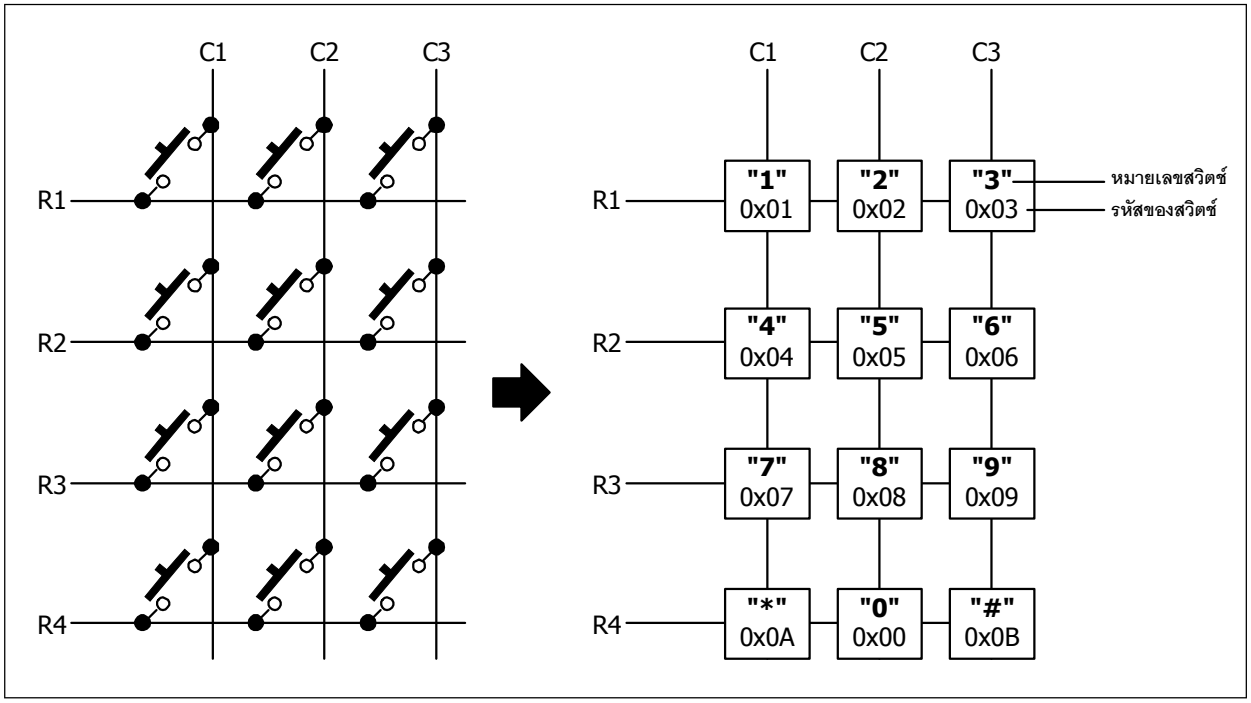
บทที่ 19

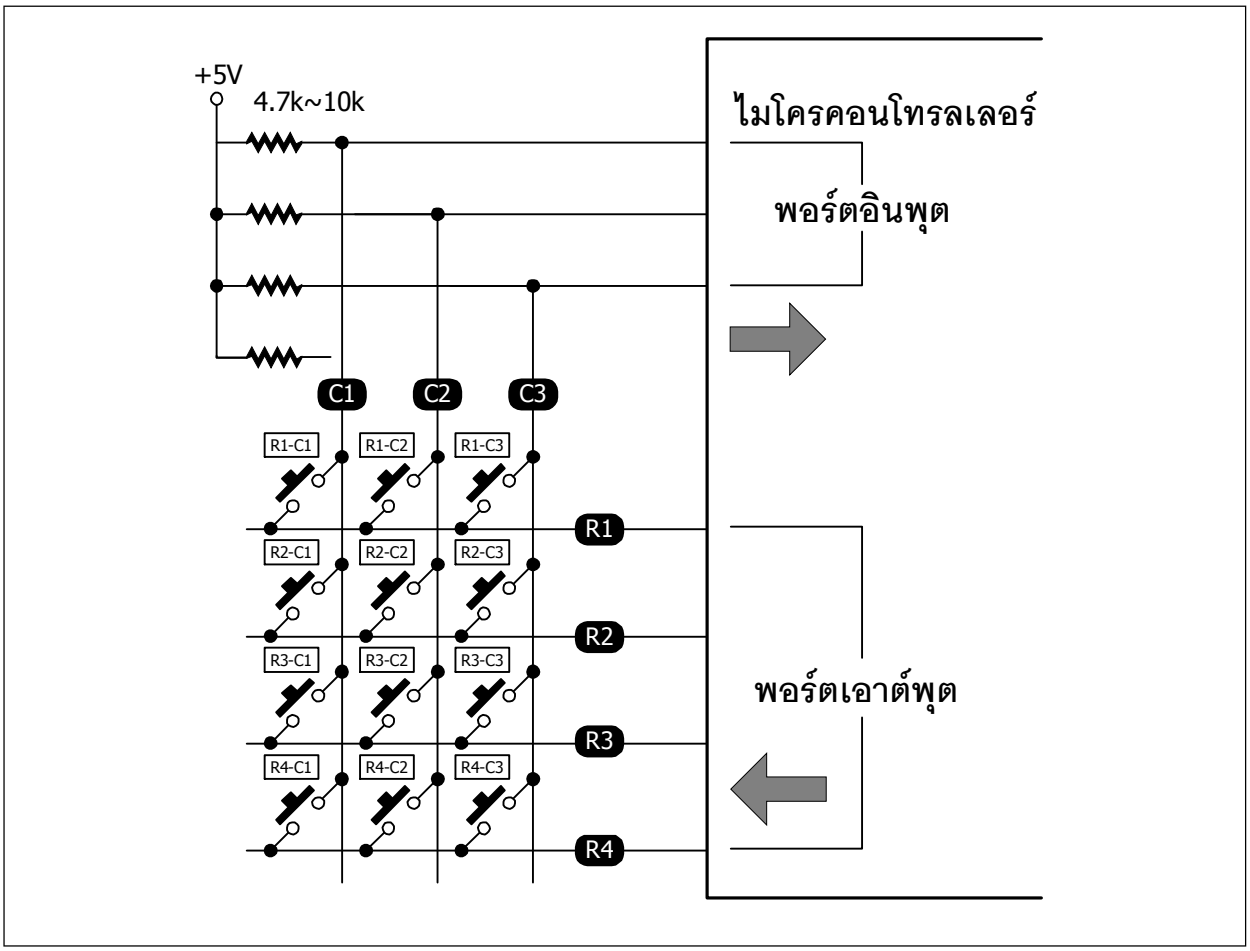
การติดต่อสวิตช์เมตริกซ์ของไมโครคอนโทรลเลอร์ PIC16F887

สวิตช์เมตริกซ์ (matrix switch)หรือคีย์แพด (keypad) ถูกนำมาใช้เป็นอุปกรณ์ในการป้อนข้อมูลให้กับงานทางด้านไมโครคอนโทรลเลอร์ นอกเหนือจากสวิตช์กดติดปล่อยดับแบบธรรมดา (push button switch) โดยเฉพาะกับงานที่ต้องมีการป้อนข้อมูลทั้งตัวอักษรและตัวเลขและมีสวิตช์จำนวนมากแล้ว สวิตช์เมตริกซ์จะเป็นตัวที่ถูกเลือกใช้งานเสมอ สวิตช์ในรูปแบบเมตริกซ์ที่เห็นได้ในชีวิตประจำวัน เช่น คีย์กดตัวเลขของระบบโทรศัพท์ เป็นต้น

การต่อใช้งานสวิตช์แบบเมตริกซ์เป็นการนำสวิตช์ธรรมดามาต่อกันในแบบเมตริกซ์ คือ ขาด้านหนึ่งจะต่อในแนวหลัก (column) และขาอีกด้านหนึ่งจะต่ออยู่ในแนวแถว (row) แสดงดังรูปที่ 19-1



รูปที่ 19–1 แสดงการกำหนดค่ารหัสของสวิตช์เมตริกซ์หรือคีย์แพดแบบ 4x3 จุด ซึ่งใช้ในการทดลอง โดยหมายเลขสวิตช์สามารถเปลี่ยนแปลงได้ตามที่ผู้ใช้งานต้องการ



