

Laboratorio 3: Uso del Teclado

1. Datos de la Practica

Carrera	INGENIERÍA ELECTRÓNICA		
Semestre		Grupo	
Tipo de Práctica	<input type="checkbox"/> Laboratorio <input type="checkbox"/> Simulación	Fecha	
Asignatura	Electrónica Digital I		
Unidad Temática			
Nº Alumnos por práctica	2	Nº Alumnos por reporte	2
Nombre del Profesor			
Nombre(s) de Alumno(s)	1. 2.		
Tiempo estimado		Vo. Bo. Profesor	
Comentarios			

2. Objetivos

- Implementar un medio de ingreso de datos mediante un teclado matricial 4x4 y su conexión con el microcontrolador de la tarjeta Arduino MEGA.

3. Componentes a Utilizar

Por cada práctica y por cada puesto de laboratorio, los materiales a utilizar son:

Cantidad	Descripción
1	Computadora
1	Arduino MEGA
1	LEDs
1	Resistencia de 330 ohm
1	Potenciómetro de 10K
1	LCD 16X2

4. Introducción

Otro de los periféricos especiales que se utilizan en los proyectos con microcontroladores es el teclado matricial. Vienen generalmente en dos presentaciones: 3x4(**Fig. 1**) y 4x4(**Fig. 2**). Donde el primer número indica las cantidades de columnas y el segundo las cantidades de filas.

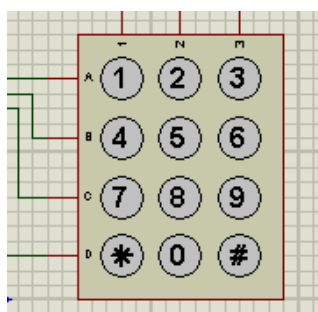


Fig. 2 Teclado 3x4

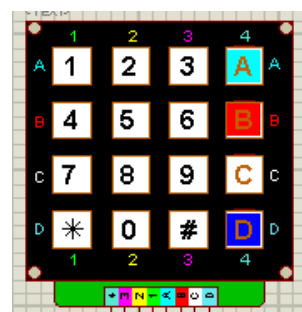


Fig. 1 Teclado 4x4

Programación del Arduino

La librería Keypad.h de Arduino no viene por defecto en el IDE, por lo que se tendrá que descargar de la siguiente dirección:

<https://github.com/tutankamon/arduino-labscodelabs/raw/master/lab3/Keypad-3.1.1.zip>

Esta librería presenta varias funciones de las que se detallan a continuación:

void begin(makeKeymap(userKeymap))

Descripción: Inicializa el mapeo interno de las teclas con las del usuario.

char waitForKey()

Descripción: Espera hasta que alguien presione una Tecla. Esto bloquea otro código hasta que se presione una tecla.

char getKey()

Descripción: Retorna la tecla que esta presionada.

KeyState getState()

Descripción: Retorna el estado actual de cualquiera de las teclas. Los cuatro estados son: IDLE, PRESSED, RELEASED y HOLD.

boolean keyStateChanged()

Descripción: Permite saber cuándo una Tecla ha cambiado de un estado a otro.

setHoldTime(unsigned int time)

Descripción: Configura una cantidad de milisegundos que el usuario tendrá que mantener la tecla hasta que el estado sea detectado.

setDebounceTime(unsigned int time)

Descripción: Configura una cantidad de milisegundos hasta que se acepte un evento de la tecla.

addEventListener(keypadEvent)

Descripción: Dispara un evento si el teclado es usado.

