UNIVERSIDAD AUTONOMA DEL ESTADO DE MEXICO
UNIDAD ACEDEMICA PROFESIONAL TIANGUISTENCO

PROGRAMACION DE MICROCONTROLADORES
PRCTICA No. 03

DR. JOSE ARTURO PEREZ MARTINEZ

SAUL RUIZ PIÑA
JOSE ALFREDO DOMINGUEZ ARISTA
EMMANUEL MARTINEZ DIAZ

OBJETIVO: Diseñar y elaborar una cerradura electrónica utilizando un PIC18f4550. Esta cerradura electrónica debía tener las siguientes funcionalidades:

DESARROLLO

PLANTEAMIENTO DEL PROBLEMA

1. Ingreso de una contraseña: El usuario será capaz de introducir una contraseña utilizando un teclado matricial 4x4. Esta contraseña se compararía con una contraseña predefinida para determinar si es correcta.
2. Verificación de contraseña: Una vez ingresada la contraseña, el sistema debería verificar si coincide con la contraseña predefinida. Si la contraseña es correcta, la cerradura electrónica debería abrirse, permitiendo el acceso.
3. Cambio de contraseña: El sistema debe proporcionar la opción de cambiar la contraseña predefinida. Esto permitiría al usuario actualizar la contraseña según sea necesario para mantener la seguridad del sistema.
4. Control de acceso: Si la contraseña ingresada no es correcta, la cerradura electrónica debería permanecer cerrada, negando el acceso al usuario.

MATERIAL UTILIZADO

- Microcontrolador PIC18F4550.
- Pantalla lcd 16X2.
- Potenciometro 50K.
- Teclado Matricial 4x4.
- Placa de prueba.
- Resistencias 330Ω, 1kΩ, 100kΩ.

SOLUCION

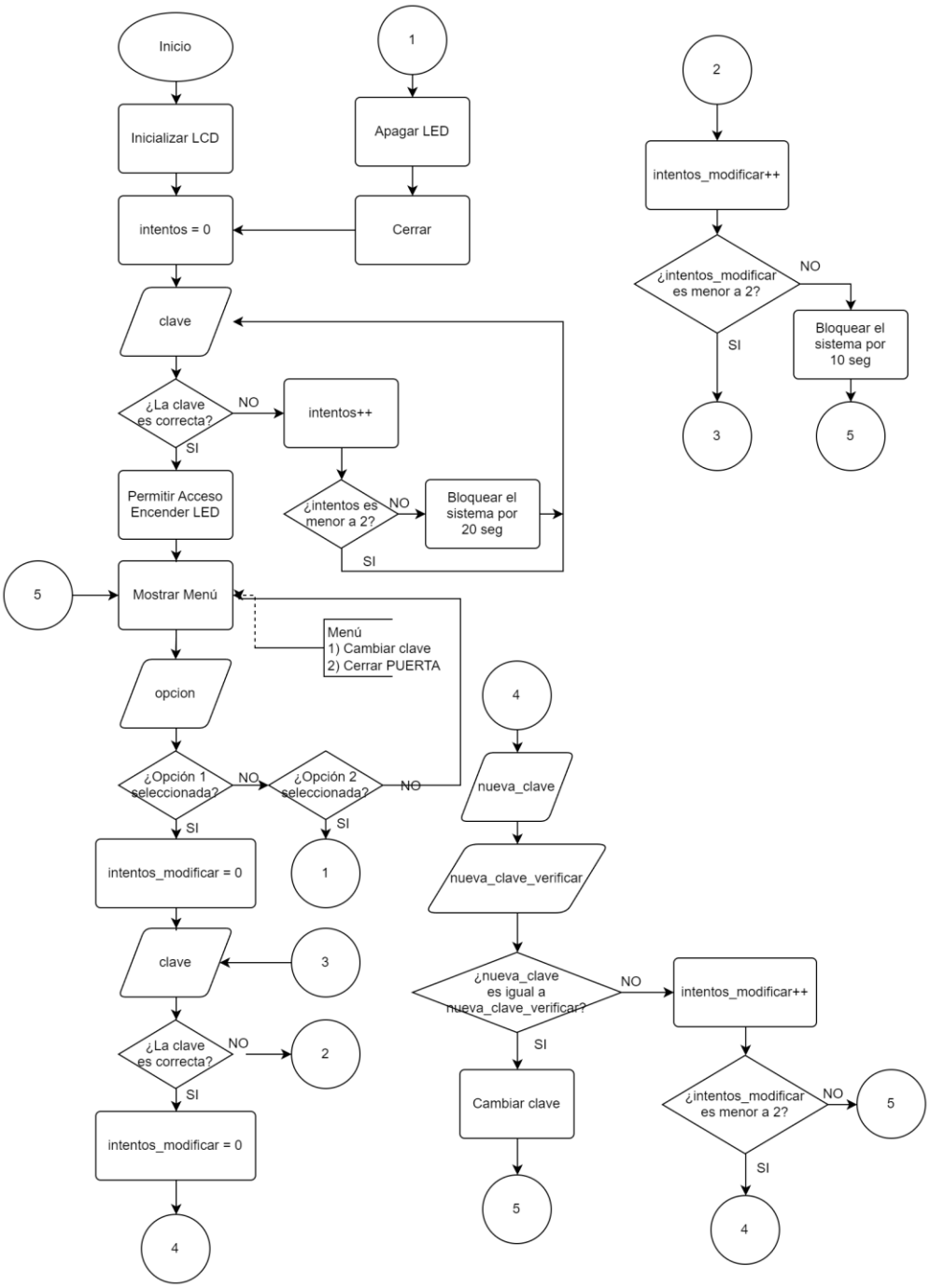
1.- Diseño de la Lógica de Acceso y Contraseña: Se definió una contraseña predefinida en el código del programa. Se implementó una función para comparar la contraseña ingresada por el usuario con la contraseña predefinida. Si la contraseña ingresada coincide con la predefinida, se activa la señal para abrir la cerradura electrónica; de lo contrario, se mantiene cerrada.