รูปที่ 19-2 วงจรเชื่อมต่อสวิตช์เมตริกซ์ขนาด 4x3 จุดเข้ากับไมโครคอนโทรลเลอร์

19.1 กำหนดรหัสประจำตำแหน่งของสวิตช์เมตริกซ์หรือคีย์แพด

หลักในการอ่านค่าจากคีย์แพดนี้คือ จะต้องกำหนดรหัสประจำตำแหน่งของสวิตช์แต่ละตัวไว้ไม่ให้ซ้ำกัน ดังนั้นเมื่อสวิตช์ตัวใดถูกกดก็จะได้ค่ารหัสของสวิตช์ตัวดังกล่าวออกมา ซึ่งในการทดลองนี้จะกำหนดค่าคงที่ให้กับสวิตช์แต่ละตัวไว้ดังนี้

สวิตช์ตำแหน่ง R1 (หมายถึง row 1 หรือแถวที่ 1), C1 (หมายถึง column 1 หรือหลักที่ 1) มีค่า 0x01

สวิตช์ตำแหน่ง R1 (หมายถึง row 1 หรือแถวที่ 1), C2 (หมายถึง column 2 หรือหลักที่ 2) มีค่า 0x02

สวิตช์ตำแหน่ง R1 (หมายถึง row 1 หรือแถวที่ 1), C3 (หมายถึง column 2 หรือหลักที่ 2) มีค่า 0x03

สวิตช์ตำแหน่ง R2 (หมายถึง row 2 หรือแถวที่ 2), C1 (หมายถึง column 1 หรือหลักที่ 1) มีค่า 0x04

สวิตช์ตำแหน่ง R2 (หมายถึง row 2 หรือแถวที่ 2), C2 (หมายถึง column 2 หรือหลักที่ 2) มีค่า 0x05

สวิตช์ตำแหน่ง R2 (หมายถึง row 2 หรือแถวที่ 2), C3 (หมายถึง column 2 หรือหลักที่ 2) มีค่า 0x06

สวิตช์ตำแหน่ง R3 (หมายถึง row 3 หรือแถวที่ 3), C1 (หมายถึง column 1 หรือหลักที่ 1) มีค่า 0x07

สวิตช์ตำแหน่ง R3 (หมายถึง row 3 หรือแถวที่ 3), C2 (หมายถึง column 2 หรือหลักที่ 2) มีค่า 0x08

สวิตช์ตำแหน่ง R3 (หมายถึง row 3 หรือแถวที่ 3), C3 (หมายถึง column 2 หรือหลักที่ 2) มีค่า 0x09

สวิตช์ตำแหน่ง R4 (หมายถึง row 4 หรือแถวที่ 4), C1 (หมายถึง column 1 หรือหลักที่ 1) มีค่า 0x10

สวิตช์ตำแหน่ง R4 (หมายถึง row 4 หรือแถวที่ 4), C2 (หมายถึง column 2 หรือหลักที่ 2) มีค่า 0x00

สวิตช์ตำแหน่ง R4 (หมายถึง row 4 หรือแถวที่ 4), C3 (หมายถึง column 2 หรือหลักที่ 2) มีค่า 0x11

19.2 การอ่านค่าสวิตช์เมตริกซ์หรือคีย์แพด

สำหรับการทดลองนี้ต้องการอ่านค่าสวิตช์เมตริกซ์ในตำแหน่ง 0 ถึง 9 ออกมาแสดงผลที่ LED ตัวเลข 7 ส่วน วิธีการอ่านค่าจากสวิตช์ในการทดลองนี้สรุปได้ดังนี้

(1) กำหนดให้ขาพอร์ต RB5 ถึง RB7 เป็นพอร์ตอินพุตดิจิตอล (ดูรูปที่ 19-2 ประกอบ)

(2) กำหนดให้ขาพอร์ต RB1 ถึง RB4 เป็นพอร์ตเอาต์พุตดิจิตอล (ดูรูปที่ 19-2 ประกอบ)

