



# Tecnológico de Monterrey

Instituto Tecnológico de Estudios Superiores de Monterrey

## **Laboratorio de microcontroladores**

### Reporte de práctica 6

Gabriela Natalia Altamirano Cruz - A01411942

## Introducción

El objetivo de esta práctica de laboratorio es programar un teclado 4x4 para el PIC18F45K50 y mostrar el valor de la tecla correspondiente en el display de 7 segmentos. Para esto se deberá implementar una rutina de barrido para adquirir 16 valores de datos de un teclado 4x4 utilizando 8 pines de entrada/salida. Así como conocer la función y las interacciones entre la resistencia de pull-up y los puertos de entrada/salida de un dispositivo.

## Desarrollo

Para el desarrollo de esta práctica se siguieron los pasos descritos en el archivo de la práctica. El código que se utilizó en la práctica es el siguiente:

```
1  /*
2   * File:   main.c
3   * Author: gabii
4   *
5   * Created on 1 de mayo de 2022, 09:44 AM
6   */
7
8
9  ///+-----| LIBRARIES / HEADERS |-----+
10 #include "device_config.h.txt"
11 #include <stdint.h>
12 #include <math.h>
13 #include <xc.h>
14
15 //+-----| DIRECTIVES |-----+
16 #define _XTAL_FREQ 8000000
17 #define SWEEP_FREQ 20
18 #define SWEEP_STEP 5000
19
20 //+-----| DATA TYPES |-----+
21 enum por_ACDC {digital, analog};          // digital = 0, analog = 1
22
23
24 //+-----| FUNCTION DECLARATIONS |-----+
25 void portsInit(void);
26 uint8_t char_to_seg(uint8_t);
27 void send_to_disp(uint32_t);
28 char key_scanner(void);
```

```

30 //+++++| MAIN |+++++
31 void main(void){
32     int num_to_disp = 0;
33     portsInit();
34     while(1){
35         char key = key_scanner();
36         send_to_disp(key);
37     }
38     while(1){
39         uint32_t num = 0x01020304;
40         send_to_disp(num);
41     }
42 }
43
44 //+++++| FUNCTIONS |+++++
45 void portsInit(void){
46     ANSELA = digital; // Set port A as Digital for keypad driving
47     TRISA = 0x0F; // For Port A, set pins 4 to 7 as inputs (columns), and pins 0 to 3 as outputs (rows)
48     ANSELB = digital; // Set port B as Digital for 7 segment cathode selection (only 4 pins used)
49     TRISB = 0x00; // For Port B, set pins as outputs for cathode selection
50     ANSEL D = digital; // Set port D as Digital for 7 segment anodes
51     TRISD = 0x00; // for Port D, set all pins as outputs for 7 segment anodes
52     OSCCON = 0x74; // Set the internal oscillator to 8MHz and stable
53 }
54

```

```

57 char key_scanner(void){
58     LATAbits.LA0 = 0;
59     LATAbits.LA1 = 1;
60     LATAbits.LA2 = 1;
61     LATAbits.LA3 = 1;
62     __delay_ms(SWEEP_FREQ);
63     if (PORTAbits.RA4 == 0) {__delay_ms(SWEEP_FREQ); return 1;}
64     else if (PORTAbits.RA5 == 0) {__delay_ms(SWEEP_FREQ); return 2;}
65     else if (PORTAbits.RA6 == 0) {__delay_ms(SWEEP_FREQ); return 3;}
66     else if (PORTAbits.RA7 == 0) {__delay_ms(SWEEP_FREQ); return 10;}
67     LATAbits.LA0 = 1;
68     LATAbits.LA1 = 0;
69     LATAbits.LA2 = 1;
70     LATAbits.LA3 = 1;
71     __delay_ms(SWEEP_FREQ);
72     if (PORTAbits.RA4 == 0) {__delay_ms(SWEEP_FREQ); return 4;}
73     else if (PORTAbits.RA5 == 0) {__delay_ms(SWEEP_FREQ); return 5;}
74     else if (PORTAbits.RA6 == 0) {__delay_ms(SWEEP_FREQ); return 6;}
75     else if (PORTAbits.RA7 == 0) {__delay_ms(SWEEP_FREQ); return 11;}
76     LATAbits.LA0 = 1;
77     LATAbits.LA1 = 1;
78     LATAbits.LA2 = 0;
79     LATAbits.LA3 = 1;
80     __delay_ms(SWEEP_FREQ);
81     if (PORTAbits.RA4 == 0) {__delay_ms(SWEEP_FREQ); return 7;}
82     else if (PORTAbits.RA5 == 0) {__delay_ms(SWEEP_FREQ); return 8;}
83     else if (PORTAbits.RA6 == 0) {__delay_ms(SWEEP_FREQ); return 9;}
84     else if (PORTAbits.RA7 == 0) {__delay_ms(SWEEP_FREQ); return 12;}
85     LATAbits.LA0 = 1;

```

```

86     LATAbits.LA1 = 1;
87     LATAbits.LA2 = 1;
88     LATAbits.LA3 = 0;
89     __delay_ms(SWEEP_FREQ);
90     if (PORTAbits.RA4 == 0) { __delay_ms(SWEEP_FREQ); return 14;}
91     else if (PORTAbits.RA5 == 0) { __delay_ms(SWEEP_FREQ); return 0;}
92     else if (PORTAbits.RA6 == 0) { __delay_ms(SWEEP_FREQ); return 15;}
93     else if (PORTAbits.RA7 == 0) { __delay_ms(SWEEP_FREQ); return 13;}
94     else return 'x';
95 }
96
97 void send_to_disp(uint32_t disp_word){
98     for (char i = 0; i < 4; i++){
99         int internal_sweep = (int) pow(2, i);
100         int sweep = 0x0F & ~internal_sweep;
101         LATB = (char) sweep;
102         uint8_t num_disp = 0x000000FF & (disp_word >> i*8);
103         LATD = char_to_seg(num_disp);
104         __delay_ms(SWEEP_FREQ);
105     }
106 }

```

```

108 uint8_t char_to_seg(uint8_t num){
109     uint8_t segments;
110     switch(num) {
111         case 0: segments = 0b00111111; break;
112         case 1: segments = 0b00000110; break;
113         case 2: segments = 0b01011011; break;
114         case 3: segments = 0b01001111; break;
115         case 4: segments = 0b01100110; break;
116         case 5: segments = 0b01101101; break;
117         case 6: segments = 0b01111101; break;
118         case 7: segments = 0b00000111; break;
119         case 8: segments = 0b01111111; break;
120         case 9: segments = 0b01100111; break;
121         case 10: segments = 0b01110111; break;
122         case 11: segments = 0b01111100; break;
123         case 12: segments = 0b01011000; break;
124         case 13: segments = 0b01011110; break;
125         case 14: segments = 0b01111001; break;
126         default: segments = 0b01110001; break;
127     }
128     return segments;
129 }

```

## **Conclusiones**

Considero que es importante aprender sobre las diferentes aplicaciones que se pueden realizar en un microcontrolador, también qué es importante aprender sobre estos temas debido a que en un futuro nos podrían ser de mucha utilidad.