**ALARMA LDR**

Ricardo Ignacio Shepstone Aramburu

DNI: 77553533-V

Propósito del proyecto

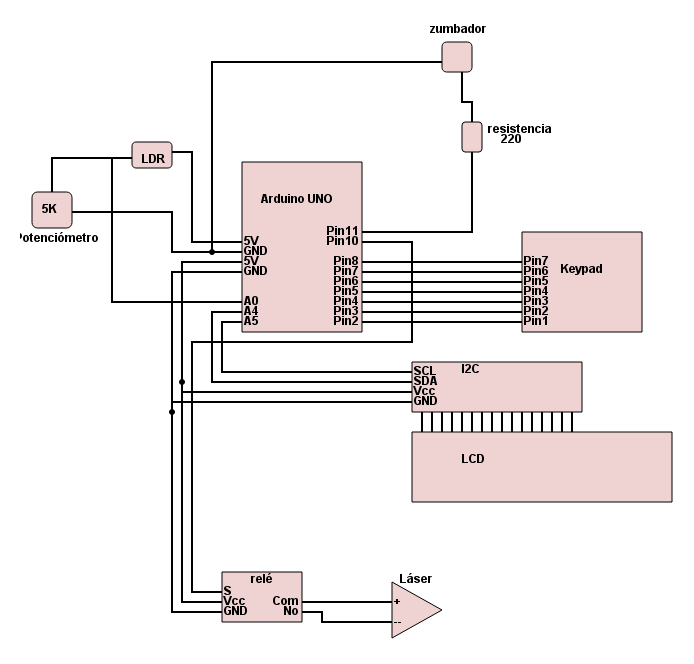
Se trata de una alarma de seguridad que funciona con una resistencia ldr y con un láser. En su funcionamiento normal el láser ilumina la resistencia, y en el momento en el que una persona corte el haz de luz, el arduino detectará el cambio del valor de la resistencia ldr, y hará sonar una alarma. El proyecto también incluye un keypad y una pantalla lcd de 2x16 para poder desactivar o activar el sistema mediante una contraseña, además incluye un contador que hace que la alarma se dispare tras tres intentos fallidos al introducir la contraseña.

Listado de materiales y presupuesto

|  |  |  |
| --- | --- | --- |
| Nº | Nombre | Coste(€) |
| 1 | Arduino UNO | 2.97 |
| 1 | 4x4 Matrix keypad | 0.46 |
| 1 | I2c interfaz lcd | 0.74 |
| 1 | Pantalla lcd 2x16 | 1.78 |
| 1 | Resistencia ldr | 0.19 |
| 1 | Circuito láser | 1.41 |
| 1 | Potenciómetro 5 kΩ | 0.3 |
| 1 | Zumbador piezoeléctrico 5 V | 0.18 |
| 1 | Módulo relé 5 V | 0.48 |
| 1 | Resistencia 220 Ω | 0.01 |
| 6 | Cables jumper | 0.09 |
| 14 | Cables jumper macho-hembra | 0.18 |
| 1 | Cable bipolar 70 cm para circuito láser | 0.22 |
| 1 | Protoboard 16.6x5.5x0.85 cm | 0.98 |
|  | Total | 9.99 |

Consulta de precios en: https://es.aliexpress.com/

Esquemático del hardware



Código fuente comentado

//librerías

#include <Password.h> //librería para la contraseña

#include <Keypad.h> //librería para el Keypad

#include <Wire.h> //librería para el módulo I2C

#include <LiquidCrystal\_I2C.h> //librería para la pantalla LCD controlada por I2C

//declaración de pines

const byte ldr=0; //define el pin A0 donde se hace la lectura analógica del divisor de ensión de la LDR

const byte laser=10; //define el pin donde irá conectado el relé que activará el laser

const byte zumbador=11; //define el pin donde irá conectado el zumbador

//declaración de variables

byte pospass=5; //una variable utilizada para ir escribiendo la contraseña en la LCD

int medida; //variable donde se almacena la lectura analógica

byte estadosistema=0; //variable usada para ver si el sistema está activo (1) o inactivo (0)

byte estadoalarma=0; //variable usada para el estado de la alarma, (1) activo y (0) inactivo

int i=4;

int n=0; //dos variables que usaremos para contadores

//Establecemos contraseña

Password password = Password("1234"); //asignamos la contraseña, en este caso es 1234

