十速科技股份有限公司 tenx technology inc.	Total page	23
	Doc No	**AP-TM57ME20_01S**
	Rev.	1.0

文件名稱	TM57ME20 DEMO CODE FOR BASIC FUNCTIONS

版次	生效日	ECN No.	制修訂者	修訂內容概要
1.0	2010/09/29	EN-AP-10-09004	劉凱	新頒。

十速科技股份有限公司
tenx technology inc.

Advance Information

TM57 系列

TM57ME20

DEMO CODE FOR BASIC FUNCTIONS

Application Note

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

tenx technology inc.

CONTENTS

PRODUCT NAME 3
TM57 系列IC 3
TITLE 3
TM57PE12_ DEMO CODE FOR BASIC FUNCTIONS 3
APPLICATION NOTE 3
01. 模拟比较器的使用方法（参考电压为外部输入） 3
1. 具体DEMO程序请参考 TM57ME20_CMP_eRef.ASM 3
2. 设置方法（内部RC4MHz时钟） 3
3. 线路图 5
02. 模拟比较器的使用方法（参考电压为内部输入） 5
1. 具体DEMO程序请参考 TM57ME20_CMP_iRef.ASM 5
2. 设置方法（内部RC4MHz时钟） 5
3. 线路图 7
03. 双时钟的使用方法（快/慢时钟的切换） 8
1. 具体DEMO程序请参考 TM57ME20_Dual_clk.ASM 8
2. 设置方法（内部RC4MHz作快时钟，外部晶振 32.768KHz作慢时钟） 8
3. 线路图 10
04. 外部中断唤醒的使用方法（INT0 和INT1） 11
1. 具体DEMO程序请参考 TM57ME20_extIWK.ASM 11
2. 设置方法（内部RC4MHz时钟） 11
3. 线路图 12
05. PWM功能的使用方法（PWM0 和PWM1） 12
1. 具体DEMO程序请参考 TM57ME20_PWM.ASM 12
2. 设置方法（内部RC4MHz时钟） 12
3. 线路图 15
06. TIMER0 功能的使用方法 15
1. 具体DEMO程序请参考 TM57ME20_TM0.ASM 15
2. 设置方法（内部RC4MHz时钟） 15
3. 线路图 16
07. TIMER1 功能的使用方法 16
1. 具体DEMO程序请参考 TM57ME20_TM1.ASM 16
2. 设置方法（内部RC4MHz时钟） 16
3. 线路图 18

08. TIMER2 功能的使用方法 18
1. 具体DEMO程序请参考 TM57ME20_TM2.ASM 18
2. 设置方法（内部RC4MHz时钟） 18
3. 线路图 19
09. 唤醒时钟功能的使用方法 20
1. 具体DEMO程序请参考 TM57ME20_WKT.ASM 20
2. 设置方法（内部RC4MHz时钟） 20
3. 线路图 21
10. PORTB端口功能的使用方法 22
1. 具体DEMO程序请参考 TM57ME20_WKUP.ASM 22
2. 设置方法（内部RC4MHz时钟） 22
3. 线路图 23

PRODUCT NAME

TM57 系列 IC

TITLE

TM57PE12_ DEMO CODE FOR BASIC FUNCTIONS

APPLICATION NOTE

01. 模拟比较器的使用方法（参考电压为外部输入）

1. 具体 DEMO 程序请参考 TM57ME20_CMP_eRef.ASM
2. 设置方法（内部 RC4MHz 时钟）

   1) 将 PC0/CP0Vi、PC1/CP0Vr、PC3/CP1Vi 和 PC4/CP1Vr 设为 Analog input。如：

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

   PIE 地址为 R 区 12H。

   2) 将 PC0、PC1、PC3 和 PC4 设为 input，而 PC2 和 PC5 设为 output。如：

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

   PCE 地址为 R 区 07H，PCD 地址为 F 区 07H。

   3) 执行完上面后还需要做如下动作：
      a) 设置中断事件的触发沿为下降沿
      b) 禁止内部参考电压接入（VREN）
      c) CMP1 使能

d) CMP0 使能
e) CMP1 输出使能
f) CMP0 输出使能
g) CMP1 的参考电压使用外部输入方式
h) CMP0 的参考电压使用外部输入方式

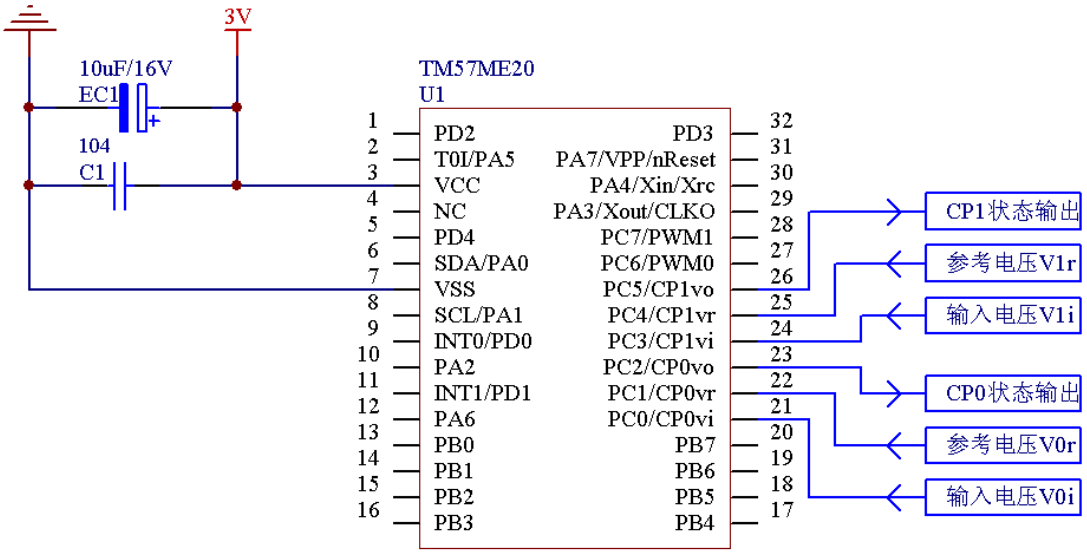
其指令执行方式如下：

```
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
```

CMPCTRL 地址为 F 区 12H。

4) DEMO 板上电后，将 PC1/CP0vr 和 PC4/CP1vr 管脚接入相应的参考电压，将 PC0/CP0vi 和 PC3/CP1vi 管脚接入相应的输入电压，之后从 PC2/CP0vo 和 PC5/CP1vo 管脚就可看到对应的比较结果。
5) 另外用户还可以通过读取 CPST,CP1ST(F 区的 14H,1) 和 CPST,CP0ST(F 区的 14H,0) 的状态获得比较的结果。

