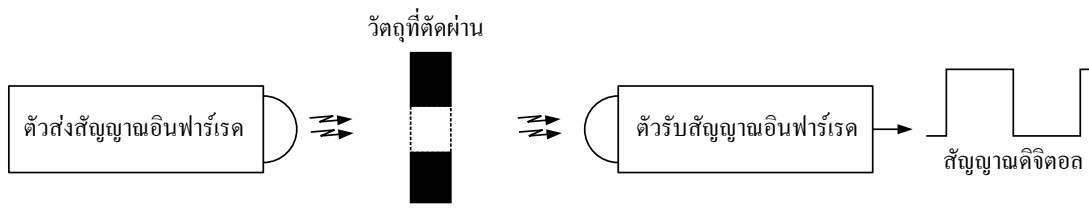
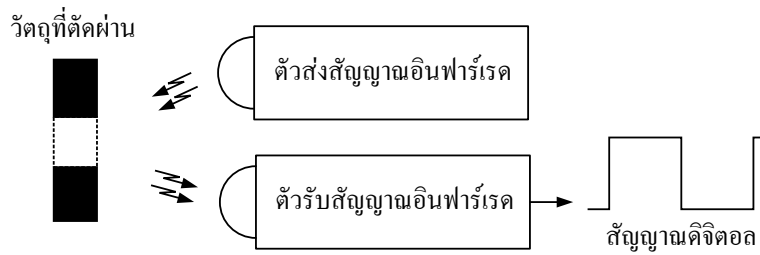


วงจรสวิตช์ด้วยแสงอินฟราเรด



(ก)แบบรับส่งโดยตรง



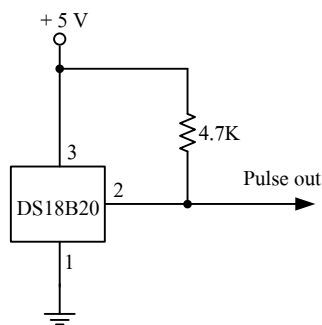
(ข)แบบรับการสะท้อนของคลื่น

รูปที่ 4.1 ลักษณะการติดตั้งตัวรับส่งสัญญาณอินฟราเรดที่ใช้ตัวสอบวัตถุ

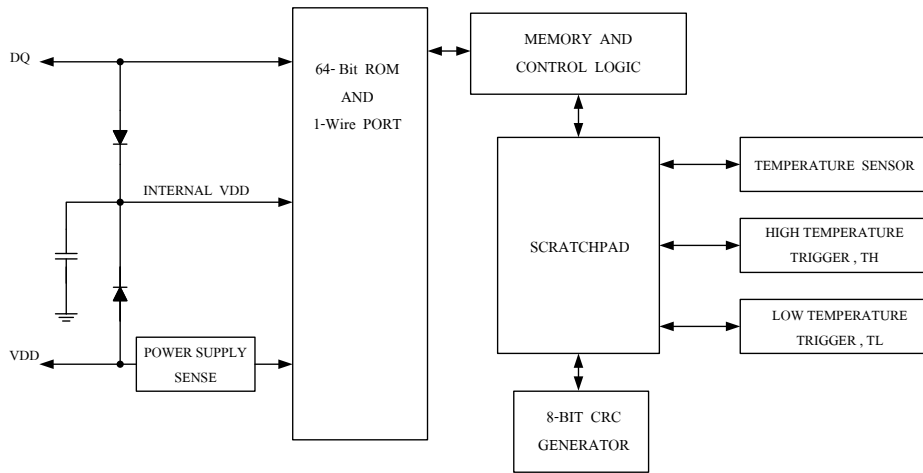
วงจรตรวจจับอุณหภูมิแบบดิจิทัลที่ใช้การสื่อสารแบบ 1-Wire