//declaración de keypad

const byte rows = 4; //cuatro filas

const byte cols = 3; //tres columnas

char keys[rows][cols] = { //definimos como va a ser nuestra matriz

{'1','2','3'},

{'4','5','6'},

{'7','8','9'},

{'#','0','\*'}

};

byte rowPins[rows] = {5, 4, 3, 2}; //los pines a los que hay que conectar las filas

byte colPins[cols] = {6, 7, 8}; //los pines a los que hay que conectar las columnas

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, rows, cols ); //se declara el Keypad

//declaración de I2C para LCD

LiquidCrystal\_I2C lcd(0x3F,16,2);

void setup() {

// Iniciar LCD

lcd.init();

//Encender la luz de fondo

lcd.backlight();

pinMode(laser, OUTPUT); //se define como pin de salida

pinMode(zumbador, OUTPUT); //se define como pin de salida

keypad.addEventListener(keypadEvent); //hace que se ejecute un evento si se usa el keypad

// Escribimos el Mensaje en el LCD.

lcd.setCursor(0, 0);

lcd.print("Ric Shepstone");

lcd.setCursor(0, 1);

lcd.print("Aramburu");

delay(2000); //se escribe el mensaje Ric Shepstone Aramburu en la LCD y aparece durante 2 segundos

lcd.clear(); //se borra el mensaje

delay(500);

lcd.setCursor(0, 0);

lcd.print("Proyecto PI:");

lcd.setCursor(0, 1);

lcd.print("Alarma LDR");

delay(2000); //se escribe el mensaje Proyecto PI: Alarma LDR en la LCD y aparece durante 2 segundos

lcd.clear();//se borra el mensaje

delay(500);

lcd.setCursor(0, 0);

lcd.print("Comprobacion:");

lcd.setCursor(0, 1);

lcd.print("apuntar laser");

digitalWrite(laser, HIGH);

do{

medida=analogRead(ldr);

delay(5000);

}while(medida < 750);

lcd.clear();

delay(500);

lcd.setCursor(0, 0);

lcd.print("sistema");

lcd.setCursor(0, 1);

lcd.print("instalado");

//esta parte sirve para instalar el laser en su posición fija y que apunte a la LDR durante un tiempo (como máximo 5 segundos),

//si todo está correcto, sale el mensaje de sistema instalado

delay(2000);

lcd.clear();

digitalWrite(laser, LOW);

n=0;

lcd.print("ajustar sensor ");

while(n <= 10){

n++;

medida=analogRead(ldr);

lcd.setCursor(0, 1);

lcd.print(medida);

lcd.print(" <750");

delay(1000);

}

lcd.clear();

//ajustamos la sensibilidad de la medida con el potenciómetro, durante este bucle while se irá mostrando la medida que toma arduino de la ldr sin tener el láser encendido,

//así poder ajustar la medida para que sea menor que 750, que es nuestro umbral para que se dispare la alarma

menu(); //lleva al programa al menú principal

}

void loop() {

keypad.getKey(); //hace lectura del keypad

medida=analogRead(ldr); //hace lectura de la lDR

delay(50);

if (medida<750 && estadosistema==1){

alarma();

}

//dispara la alarma con la función alarma siempre y cuando la medida sea menor a 750 y el sistema esté activo

}

void keypadEvent(KeypadEvent eKey){

switch (keypad.getState()){ //definimos en que casos se llama a la función, en este caso cuando pulsamos

case PRESSED:

if(pospass-5>=5){

password.reset();

lcd.clear();

menu();

pospass=5;

}

//esta condición hace que si excedemos los 5 caracteres se vuelva a la posición inicial

lcd.setCursor((pospass++),1); //nos permite ir incrementando la posición en la que escribimos la contraseña en la lcd de uno en uno

//en el siguiente switch definimos que pasa al pulsar las distintas teclas

switch (eKey){

case '#':

pospass=5;

validarpass();

break;

//en caso de pulsar "#" se llama a la función validarpass

case '\*':

password.reset(); //resetea la contraseña adivinada

lcd.setCursor(5, 1);

lcd.print(" ");

pospass=5;

break;

//en caso de pulsar "\*" se borra la contraseña que se estuviese escribiendo en caso de haber cometido algun fallo

default:

password.append(eKey);

lcd.print("\*");

