executed are as follows:

```
movlw    1ch
movwr    nPDPU    ; PD0 and PD1 enable pull up
movlw    1ch
movwr    PDE      ; Set PD0 and PD1 to input
                    mode
movlw    03h
movwf    PDD
```

The address of PDPU is R PLANE 11H

The address of PDE is R PLANE 10H

The address of PDD is F PLANE 11H

- 4) Enable INT0 and INT1 interrupts. The instructions to be executed are as follows:

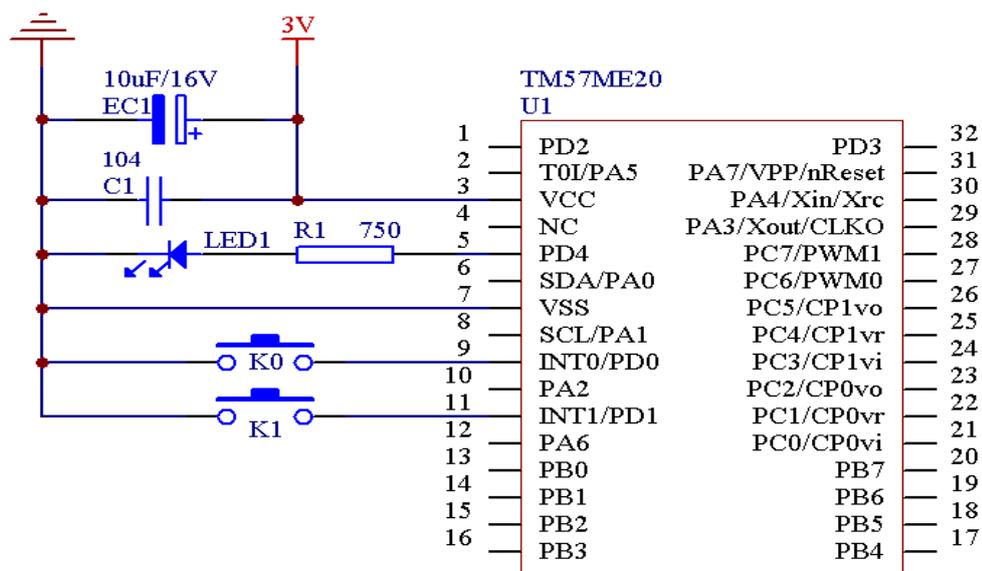
```

movlw    clr_XINT1    ;B'11111101'
movwf    INTI         ;Clear INT1 interrupt flag
movlw    clr_XINT0    ;B'11111110'
movwf    INTI         ;Clear INT0 interrupt flag
movlw    XINT1E_en    ;B'00000010'
        +XINT0E_en    ;B'00000001'
movwf    INTE         ;Enable INT0 and INT1
                    interrupts
    
```

The address of INTI is F PLANE 09H and the address of INTE is F PLANE 08H.

- 5) As such, after the system enters the sleep mode, INT0 and INT1 ports are of high voltage level. As soon as buttons K0 and K1 are pressed, the corresponding port voltage level will be pulled down, an interrupt is accepted, and the system will be woken up. After the LED goes on and off for about 2S) the system will enter the sleep mode again.

### 3. Circuit Diagram



**05. Instruction for using PWM function ( PWM0 and PWM1 )**

1. To get the detail of the DEMO program, please refer to  
TM57ME20\_PWM.ASM
2. Setup (4MHz internal RC clock)
  - 1) Set PC6 / PC7 pins (shared by PWM0 and PWM) to input mode.  
The instructions to be executed are as follows:

```
movlw    b'11000000'
movwr    PCE
```

The address of PCE is R PLANE 07H

- 2) Setting the overflow cycle of PWM (the clock source of PWM is Fosc) . **Execute the following instructions:**

```
movlw    40h
movwf    PWMPERIOD
```