รูปที่ 4.2 โครงสร้างของตัวตรวจวัดอุณหภูมิเบอร์ DS18B20



รูปที่ 4.3 การต่อใช้งาน DS18B20



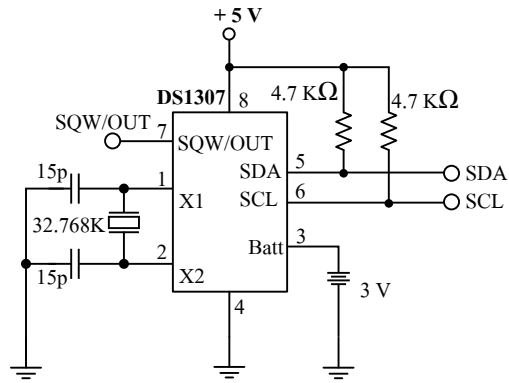
รูปที่ 4.4 โครงสร้างภายในไอซีตรวจวัดอุณหภูมิเบอร์ DS18B20
 สูตรที่ใช้คำนวณค่าของอุณหภูมิที่วัดได้สามารถหาได้จาก

$$\text{Temperature} = \text{Temp_read} - 025 + \frac{(\text{Count_per_c}) - (\text{Count_remain})}{\text{Count_per_c}}$$

วงจรสร้างฐานเวลาจริง (RTC)

ตารางที่ 4.1 โครงสร้างภายในไอซี DS1307 ในส่วนรายละเอียดของแต่ละบิต

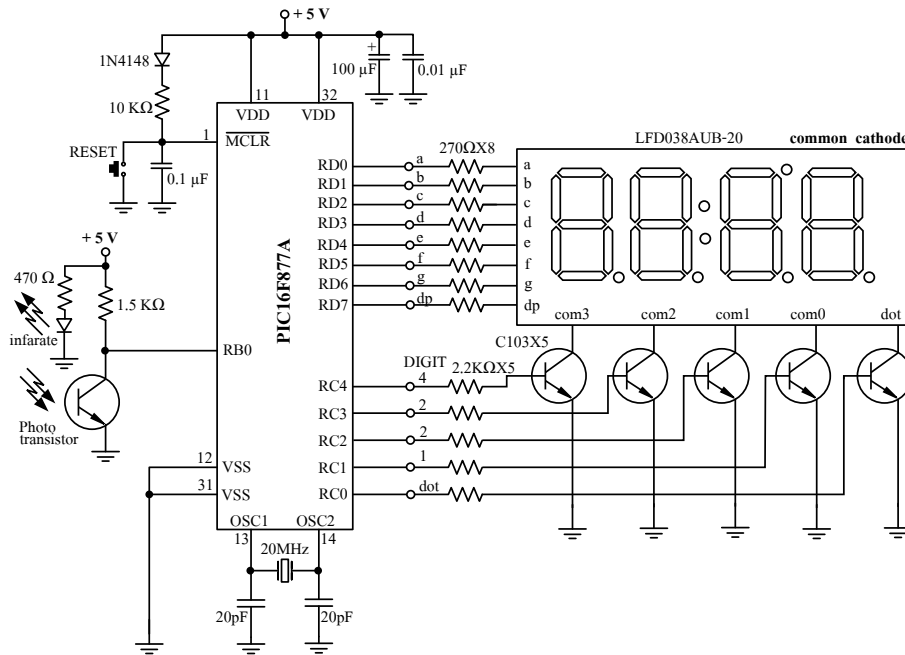
	B7	B6	B5	B4	B3	B2	B1	B0	
Address									
00H	CH	10 SECONDS		SECONDS				00 - 59	
01H	0	10 MINUTES		MINUTES				00 - 59	
02H	0	12 / 24	10HR / A/P	10 HR	HOURS			01 - 12 00 - 23	
03H	0	0	0	0	0	DAY			
04H	0	0	10 DATE		DATE			01 - 28/29 01 - 30 01 - 31	
05H	0	0	0	10 MONTH	MONTH			01 - 12	
06H	10 YEAR			YEAR				00 - 99	
07H	OUT	0	0	SQWE	0	0	RS1	RS0	