3. 线路图



02. 模拟比较器的使用方法（参考电压为内部输入）

1. 具体 DEMO 程序请参考　TM57ME20_CMP_iRef.ASM

2. 设置方法（内部 RC4MHz 时钟）

1) 将 PC0/CP0Vi、PC1/CP0Vr、PC3/CP1Vi 和 PC4/CP1Vr 设为 Analog input。如：

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

PIE 地址为 R 区 12H。

2) 将 PC0 和 PC3 设为 input，而 PC2 和 PC5 设为 output。如：

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

PCE 地址为 R 区 07H，PCD 地址为 F 区 07H。

3) 执行完上面后还需要做如下动作

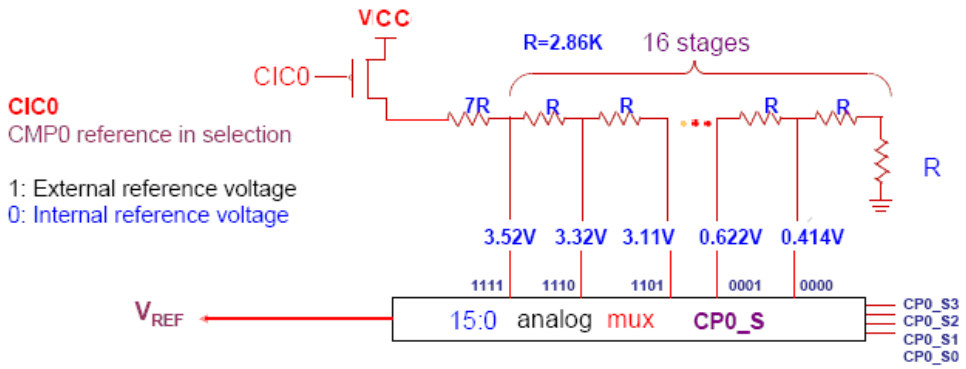
i) 设置中断事件的触发沿为下降沿
j) 使能内部参考电压接入（VREN）
k) CMP1 使能
l) CMP0 使能
m) CMP1 输出使能
n) CMP0 输出使能
o) CMP1 的参考电压使用内部输入方式
p) CMP0 的参考电压使用内部输入方式

其指令执行方式如下：

```
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
```

CMPCTRL 地址为 F 区 12H。

4) 内部参考电压设置，如下图所示：



其值分为 16(0~15)个档位，根据分压原理，得出如下计算公式：

档位电压值 Vn=(VCC/24)×(n+2)　n=0,1,2,~,15)
=0.04167×（VCC）×（n+2）

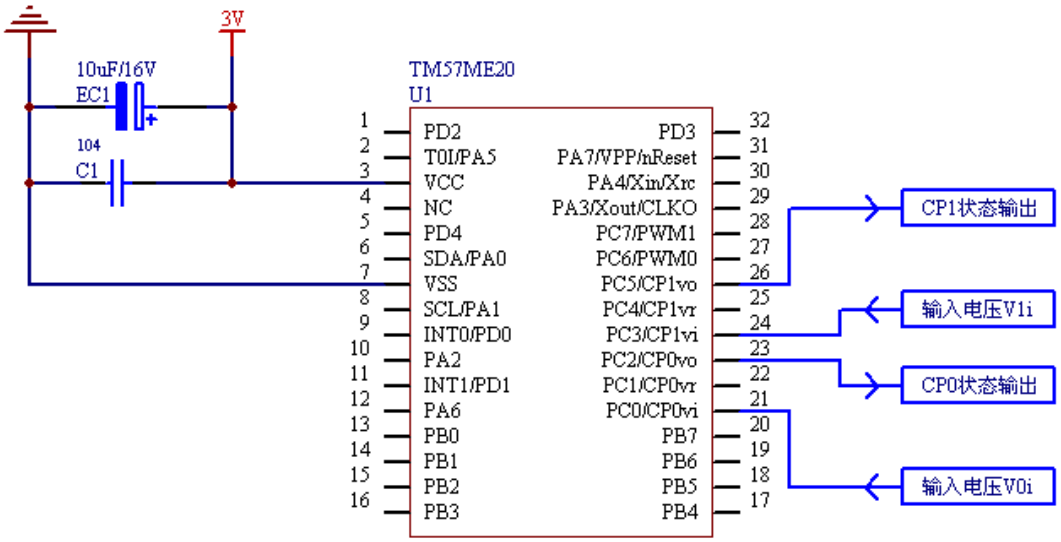
用户根据需要设置适当的档位，并将与档位对应的数据(n) 储存在 CP_S 寄存器。指令如下：

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

CP_S 地址为 F 区 13H。CP1_S_0.04167VCC3 的值为 B'00010000'，表示 CP1 的内部参考电压档位是 1；CP0_S_0.04167VCC3 的值为 B'00000001'，表示 CP0 的内部参考电压档位是 1。

5) 相关 DEMO 板上电后，将 PC0/CP0vi 和 PC3/CP1vi 管脚接入相应的输入电压，之后从 PC2/CP0vo 和 PC5/CP1vo 管脚就可看到对应的比较结果。
6) 另外用户还可以通过读取 CPST,CP1ST(F 区的 14H,1) 和 CPST,CP0ST(F 区的 14H,0) 的状态获得比较的结果。

3. 线路图



03. 双时钟的使用方法（快/慢时钟的切换）

1. 具体 DEMO 程序请参考　TM57ME20_Dual_clk.ASM
2. 设置方法（内部 RC4MHz 作快时钟，外部晶振 32.768KHz 作慢时钟）

   1) 主时钟（快时钟）到子时钟（慢时钟）的切换

      a) 选子时钟类型，由 TM2SUBTYP（地址 R 区 15H）的 Bit[1:0] 决定，有 3 种选择：0:SXT , 1: ILRC , 2: XRC，这里选低速外部晶体振荡时钟。执行指令如下：

