

UNIVERSIDAD CARLOS III DE MADRID

MEMORIA DEL PROYECTO

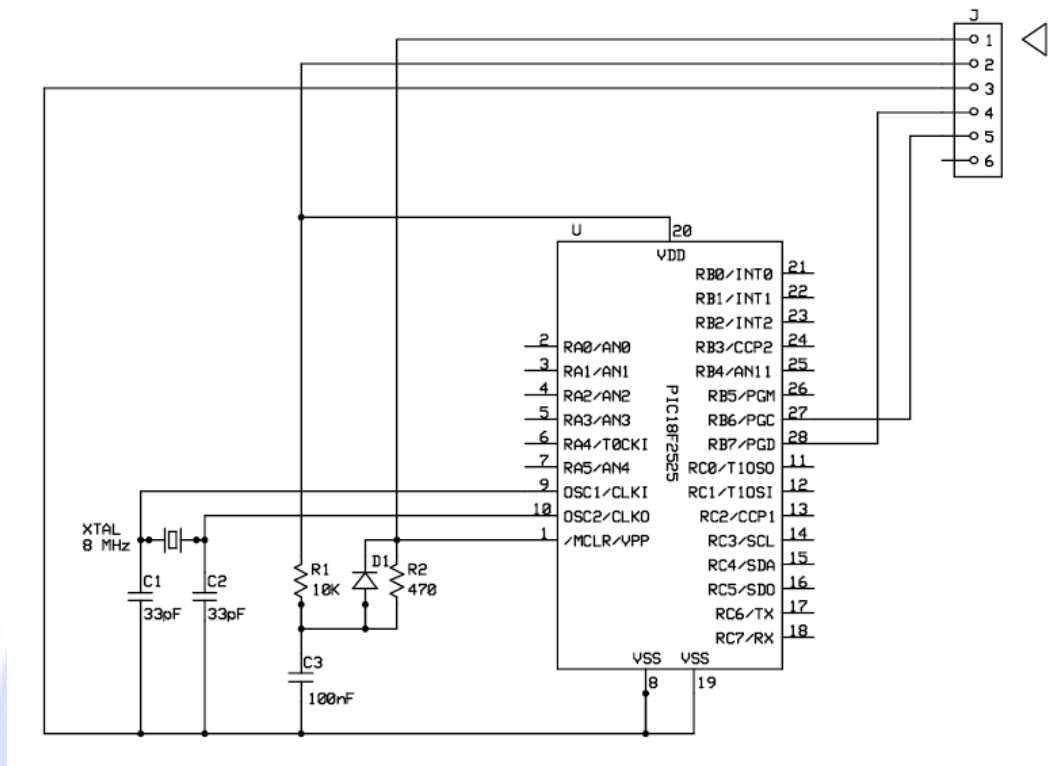
SISTEMAS DIGITALES BASADOS EN
MICROPROCESADORES

Carlos Parra Marcelo - 100304942

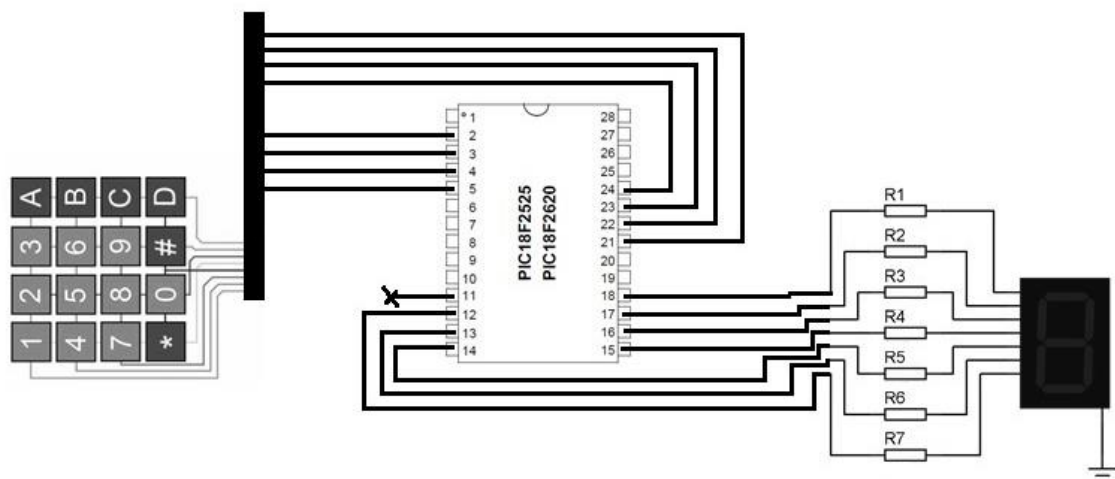
Grupo 9

1. Esquema del circuito:

Esquemático de la placa de desarrollo:



Esquemático del teclado y del display:



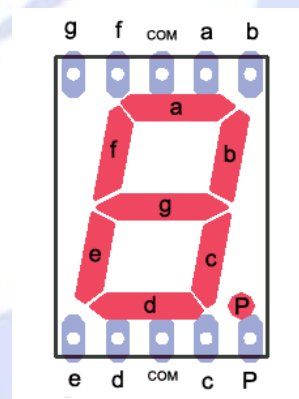
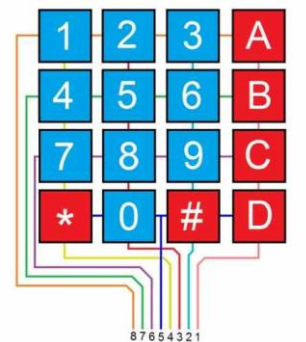
NOTA: R1, R2, R3, R4, R5, R6 y R7 valen todas 370Ω

2. Asignación de puertos:

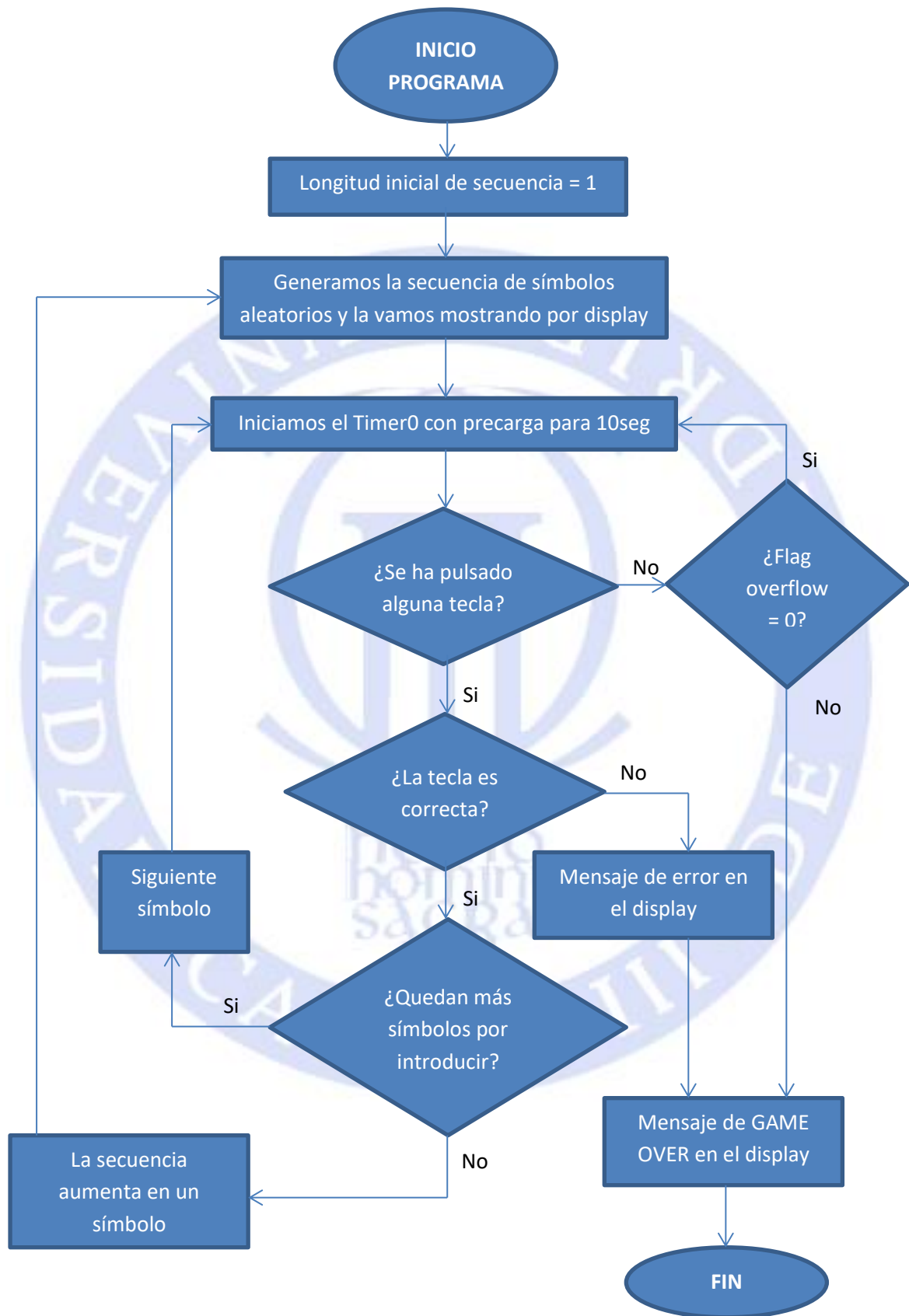
PUERTO A	
RA0	Columna 1 del teclado (Output)
RA1	Columna 2 del teclado (Output)
RA2	Columna 3 del teclado (Output)
RA3	Columna 4 del teclado (Output)
RA4	LIBRE
RA5	LIBRE
RA6	OCUPADO (CLK)
RA7	OCUPADO (CLK)