รูปที่ 4.5 วงจรที่สมบูรณ์ของวงจรสร้างฐานเวลาจริง

ลำดับขั้นการทดลอง

1. ต่อวงจรในชุดทดลองดังรูปที่ 4.5



รูปที่ 4.6 วงจรการทดลองของข้อที่ 1 อ่านข้อมูลจากสัญญาณระดับดิจิทัล

2. เขียนโปรแกรมข้างล่างลงในคอมไพเลอร์ แล้วบันทึกเป็นชื่อ DIGITAL1.pbp

```
@ DEVICE PIC16F877A, HS_OSC, LVP_OFF, BOD_OFF, WDT_OFF
```

```
DEFINE OSC 20
```

```
TRISD = %00000000
```

```
TRISC = %00000000
```

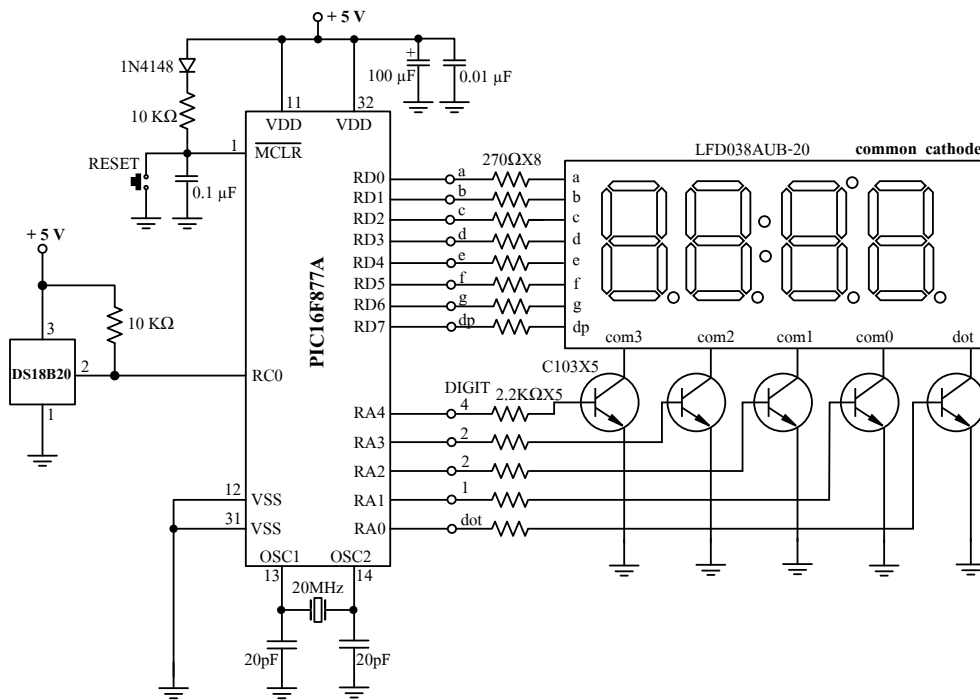
```
TRISB = %11111111
```

```

sign  VAR  PORTB.0
I  VAR  BYTE
N  VAR  WORD
      M  VAR  BYTE
      N  =  0
      ADCON1 = 7
MAIN:  GOSUB  SENSOR
      GOSUB  DISP
      GOTO  MAIN
      END
SENSOR:  IF(sign = 1) Then  SENSOR
      Pause 5
      N = N + 1
loop:  IF (sign = 0 )Then  loop
      Pause 5
      Return
DISP:  ADCON1 = 7
      For i = 1 TO 100
          M = N DIG 0
          Lookup M,[$3f,$06,$5b,$4f,$66,$6d,$7d,$07,$7f,$6f],PORTD
          PORTC = $02
          M = N DIG 1
          Lookup M,[$3f,$06,$5b,$4f,$66,$6d,$7d,$07,$7f,$6f],PORTD
          PORTC = $04
          M = N DIG 2
          Lookup M,[$bf,$86,$b,$cf,$e6,$ed,$fd,$87,$ef,$cf],PORTD
          PORTC = $08
          M = N DIG 3
          Lookup M,[$3f,$06,$5b,$4f,$66,$6d,$7d,$07,$7f,$6f],PORTD
          PORTC = $00
      Next i
      Return