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

      b) 由于此处要用到外部振荡时钟，所以 CPU 指令时钟输出脚 CLKO 必须禁止 CPU 指令时钟输出。否则会造成芯片无法起振。执行指令如下：

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

HWAUTO 地址为 R 区 0BH。

并且外部振荡对应时钟脚位（这里指 PA3 和 PA4）必须设为 Input 状态。执行指令如下：

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

PAE 地址为 R 区 05H，PAD 地址为 F 区 05H。

c) 在快时钟工作时，依次执行如下指令切换成慢时钟工作。如下所示：

```
bsf  CLKCTRL,SUBE     ;使能慢速时钟
bsf  CLKCTRL,SELSUB ;将系统时钟切换到慢速时钟
bsf  CLKCTRL,STPFCK  ;停止快时钟
```

CLKCTRL 的地址为 F 区 15H，SUBE、SELSUB 和 STPFCK 分别是其 Bit5、Bit7 和 Bit6。

d) 之后系统将工作在独立的慢时钟下。用户可以用示波器检测 Xout 脚位，观察时钟的变化。

2) 慢时钟到快时钟的切换

a) 在子时钟工作时，依次执行如下指令操作：

```
bcf    CLKCTRL,STPFCK  ;开启快时钟
bcf    CLKCTRL,SELSUB  ;将系统时钟切换到快时钟
bcf    CLKCTRL,SUBE      ;停止慢时钟
```

每条指令的意图见指令后面的注解。
CLKCTRL 的地址为 F 区 15H，SUBE、SELSUB 和 STPFCK 分别是其 Bit5、Bit7 和 Bit6。

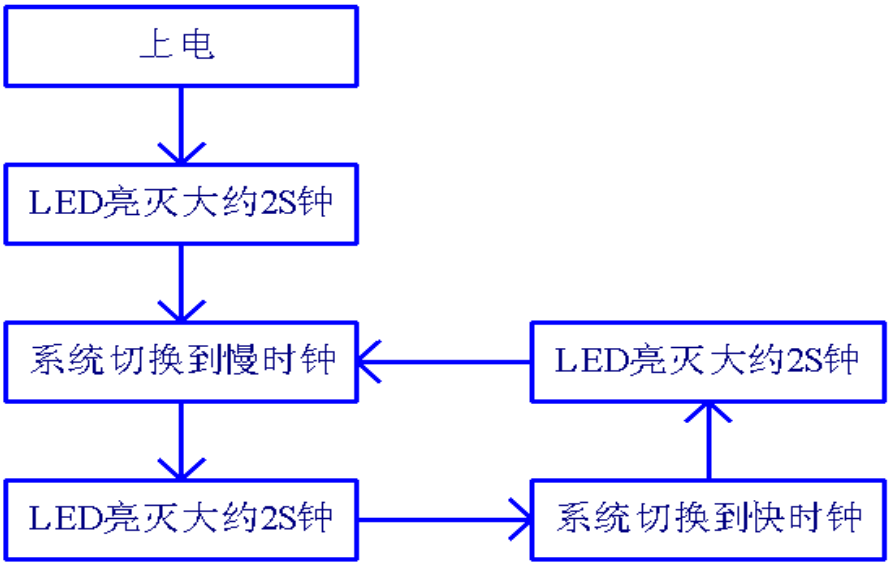
b) 由于这里主时钟采用的是内部 RC(4MHz)，所以可以将系统指令时钟通过 CLKO 引脚输出。执行指令如下：

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

HWAUTO 地址为 R 区 0BH。

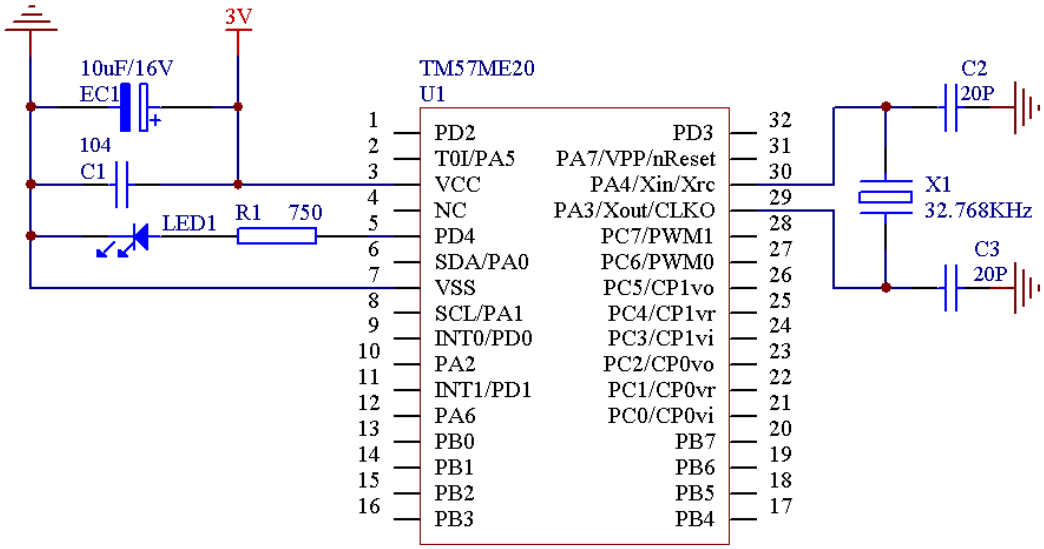
c) 之后系统将工作在独立的主时钟（即快速时钟）下。用户可以用示波器检测 CLKO（与 Xout 脚位相同）的脚位，观察时钟的变化。

3) 本程序的动作流程如下：



用户可以用示波器检测 CLKO（与 Xout 脚位相同）的脚位，观察时钟的变化来判定程序是工作在快速模式还是低速模式。

3. 线路图



04. 外部中断唤醒的使用方法（INT0 和 INT1）

1. 具体 DEMO 程序请参考　TM57ME20_extIWK.ASM
2. 设置方法（内部 RC4MHz 时钟）

   1) 由于用到中断，所以这里将系统设置为自动保护现场模式（进中断时硬件自动备份 W 和 Status，并在退出中断时自动恢复）。执行如下指令可以实现。

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

   HWAUTO 地址为 R 区 0BH。

   2) 考虑到耗电问题，系统进入休眠前尽可能将所有 IO 设置为非耗电状态，如果是空接，那么将其设置为输出 0 状态。

   3) 将 INT0 和 INT1 所共脚的 IO（分别为 PD0 和 PD1）设为输入模式，且开启上拉。执行指令如下：

```
movlw     1ch
movwr     nPDPU     ;PD0 和 PD1 开启上拉
movlw     1ch
movwr     PDE       ;PD0 和 PD1 设为输入模式
movlw     03h
movwf     PDD
```