PUERTO B	
RB0	Fila 1 del teclado (Input)
RB1	Fila 2 del teclado (Input)
RB2	Fila 3 del teclado (Input)
RB3	Fila 4 del teclado (Input)
RB4	LIBRE
RB5	LIBRE
RB6	OCUPADO
RB7	OCUPADO

PUERTO C	
RC0	Segmento G del display (Output)
RC1	Segmento F del display (Output)
RC2	Segmento E del display (Output)
RC3	Segmento D del display (Output)
RC4	Segmento C del display (Output)
RC5	Segmento B del display (Output)
RC6	Segmento A del display (Output)
RC7	Punto decimal del display (Output)



3. Diagrama de flujo:



4. Código del programa con comentarios:

```
//;   Filename:
//;   Date:
//;   File Version:   C Program Template
//;   Author:         Carlos Parra Marcelo (Grupo 9)
//;   Company:
//;*****
//Include Files-----
-
#include <pl8f2525.h>
#include "delays.h"
#include "stdlib.h"           //Para poder usar "rand()"

//Config-----
-
#pragma config OSC    = HS      // - Habilitar oscilador HS
#pragma config WDT    = OFF     // - Deshabilitar watchdog timer
#pragma config DEBUG  = ON      // - Deshabilitar debugger

#define _XTAL_FREQ 80000000    //Frecuencia del cristal: 8MHz

//Constants-----
-
#define IOconfig 0b00001111    //Salida y entrada digital

#define resistencias INTCON2bits.RBPUP //Resistencias de Pull-Up del PORTB

#define entrada_fila_1 PORTBbits.RB0  //Filas en el PORTB
#define entrada_fila_2 PORTBbits.RB1
#define entrada_fila_3 PORTBbits.RB2
#define entrada_fila_4 PORTBbits.RB3
#define salida_columna_1 PORTAbits.RA0 //Columnas en el PORTA
#define salida_columna_2 PORTAbits.RA1
#define salida_columna_3 PORTAbits.RA2
#define salida_columna_4 PORTAbits.RA3

//Global Var-----
-
int tecla;           //Última tecla pulsada
int numero;         //Posición que el programa comprobará
int longitudSecuencia; //Longitud de la secuencia aleatoria
int secuencia[10];  //Array que contiene la secuencia
const char numDisplay[] = {0x7E, 0x30, 0x6D, 0x79, 0x33, 0x5B, 0x5F, 0x71,
                          0x7F, 0x7B, 0x77, 0x1F, 0x4E, 0x3D, 0x36, 0x37};
                          //Array de los números del display

//Function Prot-----
-
void Inicializa(void);
//void delay10ms(void);
void delay1s(void);           //Retardo de 1seg
void delay10s(void);         //Retardo de aprox. 8seg (no llega a
10)
int buscarFila(void);         //Comprueba las filas
int buscarColumna(void);      //Comprueba las columnas
void generarSecuencia(int);    //Genera una secuencia aleatoria con
//la longitud que queramos
```

```

//Main-----
-
void main(void){
    tecla = 17;                //17 significa: ninguna tecla pulsada
    numero = 0;                //Posición 0 del array
    Inicializa();
    LATC = 0x00;              //Borramos display
    longitudSecuencia = 1;     //Secuencia inicial de longitud = 1
    while (1) {
        generarSecuencia(longitudSecuencia);    //Generamos la secuencia
        while (numero<longitudSecuencia) {
            delay10s();                //Mientras transcurran los 8seg:
            while (!INTCONbits.TMR0IF) {
                for (long int i = 0; i>250000; i++) {
                }                    //Esperamos 0.25seg
                tecla = buscarColumna();    //Comprobamos teclado
                if (tecla>=0 && tecla<16) { //Si se ha pulsado alguna
tecla:
                    if (secuencia[numero]==tecla) { //Si es la correcta:
                        numero++;                //Siguiete posición del array
                        INTCONbits.TMR0IF = 1;    //Final de los 10seg
                    } else {                    //Si se pulsa otra tecla
                        tecla = 20;
                        numero = 20;
                        INTCONbits.TMR0IF = 1;    //Final de los 10seg
                    }
                }
            }
            //Terminan los 10seg de espera
            if (tecla==17 && INTCONbits.TMR0IF==1) { //Si no se ha
//pulsado nada:
                delay1s();                //Para "frenar" un poco la ejecución
                longitudSecuencia = 0;    //Así salimos de uno de los bucles
            } else if (tecla==20) { //Si la tecla pulsada es incorrecta:
                delay1s();
                longitudSecuencia = 0;
            } else if (tecla>=0 && tecla<16) { //Si la tecla pulsada es
la
//correcta:
                delay1s();
            }
        }
        if (numero==longitudSecuencia) { //Si hemos terminado de
//comprobar toda la secuencia:
            numero = 0;                //Vuelta al comienzo del array
            longitudSecuencia++;        //Incrementamos la longitud de la sec.
        }
        LATC = 0b00000001;            //Símbolo FINAL DE PROGRAMA
        delay10s();
        while (!INTCONbits.TMR0IF);
        LATC = 0x00;                //Borramos display
    }
}

//Setup-----
void Inicializa(void){
    ADCON1 = IOconfig; //Set port A and B as digital I/O pins
    TRISA = 0x00;        //Las columnas serán Output
    TRISB = 0x0F;        //Las filas serán Input
    TRISC = 0x00;        //El display será Output
}