The address of PWMPERIOD is F PLANE 10H

Its overflow cycle  $T=1/F_{osc} \times 64$  (convert 40H to decimal is 64)

$$=0.25 \times 64=16\mu s$$

- 3) Execute the PWM output function for PWM0 / PWM pins. Execute the following instructions:

```
movlw    PWM0E_en    ; B'01000000'
          +PWM1E_en    ; B'00100000'
movwr    HWAUTO
```

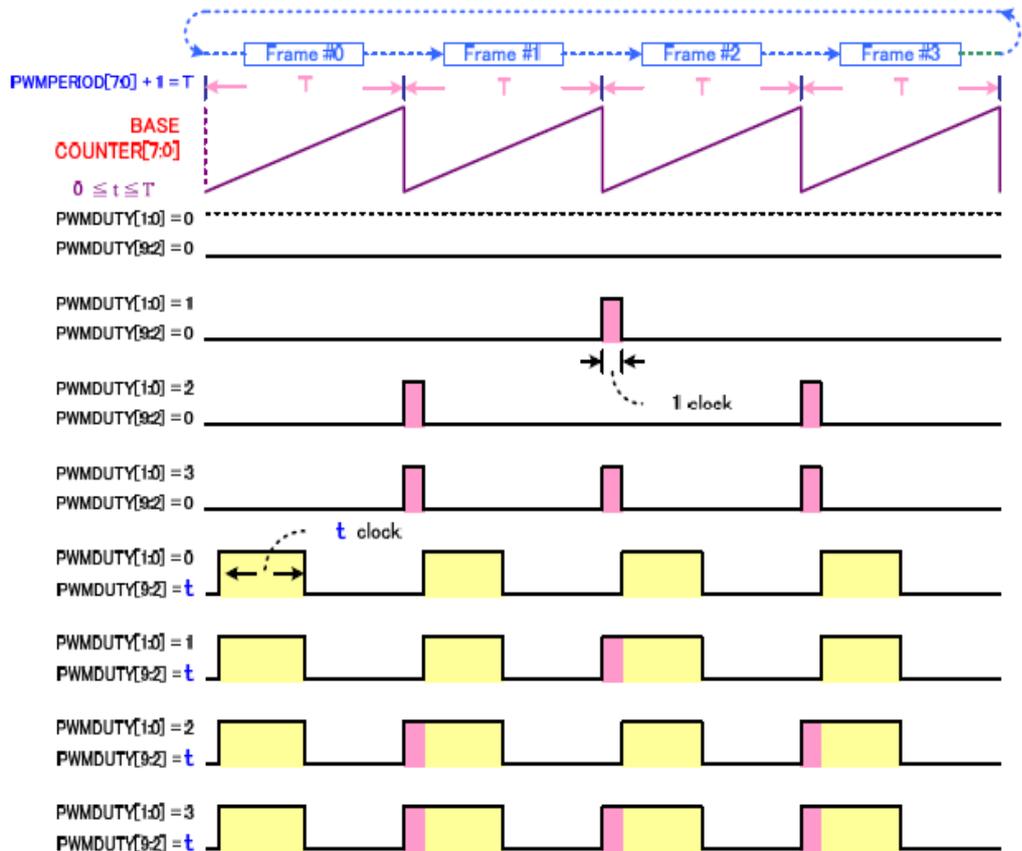
The address of HWAUTO is R PLANE 0BH.

4) Adjust the PWM output waveform. Execute the following instructions:

```

movlw    00h
movwf    PWM0DUTY_MSB
movlw    b'10000000'
movwf    PWM0DUTY_LSB    ; Adjust the PWM0
movlw    10h
movwf    PWM1DUTY_MSB
movlw    b'00000000'
movwf    PWM1DUTY_LSB    ; Adjust the PWM1
    
```