   PDPU 地址为 R 区 11H
   PDE 地址为 R 区 10H
   PDD 地址为 F 区 11H

   4) 开启 INT0 和 INT1 中断，执行指令如下：

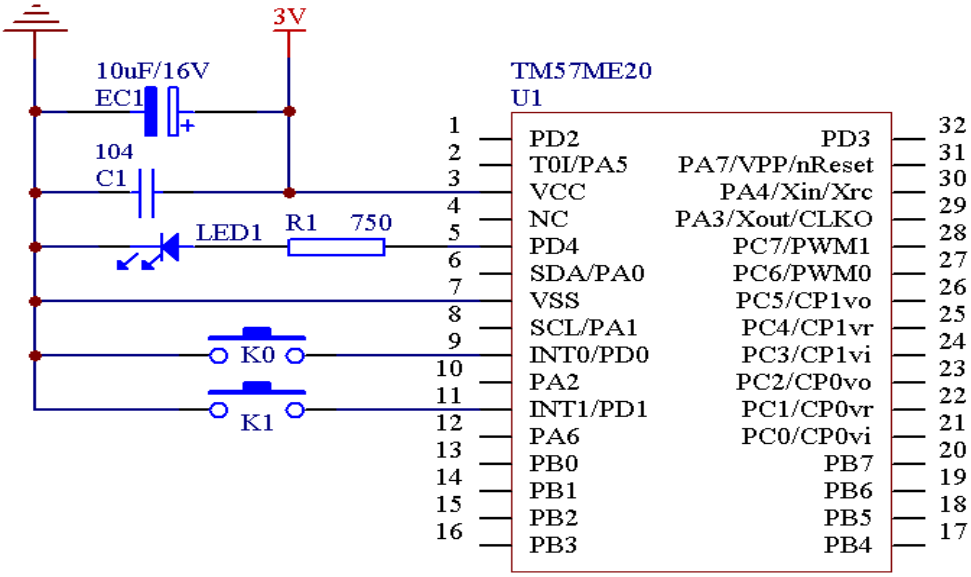
```
movlw     clr_XINT1     ;B'11111101'
movwf     INTI          ;清 INT1 中断标志
movlw     clr_XINT0     ;B'11111110'
movwf     INTI          ;清 INT0 中断标志
movlw     XINT1E_en     ;B'00000010'
```

```
                +XINT0E_en      ;B'00000001'
movwf           INTE            ;开启 INT1,INT0 中断
```

INTI 地址为 F 区 09H，INTE 地址为 F 区 08H。

5) 这样，系统在进入睡眠后，INT0 和 INT1 端口为高电平。一旦按键 K0 和 K1 按下，则对应的端口电平被拉低，信号产生跳变，系统将被唤醒，执行一系列动作后（这里为亮灭 LED 各大约 2S 钟）系统又进入睡眠。

3. 线路图



05. PWM 功能的使用方法（PWM0 和 PWM1）

1. 具体 DEMO 程序请参考　TM57ME20_PWM.ASM

2. 设置方法（内部 RC4MHz 时钟）

1) 将 PWM0 和 PWM1 所共脚的 IO（分别为 PC6 和 PC7）设为输出模式。执行如下指令：

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

PCE 地址为 R 区 07H

2) 设置 PWM 的溢出周期（PWM 的时钟源为 Fosc）。执行如下指令：

```
movlw     40h
movwf     PWMPERIOD
```