```



```

        resistencias = 0;          //Resistencias de Pull-Up activadas
    }

void delay1s(void) {
    TMR0H = 0xFF;                  //Precarga
    TMR0L = 0x0B;
    INTCONbits.TMR0IF = 0;        //Borrado flag overflow
    T0CON = 0x84;
    while (!INTCONbits.TMR0IF);    //Detección flag overflow
}

void delay10s(void) {
    TMR0H = 0xFF;                  //Precarga
    TMR0L = 0x00;
    INTCONbits.TMR0IF = 0;        //Borrado flag overflow
    T0CON = 0x87;
    //En esta función no esperamos a que transcurran los (aprox.) 10seg
}

void generarSecuencia(int longitud) {
    //Generamos una secuencia aleatoria de "longitud" símbolos
    for (int i = 0; i<longitud; i++) {
        secuencia[i] = rand() % 15; //La guardamos en el array secuencia[]
        LATC = numDisplay[secuencia[i]]; //Mostramos el símbolo
        delay1s();                      //Durante un segundo
        LATC = 0x00;                    //Borramos display
    }
}

int buscarFila(void) {
    int numFila = 0;
    if (!(entrada_fila_1)) {
        for (long int i = 0; i<40; i++) {
        }
        if (!(entrada_fila_1)) numFila = 1;
    } else if (!(entrada_fila_2)) {
        for (long int i = 0; i<40; i++) {
        }
        if (!(entrada_fila_2)) numFila = 2;
    } else if (!(entrada_fila_3)) {
        for (long int i = 0; i<40; i++) {
        }
        if (!(entrada_fila_3)) numFila = 3;
    } else if (!(entrada_fila_4)) {
        for (long int i = 0; i<40; i++) {
        }
        if (!(entrada_fila_4)) numFila = 4;
    }
    return numFila;
    //Esta función devuelve el número de la fila pulsada; en otro caso
    //devuelve 0
}

int buscarColumna(void) {
    int numFila = 0;
    int numFinal = 17;              //Valor predeterminado cuando no se pulsa nada
    PORTA = 0x0F;                   //Ponemos las 4 columnas a "1"
    salida_columna_1 = 0;           //Chequeamos la 1ª columna poniéndola a "0"
}