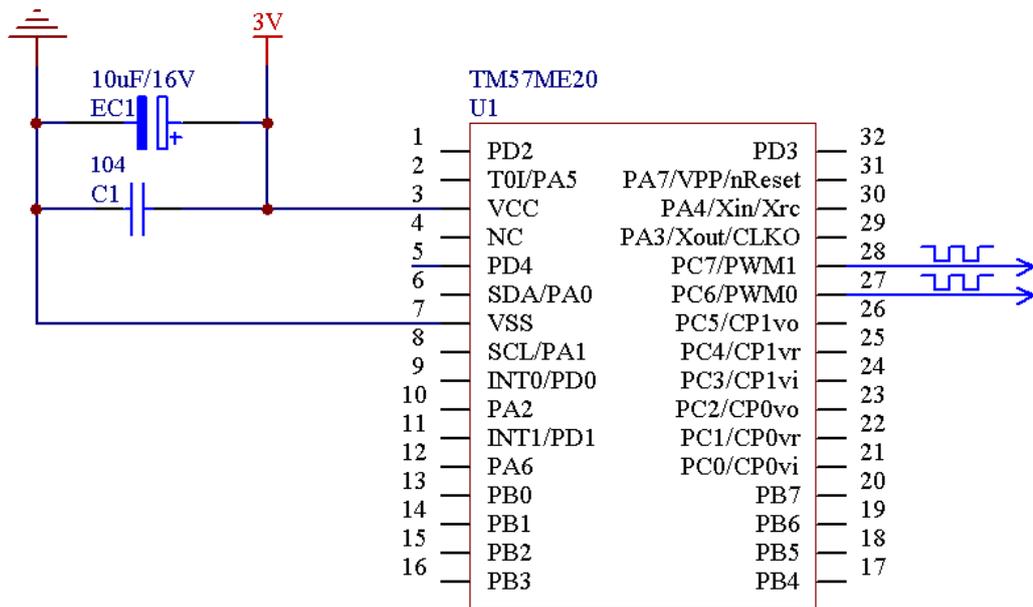
The addresses of PWM0DUTY\_MSB, PWM0DUTY\_LSB, PWM1DUTY\_MSB, and PWM1DUTY\_LSB are F PLANE 0CH, 0DH, 0EH, and 0FH respectively. The user can find out the relationship between them and the clock cycle from the following timing diagram:



PWMDUTY[1:0] in the timing diagram indicate Bit[7:6] of PWM0DUTY\_LSB or PWM1DUTY\_LSB and PWMDUTY[9:2] indicate Bit[7:0] of PWM0DUTY\_MSB or PWM1DUTY\_MSB.

- 5) After applying power to the DEMO board, The output port waveform of PWM0 and PWM1 can be observed using an oscilloscope.

### 3. Circuit Diagram



**06. Instruction for using TIMER0 function**

1. To get the detail of the DEMO program, please refer to  
TM57ME20\_TM0.ASM
2. Setup (4MHz internal RC clock)
  - 1) Configure clock source for TM0, and set up default sub-frequency. Execute the following instructions:

```

movlw    TM0_SEL0.5FOSC ;TM0 clock source (Fosc/2)
          +TM0PSC_DIV256 ;TM0 clock sub-frequency
          256
movwr    TM0CTRL

```

The address of TM0CTRL is R PLANE 02H.

- 2) After configuring TM0, TM0 will start counting immediately. Thereafter, the user can realize the TM0's status through reading the value of TM0. In this example, when TM0 counts to 250, reset TM0 and inverse the output state of PD4. Afterward, repeat this procedure, PD4 pin will output a waveform with a period of  
 $T=1/(Fosc/2) \times 256 \times 250 \times 2=64ms$  on PD4. The instruction to read the current value of TM0 is as follows:

```

Movfw    TIMER0 ; The address of TIMER0 is F PLANE
              01H

```

Of course, the counting data of TM0 can also compare to specified data directly. For example:

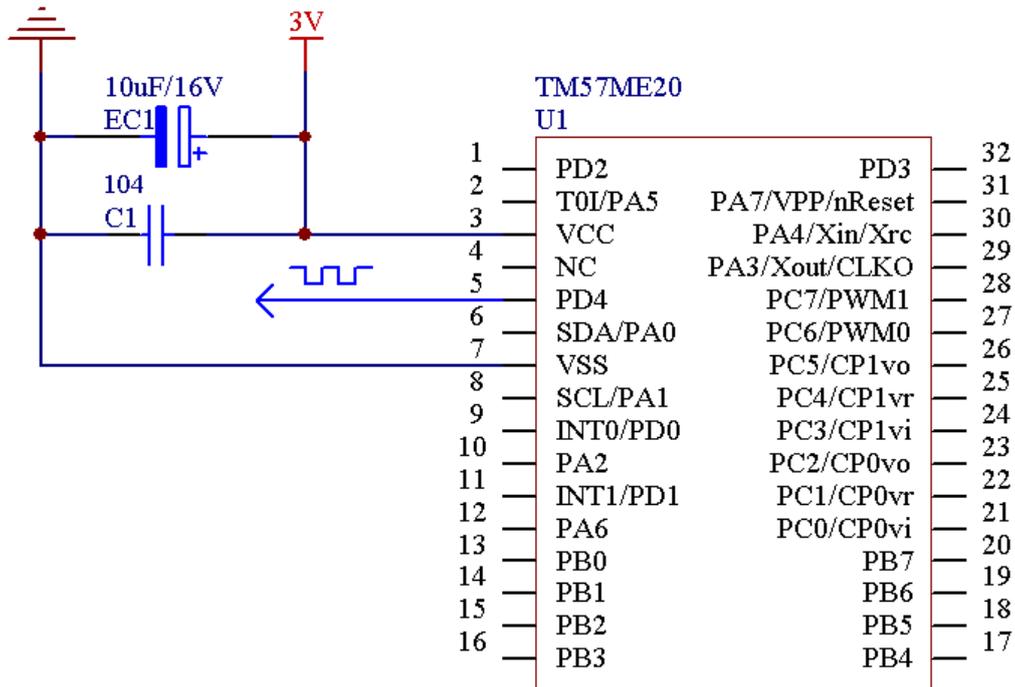
```

movlw    .250
subwf    TIMER0,w

```

Afterward, the `btfs STATUS, C` instruction can be used to determine the larger than or less than relationships.

3. Circuit Diagram



**07. Instruction for using TIMER1 function**

1. To get the detail of the DEMO program, please refer to TM57ME20\_TM1.ASM

2. Setup (4MHz internal RC clock)

1) Setting TM1's pre-scaled clock sub-frequency. Execute the following instructions:

```
movlw    TM1PSC_DIV2
movwr    TM1PSC          ;The address of TM1PSC is
                        R PLANE 0CH
```

The pre-scaled data of TM1 clock is 2, i.e. the period of pre-scaled clock is  $\left[ \frac{1}{(F_{osc}/2)} \right] \times 2 = 1\mu S$  (Note: The clock source of TM1 can only be  $F_{osc}/2$ )

2) Set up the initial value of TM1. This value is stored in TM1RELD\_MSB (R PLANE 0DH) and TM1RELD\_LSB (R PLANE 0EH). When TM1 overflows, this value will be loaded into TM1 counters TIMER1\_MSB (F PLANE 0AH) and TIMER1\_LSB (F PLANE 0BH). Then, TM1 will start counting from the initial value again until it overflows. Repeating over and over like this way. The instruction to set up the initial value of TM1 is as follows:

```
movlw    f0h
movwr    TM1RELD_LSB
movlw    d8h
movwr    TM1RELD_MSB
```

Here, the initial value of TM1 is d8f0h. Because increment 1 will increase the duration by 1 $\mu$ S,

$$\begin{aligned} \text{the overflow period of TM1 is } T &= 1 \times (10000h - d8f0h) \\ &= 10(ms) \end{aligned}$$