```

3. ต่อวงจรตามรูปที่ 4.7



รูปที่ 4.7 วงจรการทดลองของข้อที่ 3 อ่านข้อมูลจากตัวตรวจวัดอุณหภูมิแสดงผลที่ LED 7 Segment

4. เขียนโปรแกรมข้างล่างลงในคอมพิวเตอร์ แล้วบันทึกเป็นชื่อ DIGITAL2.pbp

```
@ DEVICE PIC16F877A, HS_OSC, LVP_OFF, BOD_OFF, WDT_OFF
```

```
DEFINE OSC 20
```

```
TRISD = %00000000
```

```
TRISC = %11111111
```

```
TRISA = %00000000
```

```
temp VAR WORD
```

```
cre VAR BYTE
```

```
cpe VAR BYTE
```

```
DQ VAR PORTC.0
```

```
I VAR BYTE
```

```
N VAR WORD
```

```
M VAR BYTE
```

```
N = 0
```

```
ADCON1 = 7
```

```
MAIN: GOSUB readtemp
```

```
GOSUB DISP
```

```

GOTO MAIN
END
readtemp:    OWOut DQ,1,[$cc,$44]
waitloop:    OWIn DQ,4,[cre]
            OWOut DQ,1,[$cc,$be]
            OWIn DQ,0,[temp.LOWBYTE,temp.HIGHBYTE,skip 4,cre,cpe]
            temp = (((temp>>1)*100)-25)+(((cpe-cre)*100)/cpe)
            N = temp/100
            Return
DISP:  For i = 1 TO 100
            PORTD = $04
            PORTA = $01
            PORTD = $58
            PORTA = $02
            M = N DIG 0
            Lookup M,[$3f,$06,$5b,$4f,$66,$6d,$7d,$07,$7f,$6f],PORTD
            PORTA = $04
            M = N DIG 1
            Lookup M,[$bf,$86,$b,$cf,$e6,$ed,$fd,$87,$ef,$cf],PORTD
            PORTA = $08
            M = N DIG 2
            Lookup M,[$3f,$06,$5b,$4f,$66,$6d,$7d,$07,$7f,$6f],PORTD
            PORTA = $10
        Next i
    Return

```

5. เขียนโปรแกรมข้างล่างลงในคอมพิวเตอร์ แล้วบันทึกเป็นชื่อ DIGITAL3.pbp

```
@ DEVICE PIC16F877A, HS_OSC, LVP_OFF, BOD_OFF, WDT_OFF
DEFINE OSC 20
TRISD = %00000000
TRISC = %11111111
TRISA = %00000000
temp  VAR  WORD
cre    VAR  BYTE
cpe    VAR  BYTE
DQ     VAR  PORTC.0
I VAR BYTE
N VAR WORD
      M VAR BYTE
      N = 0
      ADCON1 = 7
MAIN:  GOSUB readtemp
      GOSUB DISP
      GOTO MAIN
      END
readtemp:  OWOut DQ,1,[$cc,$44]
waitloop:  OWIn DQ,4,[cre]
      OWOut DQ,1,[$cc,$be]
      OWIn DQ,0,[temp.LOWBYTE,temp.HIGHBYTE,skip 4,cre,cpe]
      temp = (((temp>>1)*100)-25)+(((cpe-cre)*100)/cpe)
      N = (temp*/461)+3200
      N = N/10
      Return
DISP:  For i = 1 TO 100
      PORTD = $04
      PORTA = $01
      PORTD = $79
      PORTA = $02
```