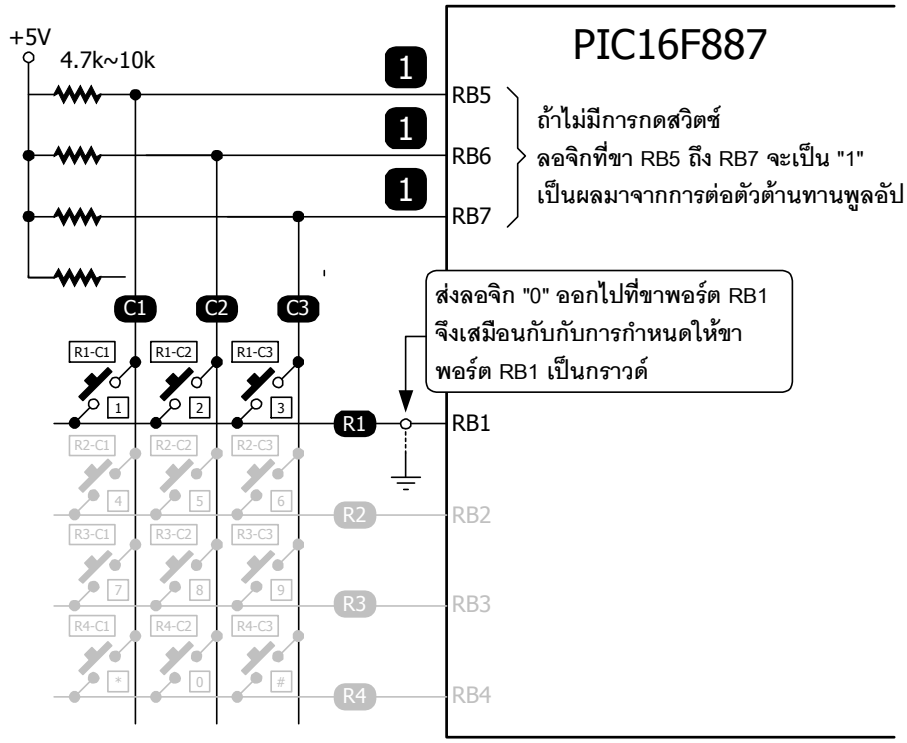
(3) ส่งลอจิก “1” ออกไปยังขาพอร์ต RB1 ถึง RB4

(4) จากนั้นเริ่มต้นการตรวจสอบแถวที่ 1 ด้วยการทำให้ขา RB1 เป็น “0” แล้วอ่านค่าจากขาพอร์ต RB5 ถึง RB7 ว่าขาใดเป็นลอจิก “0” หรือไม่ หากขาใดเป็นลอจิก “0” นั่นคือ เกิดการกดสวิตช์ที่ขานั้น หากขา RB5 เป็น “0” นั่นคือเกิดการกดสวิตช์ที่ตำแหน่ง R1, C1 ซึ่งก็คือ สวิตช์เลข 1 โปรแกรมจะคืนค่าเป็น 0x01 กลับมา การวนรออ่านค่านี้จะใช้เวลา 600 มิลลิวินาที หากไม่มีการกดสวิตช์เลย จะเปลี่ยนการตรวจสอบไปยังแถวที่ 2