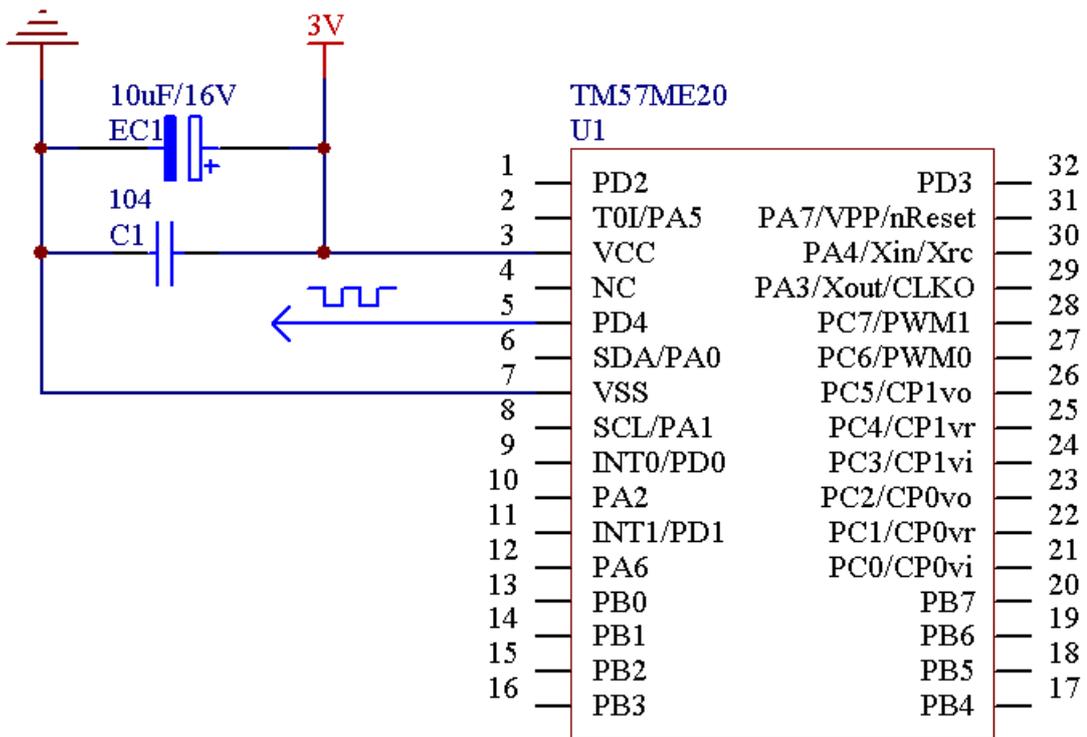
3) For convenience, this program enables the TM1 overflow interrupt function. This way, the MCU will be interrupted every 10ms. When interrupt occurs, it will toggle the output of PD4, Repeating this way, a waveform with a period of  $10 \times 2 = 20\text{ms}$  can be obtained on PD4. The instruction to enable TM1 interrupt is as follows:

```

movlw    clr_TM1I      ;B'11011111'
movwf    INTI          ;Clear TM1 interrupt flag
movlw    TM1IE_en     ; B'00100000'
movwf    INTE          ;EnableTM1 interrupt
    
```

The address of INTI is F PLANE 09H, The address of INTE is F PLANE 08H.

### 3. Circuit Diagram



**08. Instruction for using TIMER2 function**

1. To get the detail of the DEMO program, please refer to

TM57ME20\_TM2.ASM

2. Setup (4MHz internal RC clock)

1) Set up the clock source and the pre-scaled clock of TM2. Execute the following instructions:

```
Movlw   TM2CLK_foscDIV128 ; The clock source of TM2 is
                                CPUCLK/128
                                +TM2DIV_128      ; Setting pre-scaled data as
                                                1/128
Movwr   TM2SUBTYP
```

The address of TM2SUBTYP is R PLANE 15H. The overflow time for TM2 is  $0.25 \times 128 \times 128 = 4.096\text{ms}$ .