PWMPERIOD 地址为 F 区 10H
其溢出周期 T=1/Fosc×64（40H 转 10 进制为 64）
=0.25×64=16us

3) 设置 PWM 从相应的管脚输出（PWM0 对应 PC6，PWM1 对应 PC7）。执行如下指令：

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

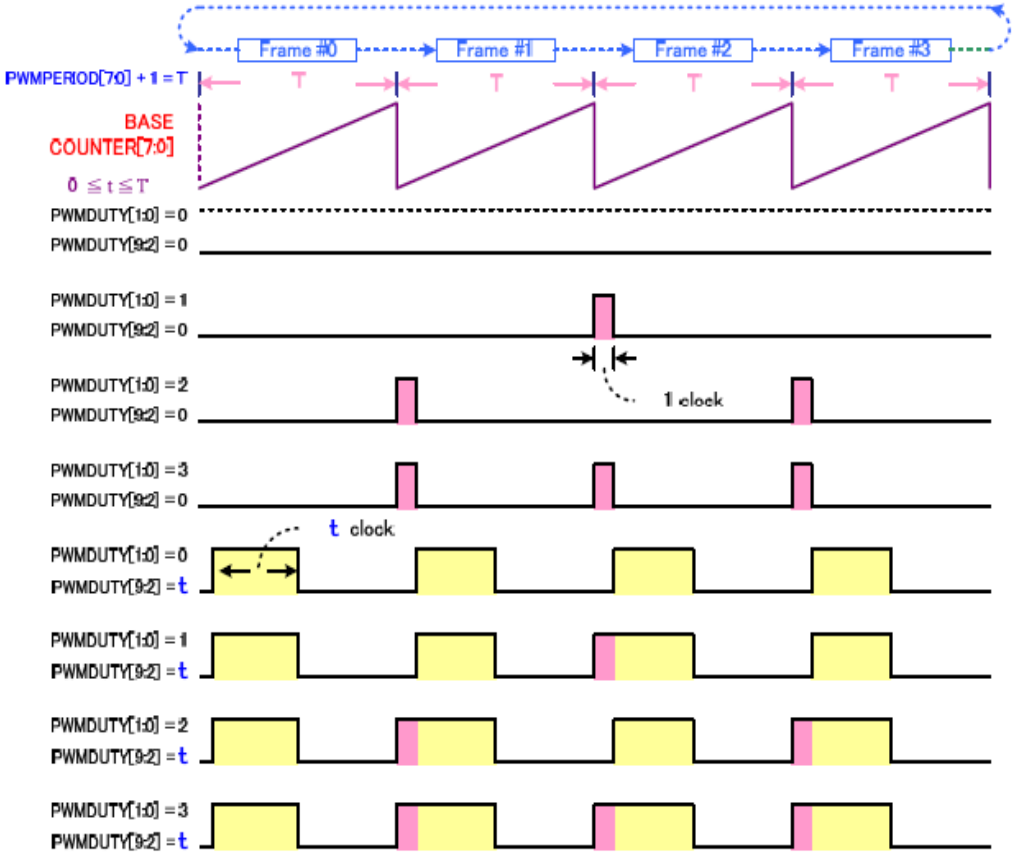
HWAUTO 地址为 R 区 0BH。

4) 调整 PWM 的输出波形。执行如下指令：

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

PWM0DUTY_MSB 、 PWM0DUTY_LSB 、

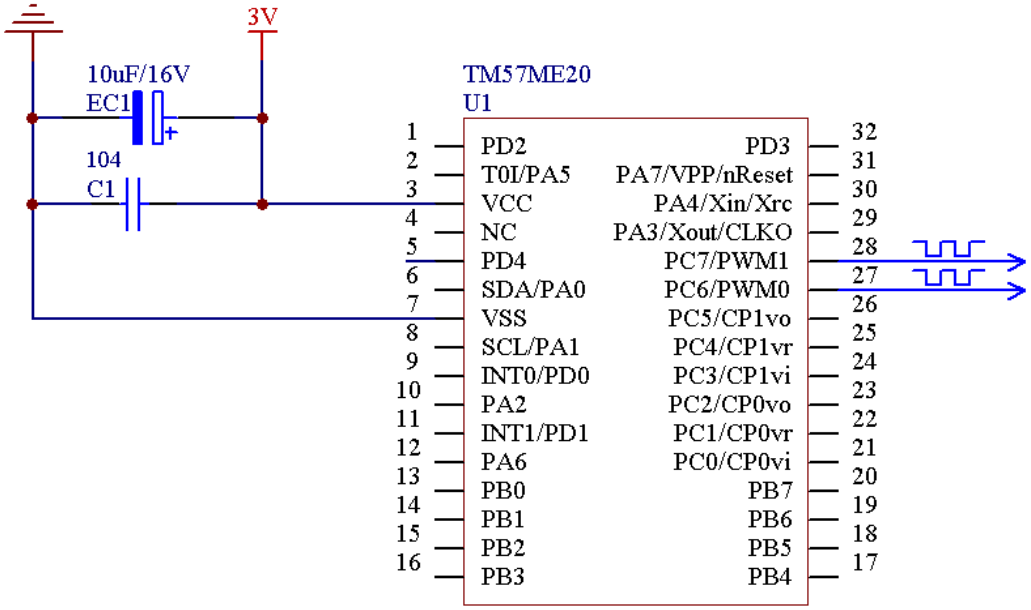
PWM1DUTY_MSB 和 PWM1DUTY_LSB 地址分别是 F 区的 0CH、0DH、0EH 和 0FH，它们与周期的关系，用户可以从以下图表中看出：



图中的 PWMDUTY【1：0】是指 PWM0DUTY_LSB 或 PWM1DUTY_LSB 的 Bit[7:6]，而 PWMDUTY【9：2】则是指 PWM0DUTY_MSB 或 PWM1DUTY_MSB 的 Bit[7:0]。

5) 相关 DEMO 板上电后，通过示波器可以观察到 PWM0 和 PWM1 输出端口的波形。

3. 线路图



06. TIMER0 功能的使用方法

1. 具体 DEMO 程序请参考　TM57ME20_TM0.ASM

2. 设置方法（内部 RC4MHz 时钟）

1) 给 TM0 配置时钟源，并设置预分频。执行如下指令：

```
movlw     TM0_SEL0.5FOSC    ;TM0 时钟源 (Fosc/2)
         +TM0PSC_DIV256     ;TM0 时钟分频 256
movwr     TM0CTRL
```

TM0CTRL 的地址为 R 区的 02H。

2) 配置好 TM0 后，TM0 立即开始计数，用户此后可以通过读取 TM0 的值来掌握系统运行的时间，本范例是当 TM0 计数到 250 时，让 TM0 复位，并且取反 PD4 的输出状态。之后反复这样操作，在 PD4 上得到一个周期
T=1/（Fosc/2）×256×250×2=64ms 的波形。读取 TM0 当前计数值的指令如下：

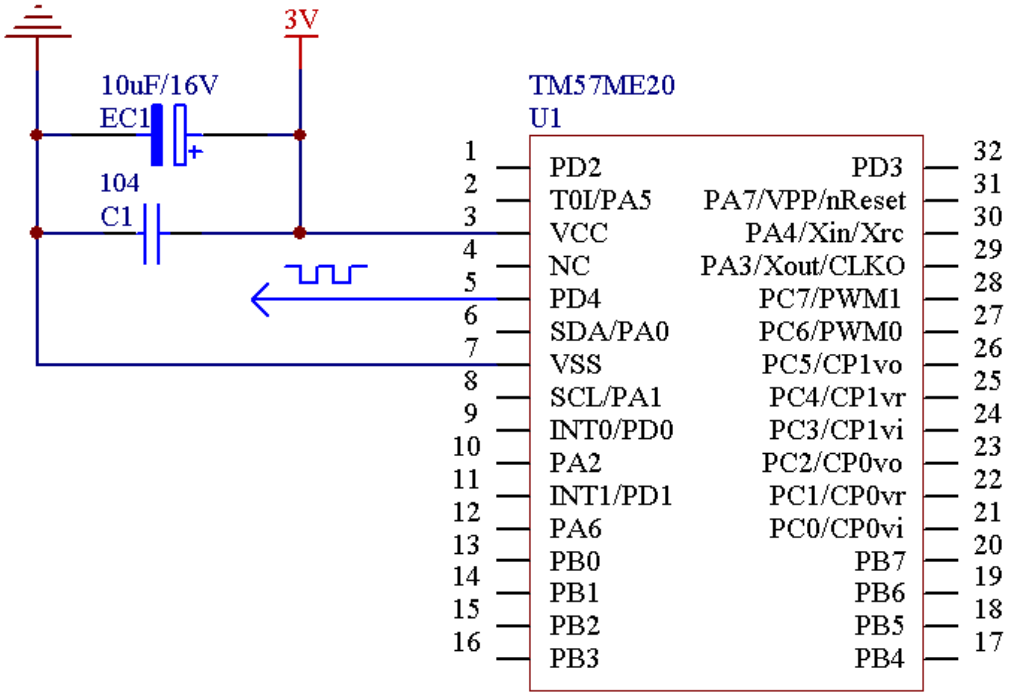
```
Movfw      TIMER0    ; TIMER0 地址为 F 区的 01H
```