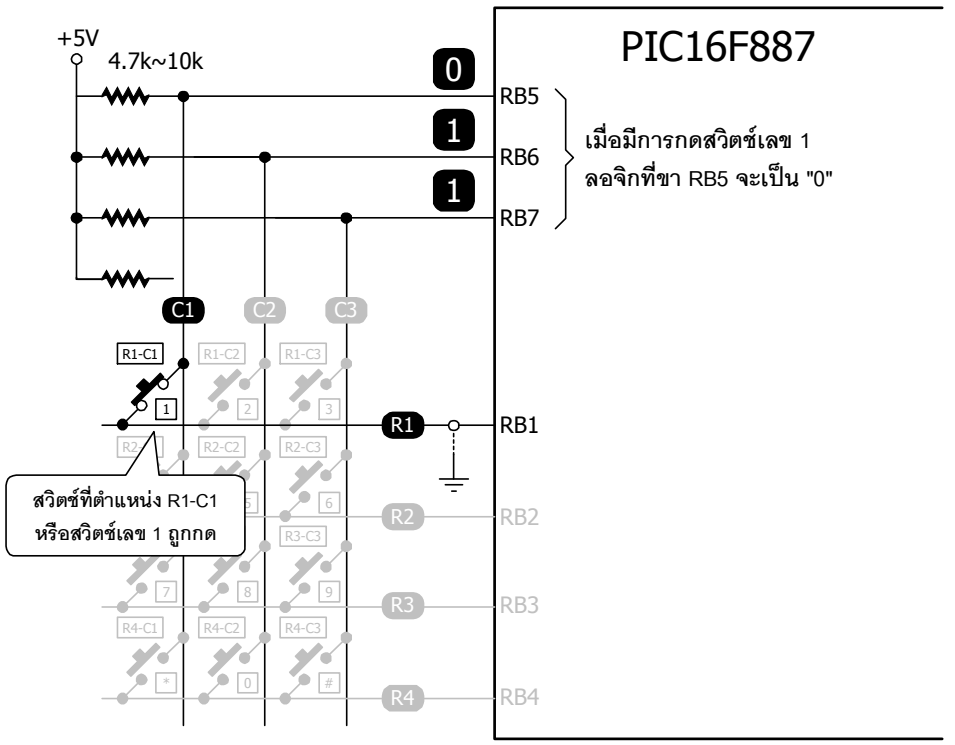
(5) การตรวจสอบแถวที่ 2 จะเกิดขึ้นเมื่อทำให้ขา RB1 กลับมาเป็น “1” แล้วทำให้ขา RB2 เป็น “0” จากนั้นรออ่านค่าจากขาพอร์ต RB5 ถึง RB7 เช่นเดิม หากในคราวนี้ที่ขา RB5 เป็น “0” นั่นหมายถึง เกิดการกดสวิตช์ที่ตำแหน่ง R2, C1 ซึ่งก็คือ สวิตช์เลข 4 โปรแกรมจะคืนค่าเป็น 0x04 กลับมา หากไม่มีการกดสวิตช์เลย จะเปลี่ยนการตรวจสอบไปยังแถวที่ 3 และ 4 แล้ววนกลับมายังแถวที่ 1 อีกครั้ง

(6) เมื่อการตรวจสอบมาถึงแถวที่ 4 จะพบว่า ต้องมีการตรวจสอบสวิตช์ 3 ตัวคือ *, 0 และ # ดังนั้นค่าที่ได้จากการตรวจสอบการกดสวิตช์จะเกิดได้ 3 ค่าคือ 10 หรือ 0x0A, 0 หรือ 0x00 และ 11 หรือ 0x0B ในการทดลองจะสนใจเพียงค่า 0x00 เท่านั้น จึงต้องมีการเขียนโปรแกรมเพื่อกรองเอาเฉพาะข้อมูลเลข 0 ถึง 9 อีกชั้นหนึ่ง

(7) เมื่อได้ค่าของสวิตช์แล้ว จึงนำค่าดังกล่าวไปใช้งานต่อได้ อาทิ นำข้อมูลไปใช้ขับ LED ธรรมดาหรือ LED ตัวเลข 7 ส่วน หรือนำไปแสดงที่โมดูล LCD หรือนำไปใช้ประมวลผลด้านเกี่ยวกับข้อมูลรหัสตัวเลขก็ได้