2.- Configuración del Teclado Matricial: Se configuraron los pines del microcontrolador para recibir las señales del teclado matricial. Se desarrolló un procedimiento para leer las teclas presionadas por el usuario y convertirlas en una cadena de caracteres que representan la contraseña ingresada.
Implementación de la Funcionalidad de Cambio de Contraseña: Se diseñó una función que permite al usuario cambiar la contraseña predefinida. Esta función verifica la contraseña actual antes de permitir el ingreso de una nueva contraseña.

3.- Implementación de la Funcionalidad de Cambio de Contraseña: Se diseñó una función que permite al usuario cambiar la contraseña predefinida. Esta función verifica la contraseña actual antes de permitir el ingreso de una nueva contraseña.

4.- Integración de Componentes y Algoritmos: Se integraron todos los componentes y algoritmos en un único programa. Se desarrolló un bucle principal que espera la entrada de la contraseña y la procesa según las funcionalidades requeridas: verificación de acceso, cambio de contraseña y control de la cerradura electrónica.

5.- Pruebas y Depuración: Se realizaron pruebas exhaustivas del sistema para verificar su correcto funcionamiento en diferentes escenarios, incluyendo casos de contraseña correcta e incorrecta, así como el proceso de cambio de contraseña. Se identificaron y corrigieron posibles errores y fallos de funcionamiento.




```
#include <18f4550.h>
#include <string.h>
#fuses xt, nowdt, intrc
#use delay(internal = 8M)
#define LCD_DATA_PORT getenv("SFR:PORTD")
#include <lcd.c>

#byte TRISB = 0xF93
#byte PORTB = 0xF81
#byte TRISC = 0xF94
#byte PORTC = 0xF82

#define VERDE        0
#define AMARILLA     1
#define ROJA         2
#define APAGADO      0
#define ENCENDIDO    1
#define MAX_LONGITUD 5

char *mensages[] = {
        "CLAVE\n   INCORRECTA",
        "ABRIENDO\n      PUERTA",
        "CERRANDO\n      PUERTA",
        "NUEVA CLAVE",
        "CONFIRMAR CLAVE",
        "CLAVE\n   ACTUALIZADA",
        "PUERTA CERRADA",
        " CLAVE ACTUAL",
        "LAS CLAVES\n   NO COINCIDEN"
    };

void cargando();
void mensage_info(char cadena[], int pos_x,
char com_tecla();
int comando_tecla();
void ingresar_clave(char clave[], char cade
void esperar(int tiempo);
void acceso_permitido(char cadena[]);
void inicializar(char cadena[], int longitu
```

```
void obtener_nueva_clave(char clave[], char mensaje[]);
void imprimir_cadena(char cadena[]);

char clave[MAX_LONGITUD] = {'1','2','3','4','\0'};

void main()
{
   int intentos_ingresar = 0;
   char clave_ingresada[MAX_LONGITUD];
   TRISC = 0x00;
   PORTC = 0x00;
   inicializar(clave_ingresada, MAX_LONGITUD, '\0');

   cargando();

   while(TRUE)
   {
      ingresar_clave(clave_ingresada, mensages[6]);
      if(!strcmp(clave, clave_ingresada))
      {
         lcd_putc("\f");
         mensage_info(mensages[1], 6, 1, AMARILLA, APAG
         bit_set(PORTC, 0);
         acceso_permitido(mensages[2]);
         intentos_ingresar = 0;
      }else
      {
         mensage_info(mensages[0], 7, 1, ROJA, APAGADO)
         if(++intentos_ingresar == 2)
         {
            esperar(20);
            intentos_ingresar = 0;
         }
      }
      lcd_putc("\f");
   }
}
```

```
void acceso_permitido(char cadena[])
{
   char tecla = 0;
   lcd_putc("PUERTA 1)CAMBIAR\n");
   lcd_putc("ABIERTA 2)CERRAR");
   do
   {
      tecla = com_tecla();
      switch(tecla)
      {
         case '1':
            cambiar_clave();
            lcd_putc("PUERTA 1)CAMBIAR\n");
            lcd_putc("ABIERTA 2)CERRAR");
            break;
         case '2':
            lcd_putc("\f");
            bit_clear(PORTC, 0);
            mensage_info(cadena, 6, 1, AMARILLA, APAGADO);
            break;
      }
   }while(tecla != '2');
}

int cambiar_clave()
{
   char clave_ingresada[MAX_LONGITUD];

   inicializar(clave_ingresada, MAX_LONGITUD, '\0');
   lcd_putc("\f");
   for(int intentos_autenticar = 0; intentos_autenticar < 2; intentos_autent
   {
      ingresar_clave(clave_ingresada, mensages[7]);

      if(!strcmp(clave, clave_ingresada))
      {
         char clave_verificar[MAX_LONGITUD];
         for(intentos_autenticar = 0; intentos_autenticar < 2; intentos_aute
```