M = N DIG 0

```
Lookup M,[$3f,$06,$5b,$4f,$66,$6d,$7d,$07,$7f,$6f],PORTD
```

PORTA = \$04

M = N DIG 1

```
Lookup M,[$bf,$86,$b,$cf,$e6,$ed,$fd,$87,$ef,$cf],PORTD
```

PORTA = \$08

M = N DIG 2

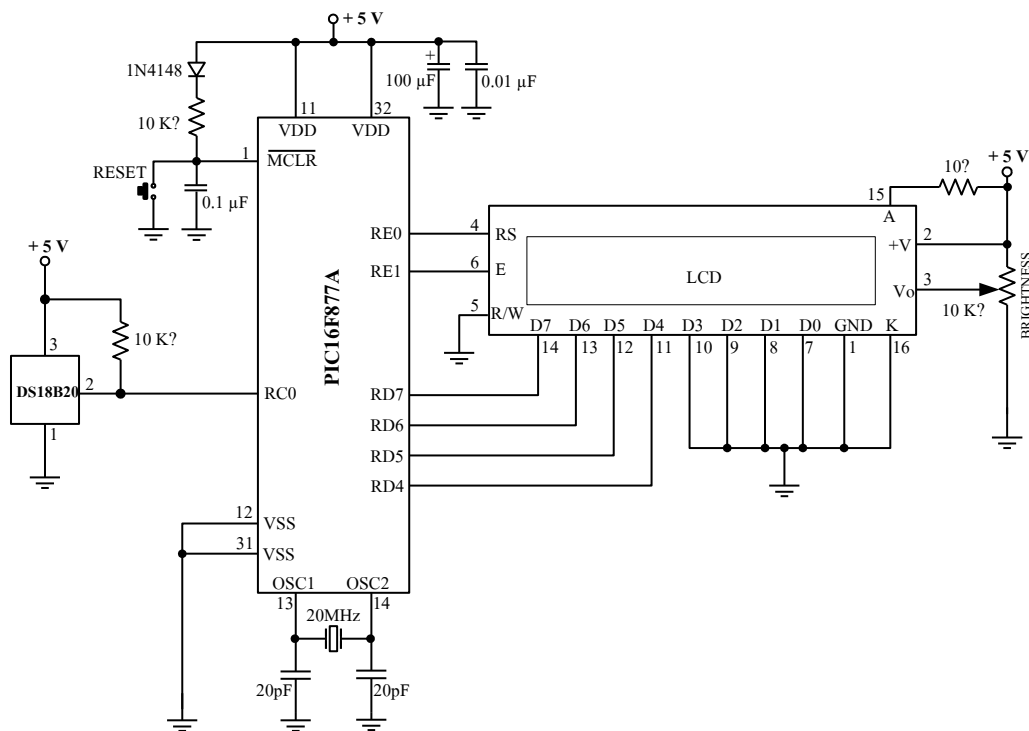
```
Lookup M,[$3f,$06,$5b,$4f,$66,$6d,$7d,$07,$7f,$6f],PORTD
```

PORTA = \$10

Next i

Return

6. ต่อดตามรูปที่ 4.8



รูปที่ 4.8 วงจรการทดลองของข้อที่ 6 อ่านข้อมูลจากตัวตรวจวัดอุณหภูมิแสดงผลที่ LCD

7. เขียนโปรแกรมข้างล่างลงในคอมพิวเตอร์ แล้วบันทึกเป็นชื่อ DIGITAL4.pbp

```
@ DEVICE PIC16F877A, HS_OSC, LVP_OFF, BOD_OFF, WDT_OFF
```

```
DEFINE OSC 20
```

```
DEFINE LCD_DREG PORTD
```

```
DEFINE LCD_DBIT 4
```