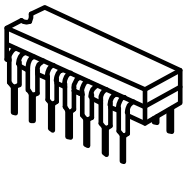


(ก) สถานะของขาพอร์ตเมื่อเริ่มต้นการตรวจสอบการกดสวิตช์เมตริกซ์แถวที่ 1



(ข) สถานะของขาพอร์ตเมื่อมีการกดสวิตช์เมตริกซ์ตำแหน่ง C1-R1

รูปที่ 19-3 แสดงสถานะการทำงานของการตรวจสอบการกดสวิตช์เมตริกซ์ของ PIC16F887



PIC16F887 microcontroller Experiment

ปฏิบัติการที่ 21 อ่านค่าสวิตช์เมตริกซ์ 4x3 จุดแสดงผลผ่าน LED ตัวเลข 7 ส่วน

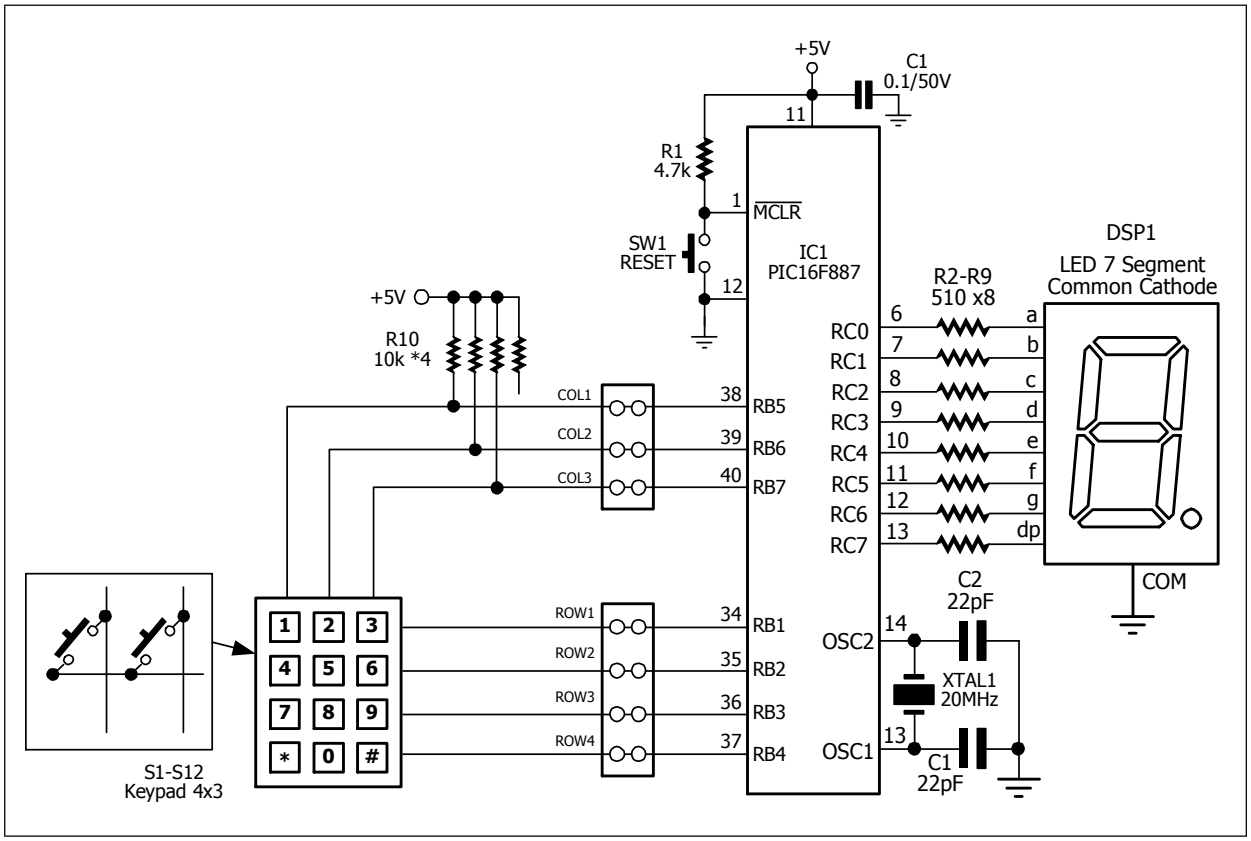
L21.1 ใช้วงจรในรูปที่ L21-1 ในการทดลอง เชื่อมต่อ LED ตัวเลข 7 ส่วนและสวิตช์เมตริกซ์เข้ากับวงจรให้เรียบร้อย

L21.2 สร้างไฟล์โปรเจ็กต์ชื่อ 1wire_01 เขียนโปรแกรมที่ L21-1 บันทึกเป็นไฟล์ keypad4x3.c

L21.3 คอมไพล์และดาวน์โหลดไฟล์ผลลัพธ์ไปยัง PIC16F887

L21.4 กดสวิตช์รีเซตหนึ่งครั้ง ทดลองกดที่สวิตช์เมตริกซ์ไล่เรียงลำดับทีละตัว สังเกตการแสดงผลที่ LED

*LED ตัวเลข 7 ส่วนจะแสดงผลตรงตามตำแหน่งของสวิตช์ 0 ถึง 9 ที่ถูกกด แต่ถ้ากดสวิตช์ * และ # จะไม่มีการแสดงผลใดๆ*