当然，也可直接将其与相关数据比较，如：

```
movlw      .250
subwf      TIMER0,w
```

之后可通过 btfss STATUS,C 指令来判断大小关系。

3. 线路图



07. TIMER1 功能的使用方法

1. 具体 DEMO 程序请参考 TM57ME20_TM1.ASM

2. 设置方法（内部 RC4MHz 时钟）

1) 设置 TM1 时钟分频。执行如下指令：

```
movlw      TM1PSC_DIV2
movwr      TM1PSC   ; TM1PSC 地址为 R 区的 0CH
```

TM1 时钟采用 2 分频，即每增加 1，耗时【1/（Fosc/2）】× 2=1uS（注意，TM1 的时钟源只能是 Fosc/2）

2) 设置 TM1 的预载值。这个值通过 TM1RELD_MSB（R 区 0DH）和 TM1RELD_LSB（R 区 0EH）存储，当 TM1 溢出的时候，这个值将会立即载入 TM1 计数器 TIMER1_MSB（F 区 0AH）和 TIMER1_LSB（F 区 0BH）里面。然后 TM1 又在此基础上递增，直到溢出，如此反复。设置 TM1 的预载值的指令如下：

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

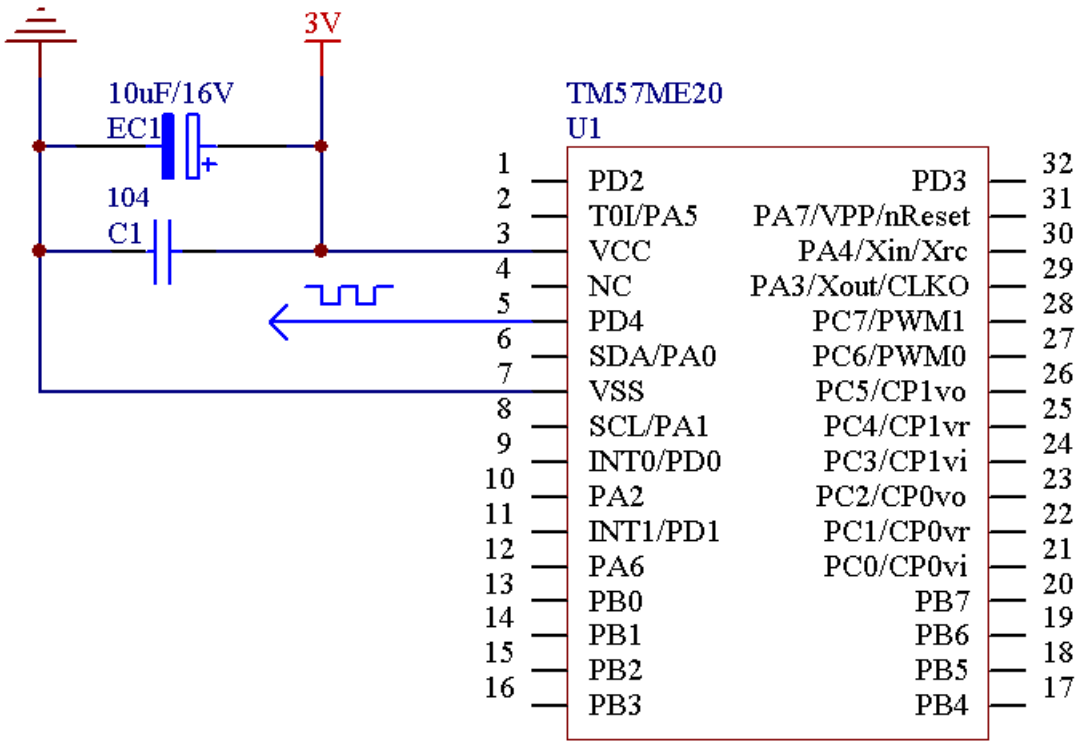
这里 TM1 预载值为 d8f0h，由于递增 1 需要 1uS 的时间，
TM1 的溢出周期 T=1×（10000h-d8f0h）
=10(ms)

3) 为了方便，本程序开启了 TM1 溢出中断使能，这样系统将会每隔 10ms 中断一次，而每中断一次，又会取反 PD4，反复这样做，在 PD4 将得到一个周期为 10×2=20ms 的波形。开启 TM1 中断的指令如下：

```
movlw    clr_TM1I       ;B'11011111'
movwf    INTI           ;清 TM1 中断标志
movlw    TM1IE_en       ; B'00100000'
movwf    INTE           ;开启 TM1 中断
```

INTI 地址为 F 区 09H，INTE 地址为 F 区 08H。

3. 线路图



08. TIMER2 功能的使用方法

1. 具体 DEMO 程序请参考 TM57ME20_TM2.ASM

2. 设置方法（内部 RC4MHz 时钟）

1) 设置 TM2 的时钟源及预分频。执行如下指令：

```
Movlw      TM2CLK_foscDIV128     ;TM2 时钟源为 CPUCLK/128
           +TM2DIV_128           ;采用 128 分频
Movwr      TM2SUBTYP
```

TM2SUBTYP 的地址为 R 区的 15H，TM2 的溢出时间为 0.25×128×128=4.096ms。

2) 为了方便，本程序开启了 TM2 溢出中断使能，这样系统将会每隔 4.096 ms 中断一次，而每中断一次，又会取反 PD4，反复这样做，在 PD4 将得到一个周期为 4.096x2=8.192ms 的波形，在开