2) For convenience, this program enables the TM2 overflow interrupt function. This way, the MCU will be interrupted every 4.096 ms. When interrupt occurs, it will toggle the output of PD4. Repeating this way, a waveform with a period of  $4.096 \times 2 = 8.192\text{ms}$  can be obtained on PD4. Before enabling TM2 interrupt, the automatic preserve current state mode can be set (the hardware will automatic backup Acc and Status when entering interrupt and restore automatically when exiting interrupt). To setup, execute the following instructions.

```
movlw   HWAUTO_en   ;B'10000000'
movwr   HWAUTO
```

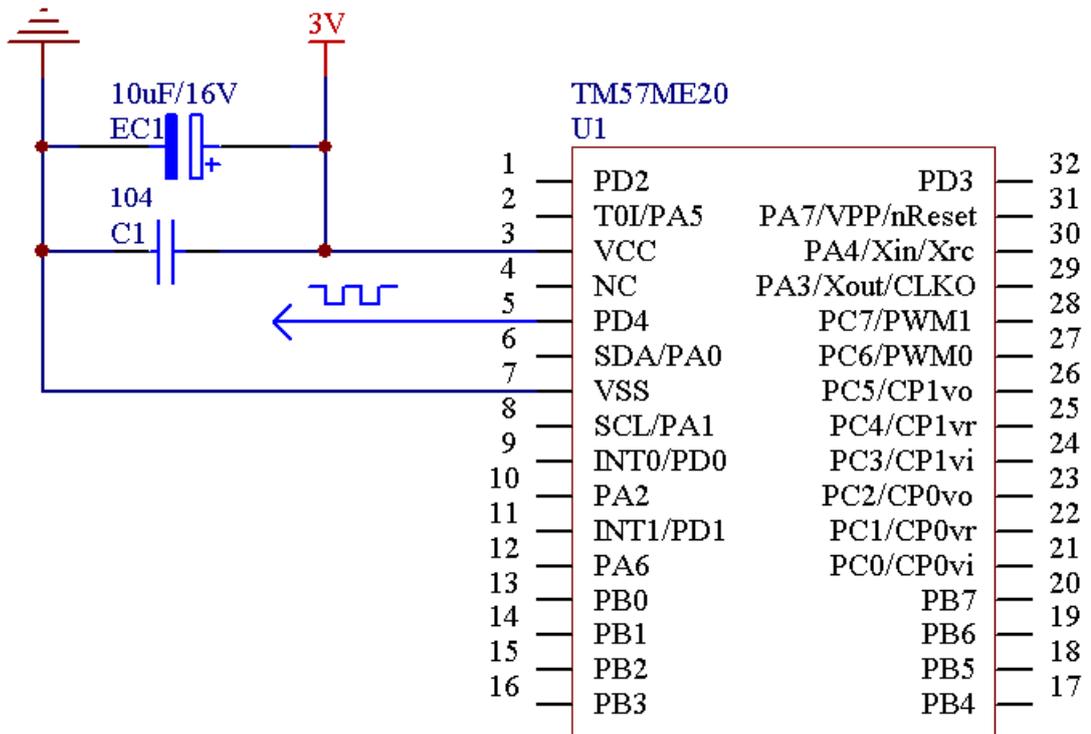
The PLANE of HWAUTO is R PLANE 0BH.

The instruction to enable TM2 interrupt is as follows:

```
movlw   clr_TM2I    ;B'11111011'
movwf   INTI        ;Clear TM2 interrupt flag
movlw   TM2IE_en   ;B'00000100'
movwf   INTE        ;Enable TM2 interrupt
```

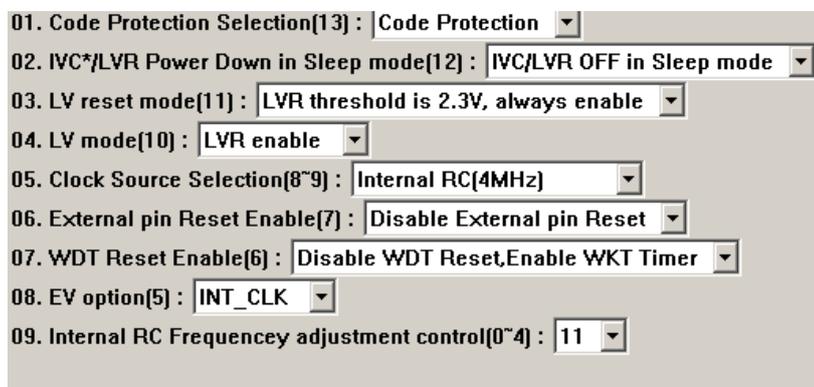
The address of INT1 is F PLANE 09H, and the address of INTE is F PLANE 08H.