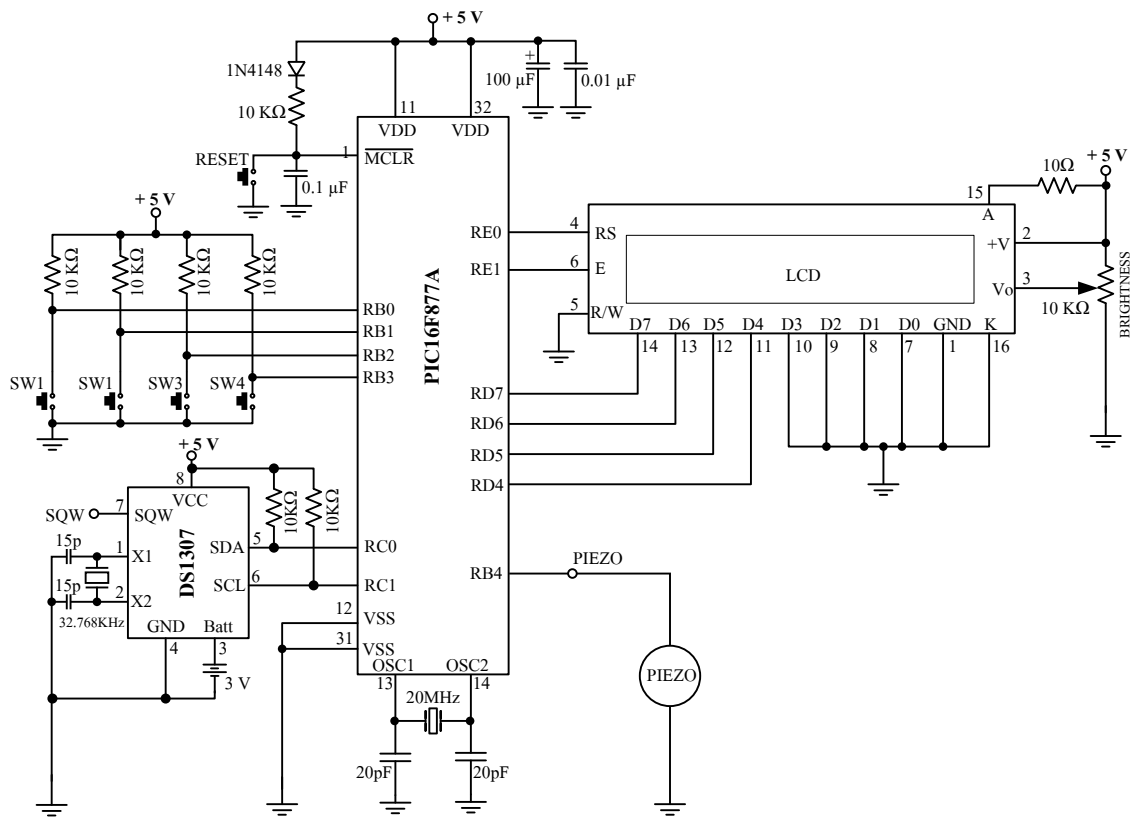


```

DEFINE LCD_RSREG PORTE
DEFINE LCD_RSBIT 2
DEFINE LCD_EREG PORTE
    DEFINE LCD_EBIT 1
    digit    VAR BYTE
    temp     VAR WORD
    temp1    VAR WORD
    temp2    VAR WORD
    cre VAR BYTE
    cpe VAR BYTE
    DQ VAR PORTC.0
    ADCON1 = 7
main:  GoSub readtemp
      GoSub disp
      GoTo main
      End
readtemp:  OWOut DQ,1,[$cc,$44]
waitloop:  OWIn DQ,4,[cre]
           OWOut DQ,1,[$cc,$be]
           OWIn DQ,0,[temp.LOWBYTE,temp.HIGHBYTE,skip 4,cre,cpe]
           temp = (((temp>>1)*100)-25)+(((cpe-cre)*100)/cpe)
           temp1 = temp/10
           temp2 = (temp*/461)+3200
           Return
disp:  LCDOut $fe,1,"TEMP = ",DEC(temp1/100),".",DEC2(temp1//100)," C"
      LCDOut $fe,$c0,"TEMP = ",DEC(temp2/100),".",DEC2(temp2//100)," F"
      Return