启 TM2 中断前，可以设置自动保护现场模式（进中断时硬件自动备份 Acc 和 Status，并在退出中断时自动恢复），执行如下指令可以实现：

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

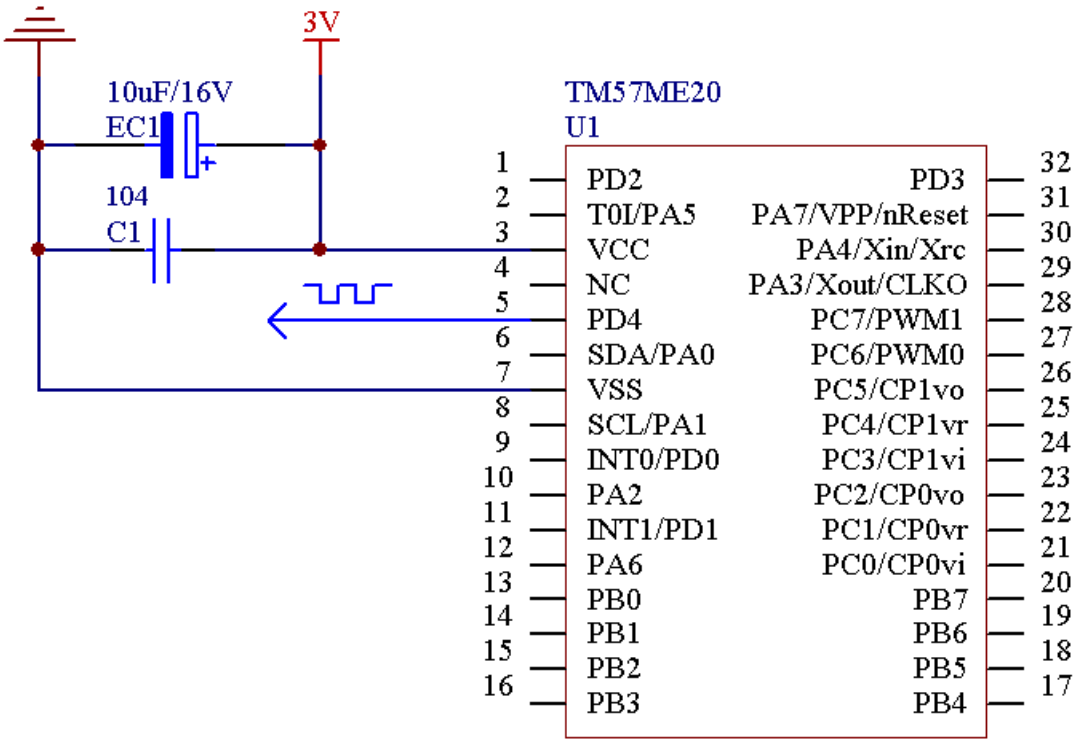
HWAUTO 地址为 R 区 0BH。

开启 TM2 中断的指令如下：

```
movlw    clr_TM2I     ;B'11111011'
movwf    INTI         ;清 TM2 中断标志
movlw    TM2IE_en     ;B'00000100'
movwf    INTE         ;开启 TM2 中断
```

INTI 地址为 F 区 09H，INTE 地址为 F 区 08H。

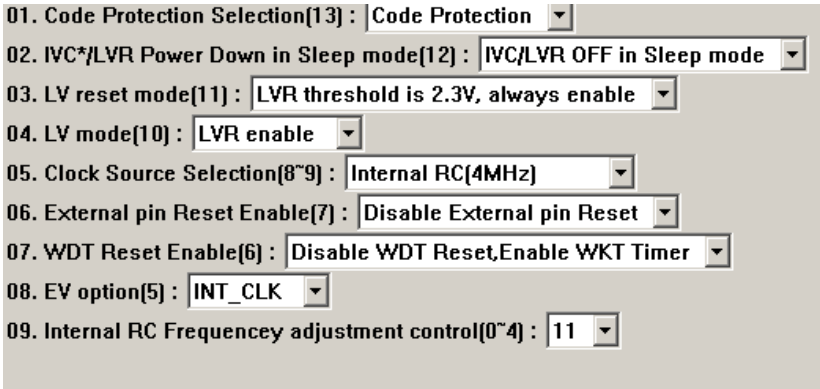
3. 线路图



09. 唤醒时钟功能的使用方法

1. 具体 DEMO 程序请参考 TM57ME20_WKT.ASM
2. 设置方法（内部 RC4MHz 时钟）

   1) 在 ICE 中 Config 的 07 项设定，即 WDTE （BIT6）必须设为 0，也就是 Disable WDT Reset, Enable WKT Timer，如下图 07 项所示：



   2) 考虑到耗电问题，系统进入休眠前尽可能将所有 IO 设置为非耗电状态，如果是空接，那么将其设置为输出 0 状态。

   3) 设定唤醒周期和使能自动保护现场模式（进中断时硬件自动备份 Acc 和 Status，并在退出中断时自动恢复）。其中唤醒周期由 HWAUTO（R 区 0BH）的 Bit[1:0]决定，有 4 钟选择：

      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

      这里选择 30ms

   而使能自动保护现场模式则与 HWAUTO（R 区 0BH）的 Bit7 有关，
   由于都使用同一寄存器 HWAUTO，因此指令可以放在一起执行：

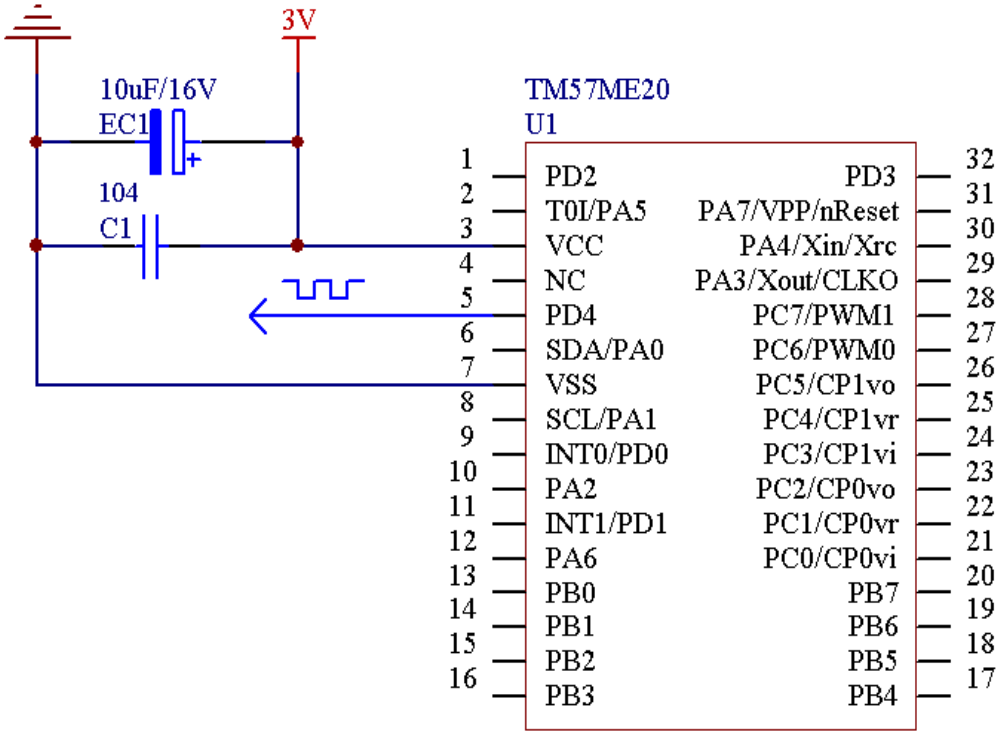
```
Movlw     HWAUTO_en       ;B'10000000'
        +WKTPSC_30mS      ;30ms 唤醒一次
Movwr     HWAUTO          ; HWAUTO 地址为 R 区 0BH
```