3. Circuit Diagram



**09. Instruction for using the function of waking up clock**

1. To get the detail of the DEMO program, please refer to  
TM57ME20\_WKT.ASM
2. Setup (4MHz internal RC clock)
  - 1) The 7th item of Config setting in IDE (S/W of ICE), i.e. WDTE (BIT6), must be set to 0, i.e. Disable WDT Reset, Enable WKT Timer, as shown in the following figure:



- 2) For the power consumption concern, try to set all the IO in a power non-floating state as much before the system enters the sleep mode. If they are floating, set them to output 0 state.
- 3) Setting wake up period and enable automatic preserve current state mode (the hardware will automatic backup W and Status when entering interrupt and restore automatically when exiting interrupt). Here, the wake up period is determined by Bit[1:0] of HWAUTO (R PLANE 0BH). There are 4 periods can be used:

00: WDT/WKT period is 15mS ; WKTPSC\_15mS  
 01: WDT/WKT period is 30mS ; WKTPSC\_30mS  
 10: WDT/WKT period is 60mS ; WKTPSC\_60mS  
 11: WDT/WKT period is 120mS; WKTPSC\_120mS

Here, 30ms is selected.

Setting Bit 7 of HWAUTO (R-plane 0BH) will enable automatic preserve current state mode.  
Because they all use the same register HWAUTO, the registers can be set at the same time:

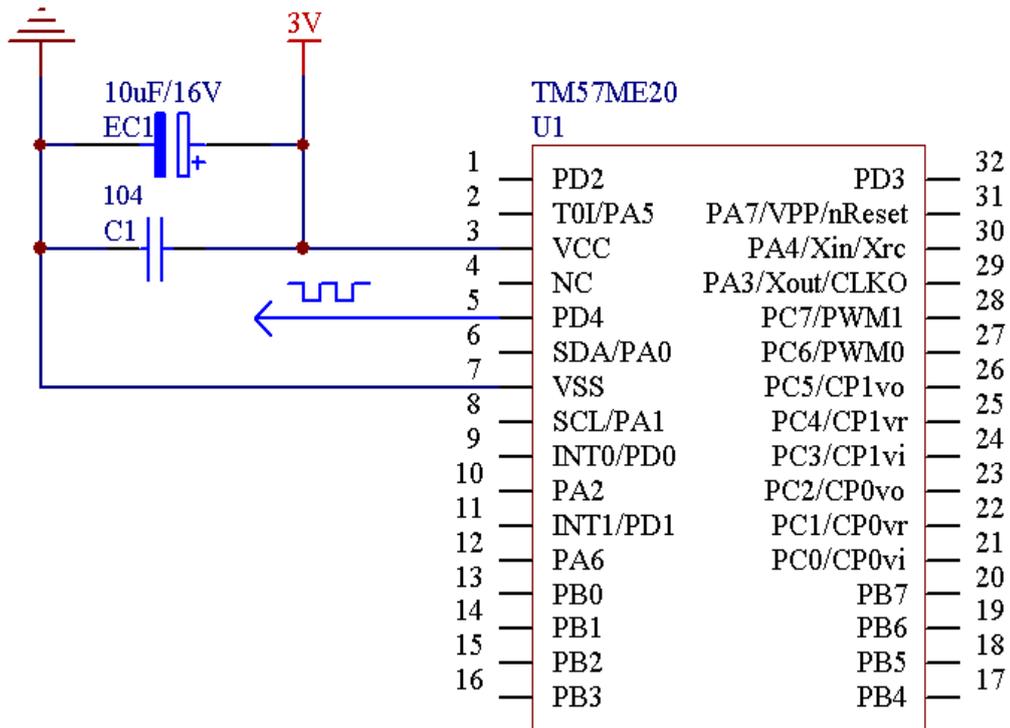
```
Movlw    HWAUTO_en      ;B'10000000'  
        +WKTPSC_30mS   ; Every 30ms wake up once  
Movwr    HWAUTO        ;The address of HWAUTO is  
                        R PLANE 0BH.
```