```

8. ต่อตามรูปที่ 4.9



รูปที่ 4.9 วงจรการทดลองของข้อที่ 8 อ่านข้อมูลจากตัวกำเนิดฐานเวลาจริงแสดงผลที่ LCD

9. เขียนโปรแกรมข้างล่างลงในคอมพิวเตอร์ แล้วบันทึกเป็นชื่อ DIGITAL5.pbp

@ DEVICE PIC16F877A , HS_OSC, LVP_OFF, BOD_OFF, WDT_OFF

DEFINE OSC 20

DEFINE LCD_DREG PORTD

DEFINE LCD_DBIT 4

DEFINE LCD_RSREG PORTE

DEFINE LCD_RSBIT 1

DEFINE LCD_EREG PORTE

DEFINE LCD_EBIT 0

TRISD = \$00

SW1 VAR PORTB.0

SW2 VAR PORTB.1

SW3 VAR PORTB.2

```

SW4  VAR PORTB.3
SPK  VAR PORTB.4
address con $00
sec      VAR byte
minute  VAR BYTE
hour    VAR BYTE
day     VAR BYTE
date    VAR BYTE
month   VAR BYTE
year    VAR BYTE
d0 VAR BYTE
d1 VAR BYTE
hr  VAR BYTE
mn      VAR BYTE
ss  VAR BYTE
sda      VAR PORTC.0
scl VAR PORTC.1
cont CON %11010000

```

```
'----- Initial -----'
```

```
    TRISB = %11101111
```

```
adcon1 = 7
```

```
hr = 0
```

```
mn      = 0
```

```
ss = 0
```

```
'----- main program -----'
```

```

start:  if (SW1 = 0 and SW2 = 0) then
        pause 2000
        if (SW1 = 0 and SW2 = 0) then
            gosub beep1
            pause 1000
            gosub setting
        endif

```

```

endif

    gosub read_time

    gosub display_time

    goto start

end

'----- End of main Program -----

setting: lcdout $fe,1,"set time"

    pause 2000

    lcdout $fe,1

    hour = 0

    minute = 0

    sec = 0

    gosub display_time

    gosub set_time

    return

set_time:  pause 500

loop2:    if sw1 = 1 then

            pause 50

            if (SW2=0) then

                gosub click

            pause 50

            hr = hr+1

            if hr > 23 then hr = 0

            d0 = hr dig 0

            d1 = hr dig 1

            d1 = d1<< 4

            hour = d1|d0

            gosub display_time

            pause 200

        endif

        if (sw3=0) then

            gosub click