รูปที่ L27-1 วงจรทดลองอ่านค่าของสวิตช์เมตริกซ์เพื่อนำมาแสดงที่ LED ตัวเลข 7 ส่วนของ PIC16F877

```
/*
Hardware Configuration
*** 7-Segment : RC0-RC7 => segment a-dp
*** keypad 4x3 :
   RB1 => ROW1
   RB2 => ROW2
   RB3 => ROW3
   RB4 => ROW4

   RB5 => COL5 with resistor 10k pull up
   RB6 => COL6 with resistor 10k pull up
   RB7 => COL7 with resistor 10k pull up
*/
#include <pic.h>              // Include header file for MCU
#define _XTAL_FREQ  20000000 // Define Frequency 20.0 MHz for function __delay_ms
__CONFIG(HS & WDTDIS & LVPDIS);// Config. High speed clock,Disable watchdog and Disable LVP
const unsigned char  num[10]=
{
   0x3F,// Code number 0
   0x06,// Code number 1
   0x5B,// Code number 2
   0x4F,// Code number 3
   0x66,// Code number 4
   0x6D,// Code number 5
   0x7D,// Code number 6
   0x07,// Code number 7
   0x7F,// Code number 8
   0x6F // Code number 9
};
void delay(unsigned int tick)
{
   while(tick—)                     // Loop couter delay time
   {
     __delay_ms(1);                 // Delay 1 ms
   }
}
unsigned char get_keypad()          // Return 0-9 keypad ,* = 10,# = 11
{
   unsigned char result=0xFF;
   // R Pull up at colume 1,2,3
   ANS10 = 0;                       // RB1
   ANS8 = 0;                        // RB2
   ANS9 = 0;                        // RB3
   ANS11 = 0;                       // RB4
   ANS13 = 0;                       // RB5

   TRISB5 = 1;                      // RB5 colume 1
   TRISB6 = 1;                      // RB6 colume 2
   TRISB7 = 1;                      // RB7 colume 3

   TRISB1 = 0;                      // RB1 row 1
   TRISB2 = 0;                      // RB2 row 2
   TRISB3 = 0;                      // RB3 row 3
   TRISB4 = 0;                      // RB4 row 4
   RB1 = 1;
   RB2 = 1;
   RB3 = 1;
   RB4 = 1;
```

โปรแกรมที่ L21-1 วงจรทดลองการอ่านค่าสวิตช์เมตริกซ์ 4x3 จุดของ PIC16F887 (มีต่อ)

```
RB1 = 0;            // Scan row 1
if(RB5==0)
{
  result =1;
  delay(200);
}
else if(RB6==0)
{
  result =2;
  delay(200);
}
else if(RB7==0)
{
  result =3;
  delay(200);
}
RB1 = 1;            // end Scan row 1

RB2 = 0;            // Scan row 2
if(RB5==0)
{
  result =4;
  delay(200);
}
else if(RB6==0)
{
  result =5;
  delay(200);
}
else if(RB7==0)
{
  result =6;
  delay(200);
}
RB2 = 1;

RB3 = 0;            // end Scan row 2
if(RB5==0)
{
  result =7;
  delay(200);
}
else if(RB6==0)
{
  result =8;
  delay(200);
}
else if(RB7==0)
{
  result =9;
  delay(200);
}
RB3 = 1;            // end Scan row 3

RB4 = 0;            // Scan row 4
if(RB5==0)
{
  result =10;
  delay(200);
}
```

โปรแกรมที่ L21-1 วงจรทดลองการอ่านค่าสวิตช์เมตริกซ์ 4x3 จุดของ PIC16F887 (มีต่อ)

```
   else if(RB6==0)
   {
     result =0;
     delay(200);
   }
   else if(RB7==0)
   {
     result =11;
     delay(200);
   }
   RB4 = 1;                    // end Scan row 4
   return result;
}
void main()
{
   unsigned char key=0xFF;
   TRISC = 0x00;               // Set PORTC output mode
   PORTC = 0x00;               // Initial value
   while(1)                    // Infinite loop
   {
     key = get_keypad();       // Get keypad value
     if(key<10)                // Check key 0-9 pressed?
     {
        PORTC = num[key];      // Display value on 7-segment
     }
   }
}
```

โปรแกรมที่ L21-1 วงจรทดลองการอ่านค่าสวิตช์เมตริกซ์ 4x3 จุดของ PIC16F887 (จบ)