//este caso se refiere a cualquier otra tecla, en este caso los números. Al pulsarlos añadimos ese número a la contraseña adivinada (password.append)

//y en la pantalla lcd aparece un asterisco

}

}

}

void alarma(){

password.reset();

estadoalarma = 1;

tone (zumbador, 440); //función que se ejecuta cuando se dispara la alarma, hace que el zumbador emita ruido y asigna un 1 al estado de alarma

}

void validarpass(){ //función para validar la contraseña adivinada con la contraseña correcta

if (password.evaluate()){ // si password.evaluate() es cierta, o sea nuestra contraseña adivinada corresponde con la contraseña la función es cierta

if (estadosistema==0 && estadoalarma==0){ //en caso de que el sistema este desactivado y la alarma no esté sonando, se llama a la función activar, que activa el sistema

activar();

}

else{

desactivar(); //en caso contrario, desactivar el sistema llamando a la función desactivar

}

}

else{

pass\_error(); //en caso de que la contraseña no sea correcta, ejecutar la función pass\_error

}

}

void pass\_error(){ //función pass\_error, incluye un contador que va decreciendo cada vez que se llama a la función, después de 3 intentos, la alarma se disparará

i--;

if(i==0){

alarma();

}

password.reset(); //resetea la contraseña adivinada

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("PASS ERRONEA"); //escribe pass erronea en la LCD

lcd.setCursor(10, 1);

lcd.print(i);

lcd.print("tries"); //indaca el número de intentos antes de que se dispare la alarma

delay(1000);

menu(); //vuelve al menú

}

void activar(){ //función que activa el sistema

password.reset(); //resetea la contraseña adivinada

digitalWrite (laser,HIGH); //activa el láser

delay(1000);

if (analogRead(ldr) > 750){ //esta condición hace que sólo se pueda pasar a estado activo si el láser ilumina la ldr

estadosistema = 1; //asigna el valor 1 a estadosistema

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("SISTEMA ACTIVO"); //escribe sistema activo en la lcd

password.reset(); //resetea la contraseña adivinada

delay(1000);

menu();//vuelve al menú

}

else{

desactivar(); //en caso de que el láser no apunte a la ldr, no se activa el sistema

}

}

void desactivar(){ //con esta función se define el estado desactivado

digitalWrite (laser,LOW); //desactiva el láser

i=4; //resetea nuestro contador de intentos de contraseña

estadoalarma = 0; //asigna el valor 0 a nuestra variable estado alarma

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("SISTEMA INACTIVO");//escribe sistema inactivo en la lcd

noTone(zumbador); //apaga el zumbador en caso de que la alarma este disparada

estadosistema=0; //asigna el valor 0 a nuestra variable estadosistema

password.reset(); //resetea la contraseña adivinada

delay(1000);

menu();//vuelve al menú principal

}

void menu(){ //define el menú principal

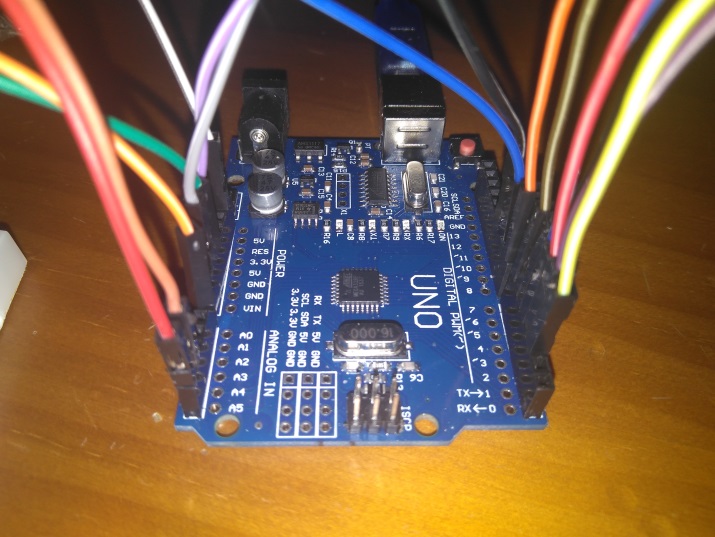
lcd.setCursor(0, 1);

lcd.print("Pass:"); //en nuestro menú principal aparece en la parte de abajo "Pass:", ya que son las funciones activar o desactivar que escriben arriba si el sistema está activo o inactivo