```

```

        pause 50
        mn = mn+1
        if mn > 59 then mn = 0
        d0 = mn dig 0
        d1 = mn dig 1
        d1 = d1<< 4
        minute = d1|d0
        gosub display_time
        pause 150
endif
if (SW4 = 0) then
    gosub click
    pause 50
    ss = ss+1
    if ss > 59 then ss = 0
    d0 = ss dig 0
    d1 = ss dig 1
    d1 = d1<< 4
    sec = d1|d0
    gosub display_time
    pause 150
endif
goto loop2
endif
gosub write_time
return
write_time:    i2cwrite sda,scl,cont,address,[sec,minute,hour]
               return
read_time:    i2cread sda,scl,cont,address,[sec,minute,hour]
              return
display_time:  lcdout $fe,$80,hex2 hour,":",$fe,$84,hex2 minute,":"
               lcdout $fe,$c0,hex2 sec

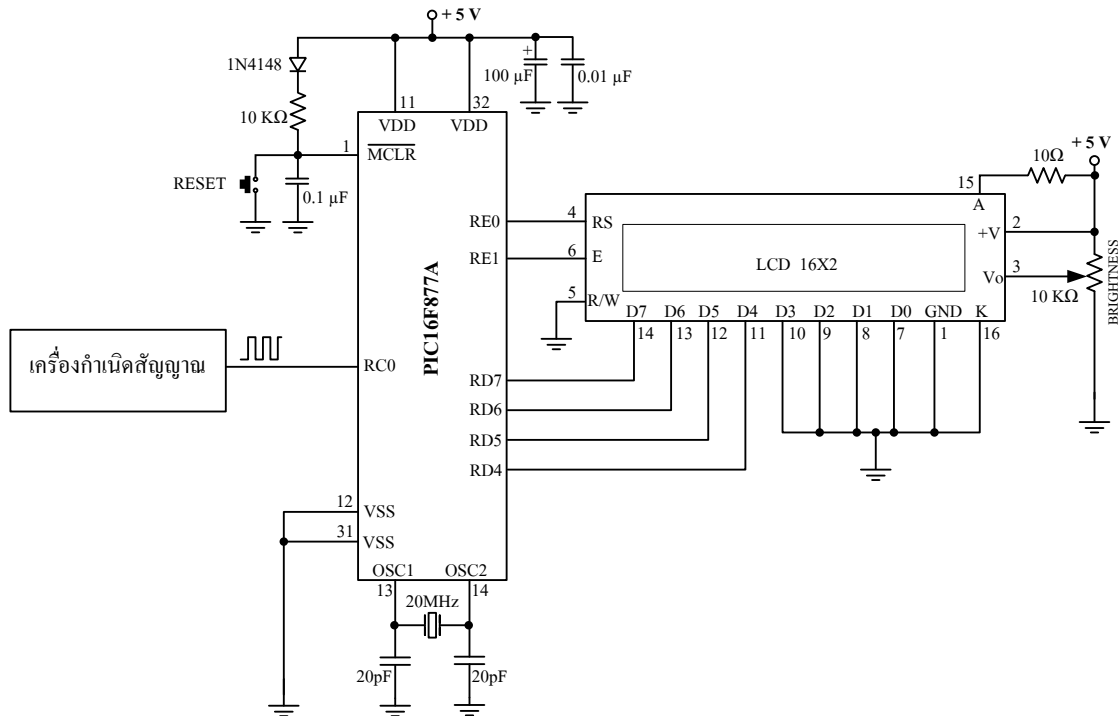
```

```

return
beep1:    freqout spk,20,2000
          pause 200
          freqout spk,20,2000
          pause 100
          return
click:    freqout spk,5,2000
          return

```

10. ต่อดตามรูปที่ 4.10



รูปที่ 4.10 วงจรอ่านค่าความถี่แสดงผลที่ LCD

11. เขียนโปรแกรมข้างล่างลงในคอมพิวเตอร์ แล้วบันทึกเป็นชื่อ DIGITAL6.pbp

```

@ DEVICE PIC16F877A , HS_OSC, LVP_OFF, BOD_OFF, WDT_OFF
DEFINE OSC 20
DEFINE LCD_DREG PORTD
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTE
DEFINE LCD_RSBIT 0

```

```
DEFINE LCD_EREG PORTE
```

```
DEFINE LCD_EBIT 1
```

```
TRISD = $00
```

```
F VAR PORTC.0
```

```
N VAR WORD
```

```
ADCON1 = 7
```

```
MAIN: GOSUB MEANS
```

```
      GOSUB DISP
```

```
      GOTO MAIN
```

```
      END
```

```
MEANS: COUNT F,200,N
```

```
      N = N*5
```

```
      RETURN
```

```
DISP:  LCDOut $fe,1,"FREQ = ", DEC N," Hz"
```

```
      PAUSE 50
```

```
      RETUR
```