5. Prácticas de Laboratorio

Practica #1 Cerradura Electrónica

Código Arduino

El Circuito a Simular corresponde a una cerradura digital donde se utiliza un teclado matricial para el ingreso de la contraseña y una pantalla para la visualización de los datos. Como sistema de visualización se usará un led para la apertura de la puerta. El código es el siguiente:

```
#include <LiquidCrystal.h>
#include <Keypad.h>
#define Password_Length 8

int signalPin = 39;
char Data[Password_Length], customKey, Master[Password_Length] = "123A456";
byte data_count = 0, master_count = 0;
bool Pass_is_good;
const int rs = 6, en = 7, d4 = 8, d5 = 9, d6 = 10, d7 = 11;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
const byte ROWS = 4, COLS = 4;

char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'};
byte rowPins[ROWS] = {25, 24, 23, 22}; //(A,B,C,D)connect to the row pinouts of the keypad
byte colPins[COLS] = {29, 28, 27, 26}; //(1,2,3,4)connect to the column pinouts of the keypad

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

void setup() {
  lcd.begin(16,2);
  pinMode(signalPin, OUTPUT);}

void loop() {
  lcd.setCursor(0, 0);
  lcd.print("Enter Password:");
  customKey = customKeypad.getKey();
  if (customKey) {
    Data[data_count] = customKey;
    lcd.setCursor(data_count, 1);
    lcd.print(Data[data_count]);
    data_count++;}
```

```

if (data_count == Password_Length - 1) {
    lcd.clear();
    if (!strcmp(Data, Master)) {
        lcd.print("Correct");
        digitalWrite(signalPin, HIGH);
        delay(5000);
        digitalWrite(signalPin, LOW);}
    else {
        lcd.print("Incorrect");
        delay(1000);}
    lcd.clear();
    clearData();
}

void clearData() {
    while (data_count != 0) {
        Data[data_count--] = 0;
    }
    return;}

```

Simulación en Proteus

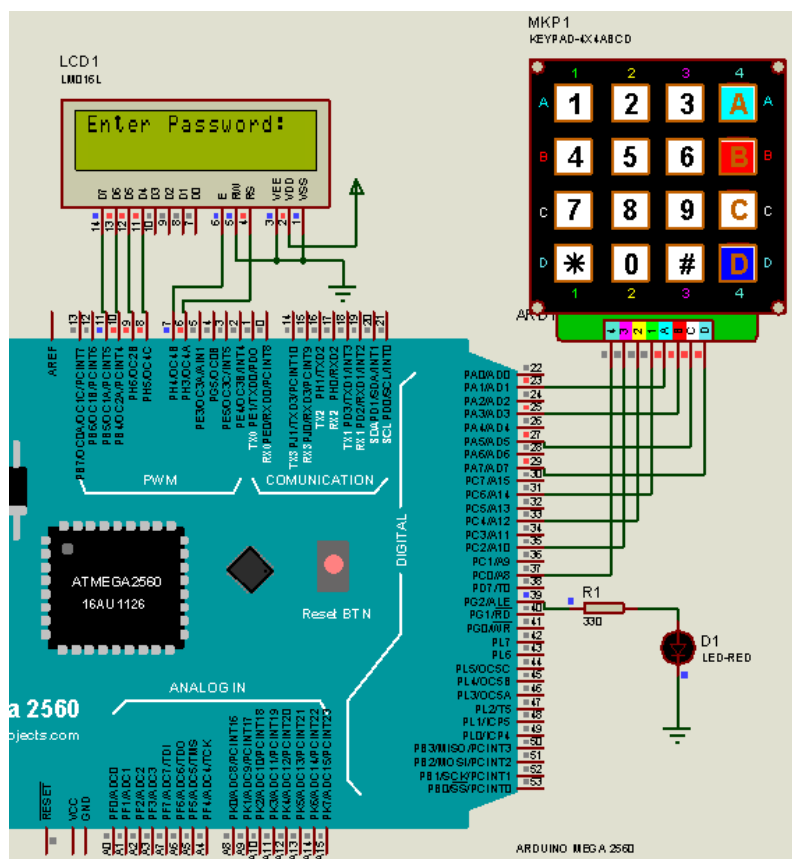


Fig. 3 Simulación de Cerradura Electrónica

Montaje del Circuito

El circuito de la Cerradura se podría simular con un módulo relé para controlar cargas de potencia, pero por tiempo de montaje se realizará con un led (conexión que se vio en el laboratorio #1). El circuito con el modulo será como se muestra en la **Fig. 4**.