这样系统将会每隔 30 ms 唤醒一次，并进入中断，且取反 PD4，如此反复，于是在 PD4 脚位上得到一个周期 T=30×2=60ms 的波形。开启 WKT 中断的指令如下：

```
movlw     clr_WKTI        ;B'11110111'
movwf     INTI            ;清 WKT 中断标志
movlw     WKTIE_en        ;B'00001000'
movwf     INTE            ;开启 WKT 中断
```

INTI 地址为 F 区 09H，INTE 地址为 F 区 08H。

3. 线路图



10. PORTB 端口功能的使用方法

1. 具体 DEMO 程序请参考 TM57ME20_WKUP.ASM

2. 设置方法（内部 RC4MHz 时钟）

1) IO 初始化。考虑到耗电问题，系统进入休眠前尽可能将所有 IO 设置为非耗电状态，如果是空接，那么将其设置为输出 0 状态。

2) 将按键输入的脚位对应 IO 设为 Input 状态，并开启上拉。设置方法如下：

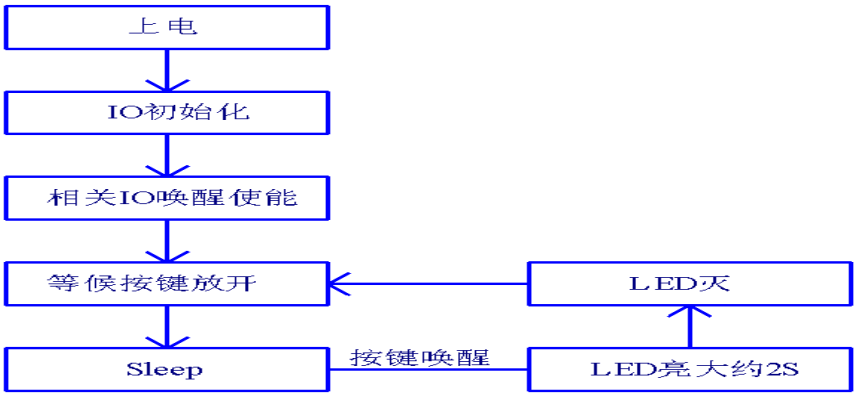
```
movlw    c0h
movwr    PBPU        ;PB0~PB5 上拉开启
movlw    c0h
movwr    PBE         ;PB0~PB5 设为 input
movlw    3fh
movwf    PBD
```

PBPU 和 PBE 的地址分别对应 R 区的 06H 和 09H，PBD 则是 F 区的 06H。

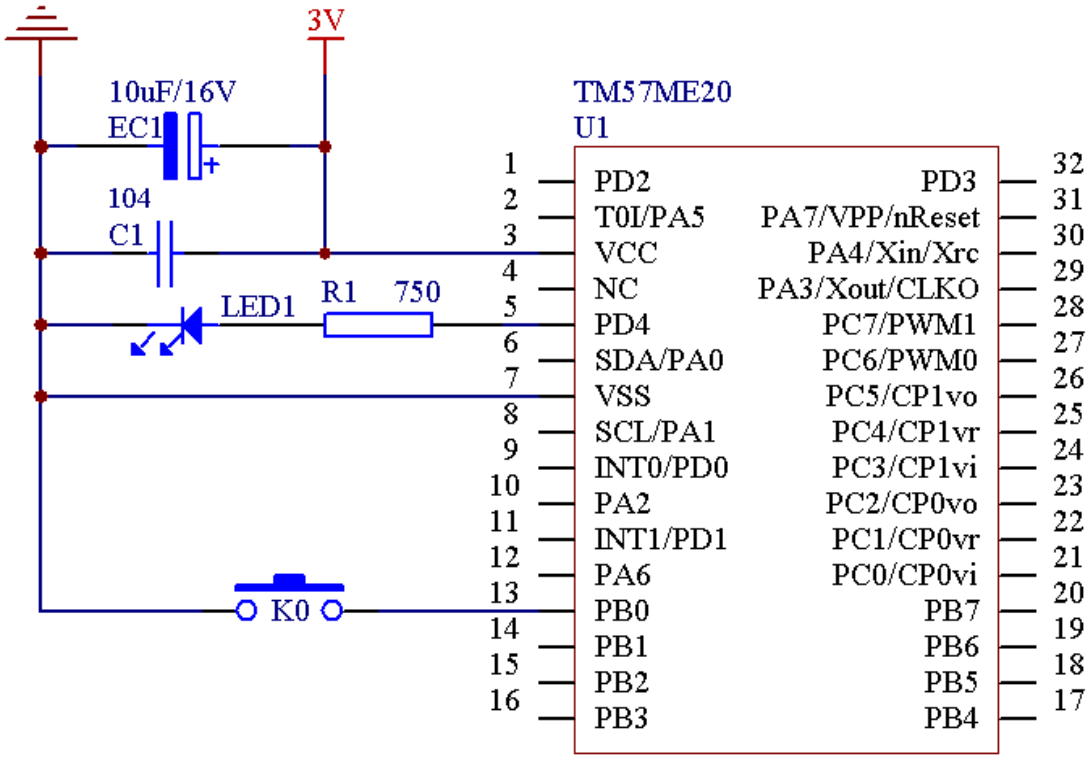
3) 使能唤醒功能。这里仅使能 PortB 端口 PB0～PB5 的唤醒功能。其指令方式如下：

```
movlw    3fh
movwr    PBWKUP      ; PBWKUP 为 R 区的 14H
```

4) 本程序的动作流程如下：



3. 线路图


3V
10uF/16V
EC1
104
C1
LED1
R1
750
K0
TM57ME20
U1
1 PD2
2 T0I/PA5
3 VCC
4 NC
5 PD4
6 SDA/PA0
7 VSS
8 SCL/PA1
9 INT0/PD0
10 PA2
11 INT1/PD1
12 PA6
13 PB0
14 PB1
15 PB2
16 PB3
32 PD3
31 PA7/VPP/nReset
30 PA4/Xin/Xrc
29 PA3/Xout/CLKO
28 PC7/PWM1
27 PC6/PWM0
26 PC5/CP1vo
25 PC4/CP1vr
24 PC3/CP1vi
23 PC2/CP0vo
22 PC1/CP0vr
21 PC0/CP0vi
20 PB7
19 PB6
18 PB5
17 PB4