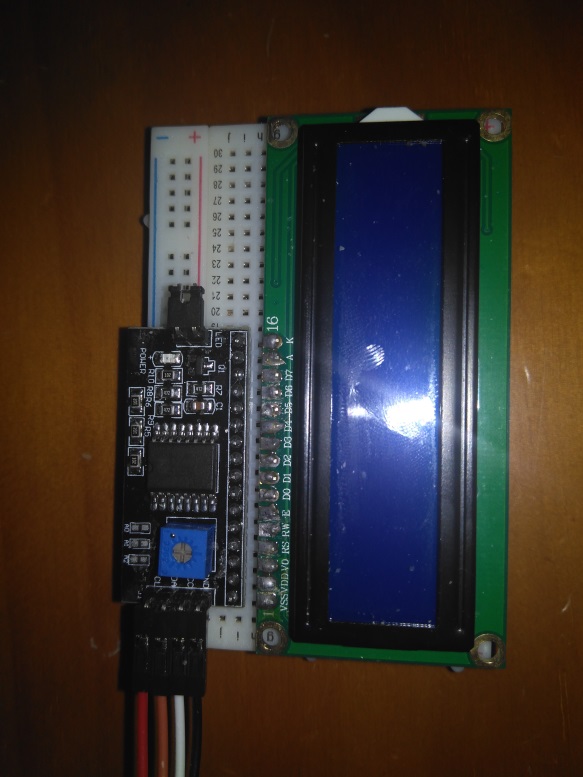
}

Fotografías del montaje

1. Arduino UNO



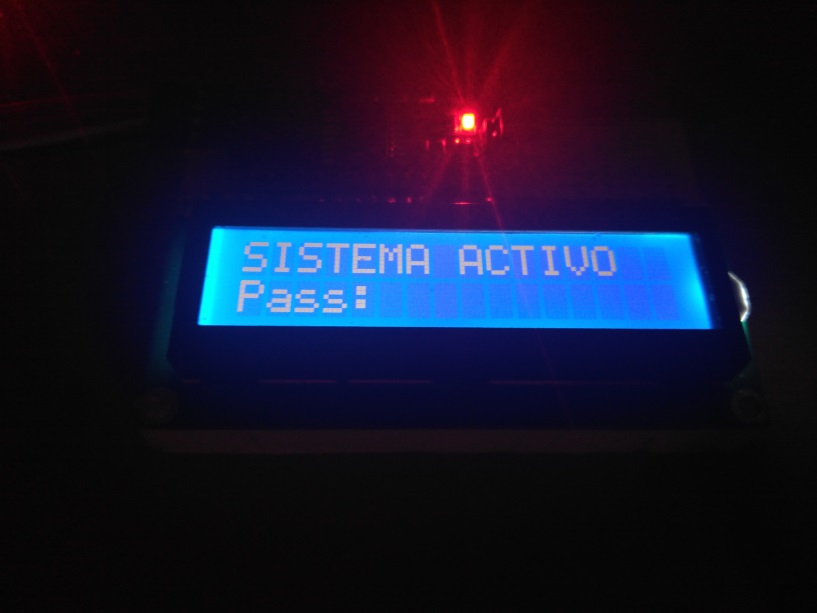
1. Pantalla LCD con módulo I2C



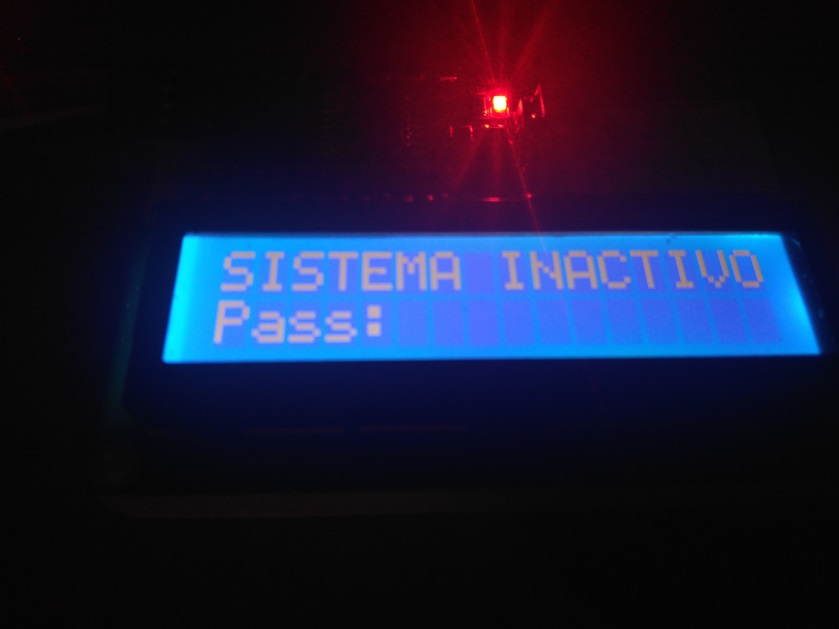