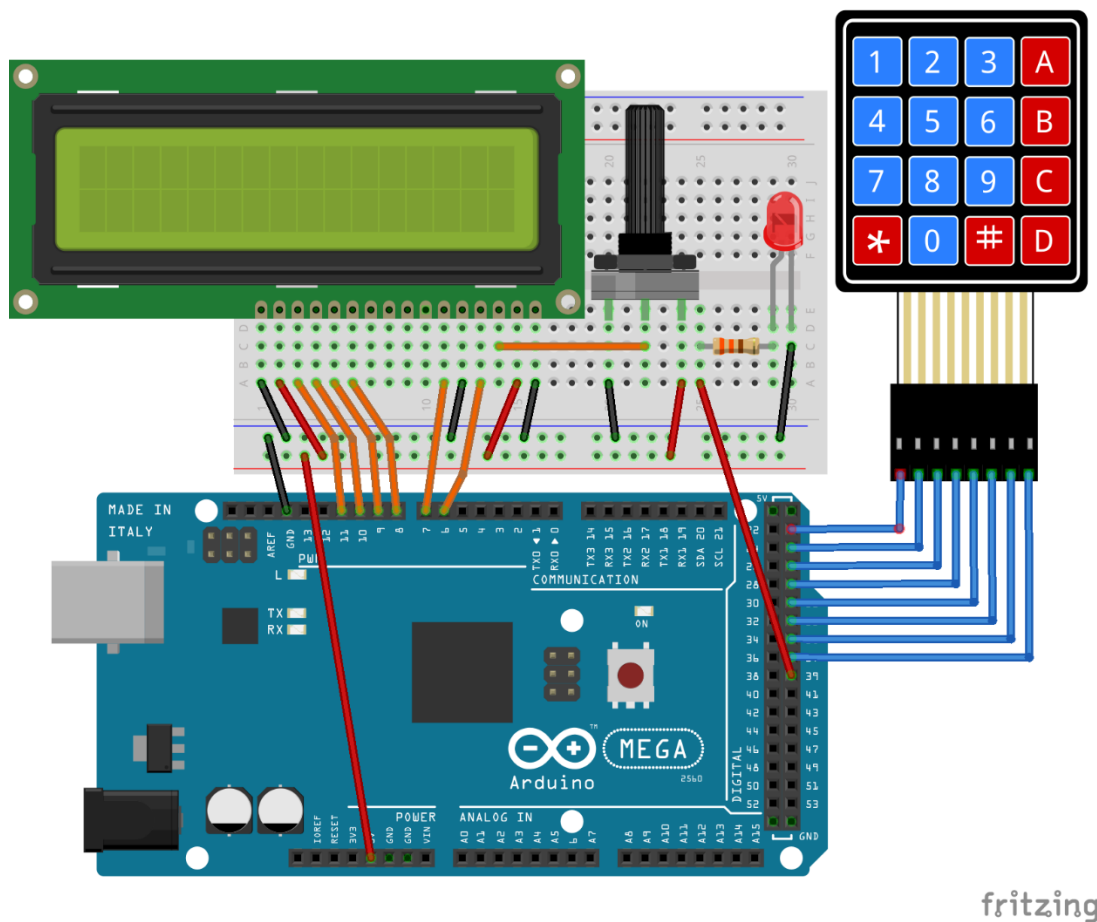


Fig. 4 Diagrama de Montaje de Cerradura Electrónica.

Actividad

Realice cambios en la programación para que la cerradura cumpla lo siguiente:

- Cuando se ingrese los dígitos de la contraseña, se visualicen asteriscos y no el carácter.
- Cuente los intentos fallidos en el ingreso de la contraseña, tendrá un conteo máximo de 3 intentos, después el sistema se bloqueará y se liberará con las teclas * o #.

6. Actividades Propuestas

Realizar cambios en la programación para que la contraseña almacenada de la cerradura, el usuario la pueda modificar y al mismo tiempo sea almacenada en la memoria EEPROM del microcontrolador. Este almacenaje se hace porque la memoria EEPROM retiene esa información, aunque no posea voltaje.