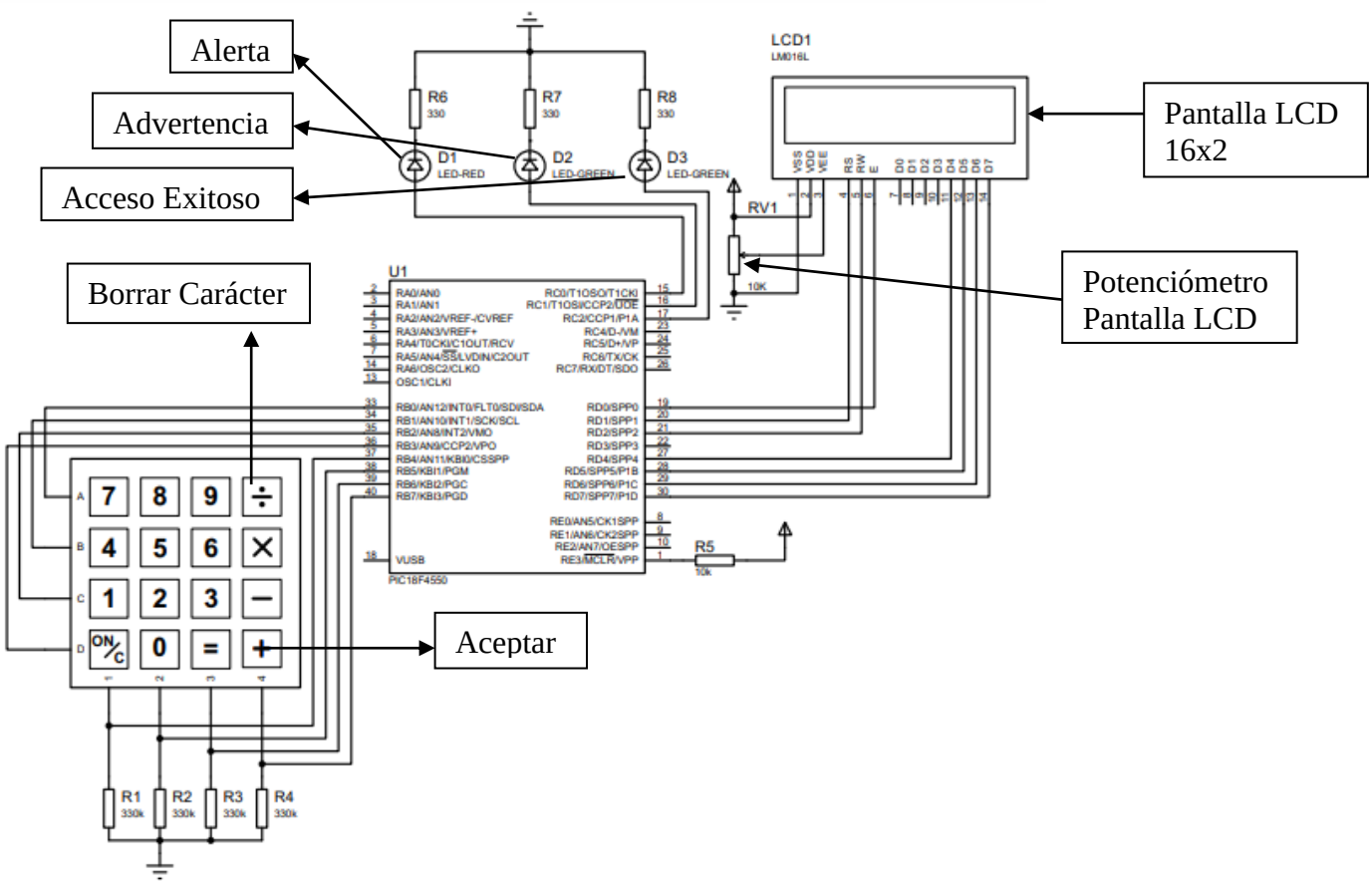
```
      {
         clave[longitud--] = '\0';
         lcd_putc("\b \b");
      }
   }while(tecla != 'D' || longitud != MAX_LONGITUD - 1);
   clave[MAX_LONGITUD - 1] = '\0';
   lcd_putc("\f");
}
void esperar(int tiempo)
{
   lcd_putc("\f");
   bit_set(PORTC, 1);
   lcd_gotoxy(3,1);
   lcd_putc("INTENTE EN :");
   lcd_gotoxy(8,2);
   for(int i = tiempo; i > 0; i--)
   {
      lcd_putc((char)(48 + (i/10)));
      lcd_putc((char)(48 + (i%10)));
      delay_ms(250);
      lcd_putc("\b \b\b \b");
   }
   bit_clear(PORTC, 1);
   lcd_putc("\f");
}
void ingresar_clave(char clave[], char cadena[])
{
   lcd_gotoxy(2,1);
   printf(lcd_putc,"%s", cadena);
   lcd_gotoxy(3,2);
   lcd_putc("CLAVE : ");
   int longitud = 0;
   char tecla = 0;
   do
   {
      tecla = comando_tecla();
      if(longitud < MAX_LONGITUD - 1)
      {
         if(tecla >= '0' && tecla <= '9')
```

```
      {
         inicializar(clave_ingresada, MAX_LONGITUD, '\0');
         inicializar(clave_verificar, MAX_LONGITUD, '\0');
         obtener_nueva_clave(clave_ingresada, mensages[3]);
         obtener_nueva_clave(clave_verificar,mensages[4]);
         if(!strcmp(clave_ingresada, clave_verificar))
         {
            strcpy(clave, clave_verificar);
            mensage_info(mensages[5], 7, 1, VERDE, ENCENDIDO);
            return 0;
         }else
         {
            mensage_info(mensages[8], 7, 1, ROJA, APAGADO);
         }
      }
   }else
   {
      mensage_info(mensages[0], 7, 1, ROJA, APAGADO);
   }

sperar(10);
eturn 1;

 obtener_nueva_clave(char clave[], char mensaje[])

nt longitud = 0;
har tecla = 0;
cd_gotoxy(2,1);
rintf(lcd_putc,"%s", mensaje);

cd_gotoxy(6,2);
o

   tecla = comando_tecla();
   if(longitud < MAX_LONGITUD - 1)
   {
      if(tecla >= '0' && tecla <= '9')
```

```
         {
            lcd_putc('*');
            clave[longitud++] = tecla;
            clave[longitud] = '\0';
         }
      }
      if(tecla == 'A' && longitud > 0)
      {
         clave[longitud--] = '\0';
         lcd_putc("\b \b");
      }
   }while(tecla != 'D' || longitud == 0);
   lcd_putc("\f");