This way, the system will wake up every 30 ms and triggers an interrupt and toggle the output of PD4. Repeating over and over like this way, a waveform with a period of  $T=30 \times 2=60\text{ms}$  will be outputted on PD4 pin. The instruction to enable WKT interrupt is as follows:

```
movlw    clr_WKTI      ;B'11110111'  
movwf    INTI         ;Clear WKT interrupt flag  
movlw    WKTIE_en     ;B'00001000'  
movwf    INTE         ;Enable WKT interrupt
```

Register INTI locates at F PLANE 09H and Register INTE locates at F PLANE 08H.

3. Circuit Diagram



**10. Instruction for using port B port function**

1. To get the detail of the DEMO program, please refer to  
TM57ME20\_WKUP.ASM

2. Setup (4MHz internal RC clock)

1) IO initialization. For the power consumption concern, try to set all the IO in a power non-floating state as much as possible before the system enters the sleep mode. If they are floating, set them to output 0 state.

2) Set the IO pin corresponding to the key input function as Input mode and enable pull up. The setting is as follows:

```

movlw    c0h
movwr    PBPU           ;PB0~PB5 pull up enabled
movlw    c0h
movwr    PBE           ;PB0~PB5 set to input
movlw    3fh
movwf    PBD

```

The addresses of PBPU and PBE correspond to R PLANE 06H and 09H respectively and that of PBD F PLANE 06H.

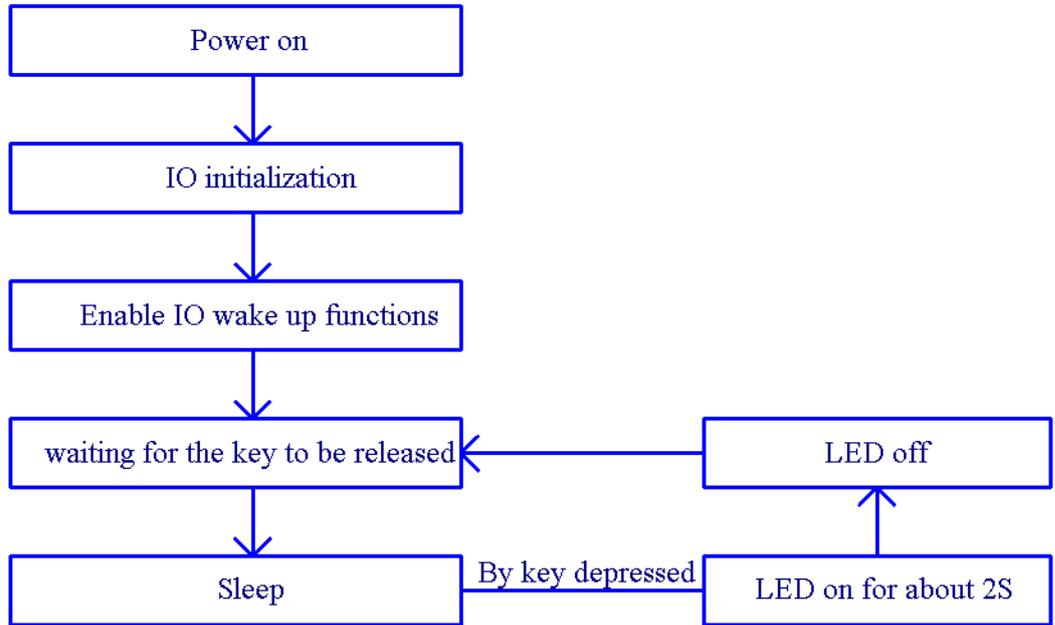
3) Enable wake up function. Here, only the wake up function of PB0~PB5 on PortB port PB0~PB5 are enabled. The instructions and flow chart are shown as follows:

```

movlw    3fh
movwr    PBWKUP       ; The addresses of PBWKUP
                      correspond to RPLANE 14H

```

4) The action flow for this program is as follows:



3. Circuit Diagram