```

```

numFila = buscarFila();          //Si se ha pulsado algo "numFinal" guarda el
                                //número de la tecla
if (numFila>0 && numFila<5) {
    if (numFila==1) numFinal = 1;
    else if (numFila==2) numFinal = 4;
    else if (numFila==3) numFinal = 7;
    else if (numFila==4) numFinal = 15;
}
PORTA = 0x0F;                    //Volvemos a poner las 4 columnas a "1"

if (numFinal==17) {
    salida_columna_2 = 0;        //Chequeamos la 2ª columna poniéndola a "0"
    numFila = buscarFila();
    if (numFila>0 && numFila<5) {
        if (numFila==1) numFinal = 2;
        else if (numFila==2) numFinal = 5;
        else if (numFila==3) numFinal = 8;
        else if (numFila==4) numFinal = 0;
    }
}
PORTA = 0x0F;

if (numFinal==17) {
    salida_columna_3 = 0;
    numFila = buscarFila();
    if (numFila>0 && numFila<5) {
        if (numFila==1) numFinal = 3;
        else if (numFila==2) numFinal = 6;
        else if (numFila==3) numFinal = 9;
        else if (numFila==4) numFinal = 14;
    }
}
PORTA = 0x0F;

if (numFinal==17) {
    salida_columna_4 = 0;
    numFila = buscarFila();
    if (numFila>0 && numFila<5) {
        if (numFila==1) numFinal = 10;
        else if (numFila==2) numFinal = 11;
        else if (numFila==3) numFinal = 12;
        else if (numFila==4) numFinal = 13;
    }
}

return numFinal;                //La función devuelve el número de la tecla
                                //pulsada; en otro caso devuelve "17"
}

```


5. Comentarios sobre el proyecto:

Funcionamiento del programa:

En la 4ª y la 5ª (extra) práctica me he topado con el mismo problema al probar el programa depurándolo paso a paso: en la función “generarSecuencia(int)” debe de haber algún tipo de problema ya que a partir de la segunda ejecución de la función los retardos (ya sean un “for” o el Timer0) dejan de funcionar y se muestran por display cantidad de símbolos a una velocidad tan rápida que da la sensación de parpadeo.

En la última práctica extra hice pruebas con el coordinador (Guillermo Carpintero) y no sabíamos de dónde surgía el error, además de tener problemas con el depurador de MPLAB. Por eso no he podido progresar más en el proyecto.

Retardos:

Solo he utilizado un timer (Timer0) al programar ya que no he visto necesario utilizar ninguno más. Posiblemente se podría utilizar el mismo u otro diferente para generar la semilla de la función “srand()” y así conseguir una secuencia completamente aleatoria al ejecutar el programa.

A su vez he implementado varios “for” cuando la función del temporizador de 10 segundos está activa ya que no podía volver a utilizar el Timer0 en ese período de tiempo (eso generaría problemas con el flag de overflow). Para calcular cuántas veces tenía que repetirse el bucle consideraba que el “for” tarda en ejecutarse 2 ciclos de trabajo (0.001ms), por lo que si quería implementar un retardo de 0.25seg simplemente el bucle tendría que repetirse 250000 veces.

Observación sobre el teclado:

Como curiosidad, dado a cómo implementé el código para la lectura del teclado (el programa comprueba en cada columna el estado de cada fila), si mantenemos pulsada una tecla y a la vez probamos a pulsar simultáneamente pueden ocurrir dos cosas:

- Si las dos teclas pertenecen a la misma columna en el display se mostrará la tecla perteneciente al número de fila más alto.
- Si las dos teclas no pertenecen a la misma columna en el display podremos observar el símbolo perteneciente a la tecla de la columna mayor (Ej: col. 2 < col. 4).

Nota sobre mi compañero de prácticas:

Mi compañero Jorge Sánchez Alberto abandonó la asignatura prácticamente al comienzo de los laboratorios y me he encargado yo exclusivamente de todo el trabajo del proyecto. Él ha accedido a que hiciese éste comentario para que constase que aunque ha acudido a todas las prácticas menos a la extra realmente el progreso en el proyecto lo he llevado a cabo solo yo.