}
void mensage_info(char cadena[], int pos_x, int pos_y, int luz, int esta
{
   for(int j = 1; j < 4; j++)
   {
      lcd_gotoxy(pos_x, pos_y);
      printf(lcd_putc,"%s", cadena);
      if(luz == 0 || luz == 1 || luz == 2)
      {
         bit_set(PORTC, luz);
         delay_ms(95);
         lcd_putc("\f");
         bit_clear(PORTC, luz);
         delay_ms(45);
      }else
      {
         lcd_putc("\f");
      }
   }
   if(luz == 0 || luz == 1 || luz == 2)
   {
      if(establecer)
      {
         bit_set(PORTC, luz);
      }
   }
```

```
void inicializar(char cadena[], int longitud, char valor)
{
   for(int a = 0; a < longitud; a++)
   {
      cadena[a] = valor;
   }
}
void cargando()
{
   lcd_init();
   lcd_gotoxy(5,1);
   lcd_putc("INICIANDO");
   for(int i = 1; i < 17; )
   {
      for(int j = 0; j < 3; j++)
      {
         lcd_gotoxy(i++,2);
         lcd_putc((unsigned char)255);
         bit_set(PORTC, 2 - j);
         delay_ms(30);
         bit_clear(PORTC, 2- j);
      }
   }
   lcd_putc("\f");
}
char com_tecla()
{
   const int FILA = 4;
   const int COLUMNA = 4;
   char const KEYS[4][4] = {
                              {'1','2','3','A'},
                              {'4','5','6','B'},
                              {'7','8','9','C'},
                              {'*','0','#','D'}
                            };
   char tecla = 0;

   TRISB=0xF0;
   PORTB=0x00;
```

```
   for(int i = 0; i < FILA; i++)
   {
      bit_set(PORTB, i);
      for(int j = 0; j < COLUMNA; j++)
      {
         if(bit_test(portb,j + COLUMNA) == 1)
         {
            tecla=KEYS[i][j];
            delay_ms(45);
            return tecla;
         }
      }
      bit_clear(PORTB, i);
   }
   return tecla;
}
int comando_tecla()
{
   char tecla = 0;
   lcd_putc((unsigned char)255);
   for(int i = 0; i < 6; i++)
   {
      delay_ms(10);
      tecla = com_tecla();
      if(tecla)
      {
         lcd_putc("\b \b");
         return tecla;
      }
   }
   lcd_putc("\b \b");
   for(int j = 0; j < 6; j++)
   {
      delay_ms(10);
      tecla = com_tecla();
      if(tecla)
      {
         return tecla;
      }
```



CONCLUSIONES

La práctica de diseño y desarrollo de una cerradura electrónica utilizando el microcontrolador PIC 18F4550 representó una experiencia enriquecedora en la aplicación y consolidación de diversos conocimientos en el campo de la programación de microcontroladores, la electrónica digital y la seguridad informática. A lo largo del proceso, se pudo observar la complejidad y los desafíos inherentes a la implementación de sistemas embebidos que requieren un alto grado de precisión y confiabilidad.

Uno de los aspectos más destacados de esta práctica fue la comprensión de la importancia de un diseño modular y bien estructurado. La división del problema en componentes más pequeños, como la lógica de acceso, la gestión de contraseñas y la interfaz de usuario, facilitó la implementación y depuración del sistema en su conjunto. Esta metodología se mantuvo durante todo el desarrollo, lo que permitió mantener un código limpio, organizado y fácilmente mantenible.

El diseño e implementación del algoritmo de verificación de contraseñas representó uno de los puntos críticos de la práctica. La necesidad de garantizar la seguridad y la confiabilidad en la autenticación del usuario requirió un enfoque cuidadoso y meticuloso. Se empleó una comparación de contraseñas segura y eficiente para evitar posibles vulnerabilidades, como la exposición de la contraseña predefinida en el código fuente.

La integración de hardware y software fue fundamental para el éxito del proyecto. La correcta configuración de los pines del microcontrolador, la comunicación con el teclado matricial y el control de la cerradura electrónica demandaron un entendimiento profundo de la arquitectura y las capacidades del PIC 18F4550. La habilidad para resolver problemas de compatibilidad y asegurar la interoperabilidad entre los diferentes componentes se mantuvo a lo largo de todo el desarrollo.

Además, se implementó una funcionalidad para el cambio de contraseña, lo que agregó un nivel adicional de versatilidad y seguridad al sistema.

Esto permitió una mayor personalización y adaptabilidad a las necesidades específicas del usuario, manteniendo la integridad del sistema a lo largo del tiempo.

En resumen, esta práctica representó una oportunidad invaluable para aprender sobre el diseño y desarrollo de sistemas embebidos seguros y eficientes. La metodología modular, la atención a la seguridad y la integración efectiva de hardware y software fueron aspectos clave que se mantuvieron a lo largo de todo el proceso de desarrollo de la cerradura electrónica.