* 1. Ajuste sensor



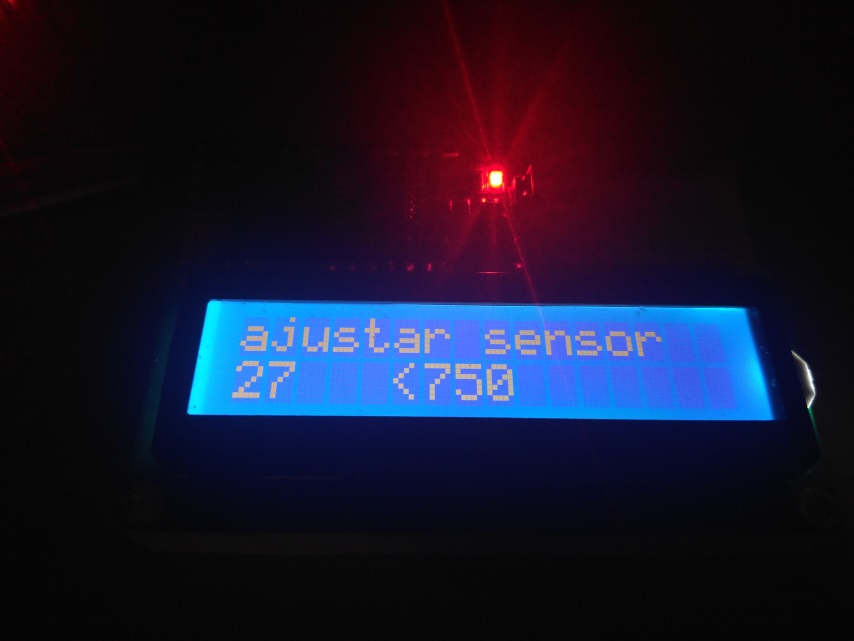
* 1. Sistema activo



* 1. Sistema inactivo



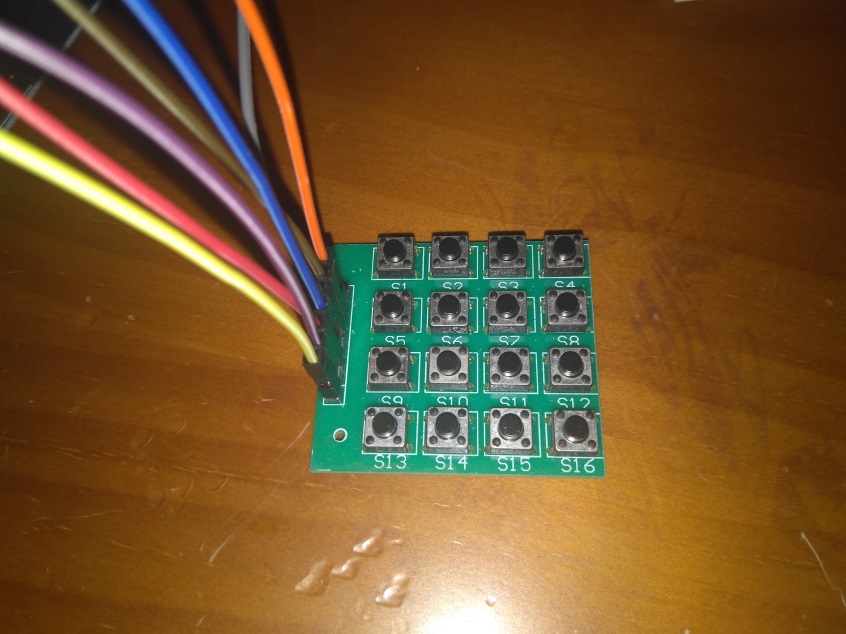
* 1. Pass errónea



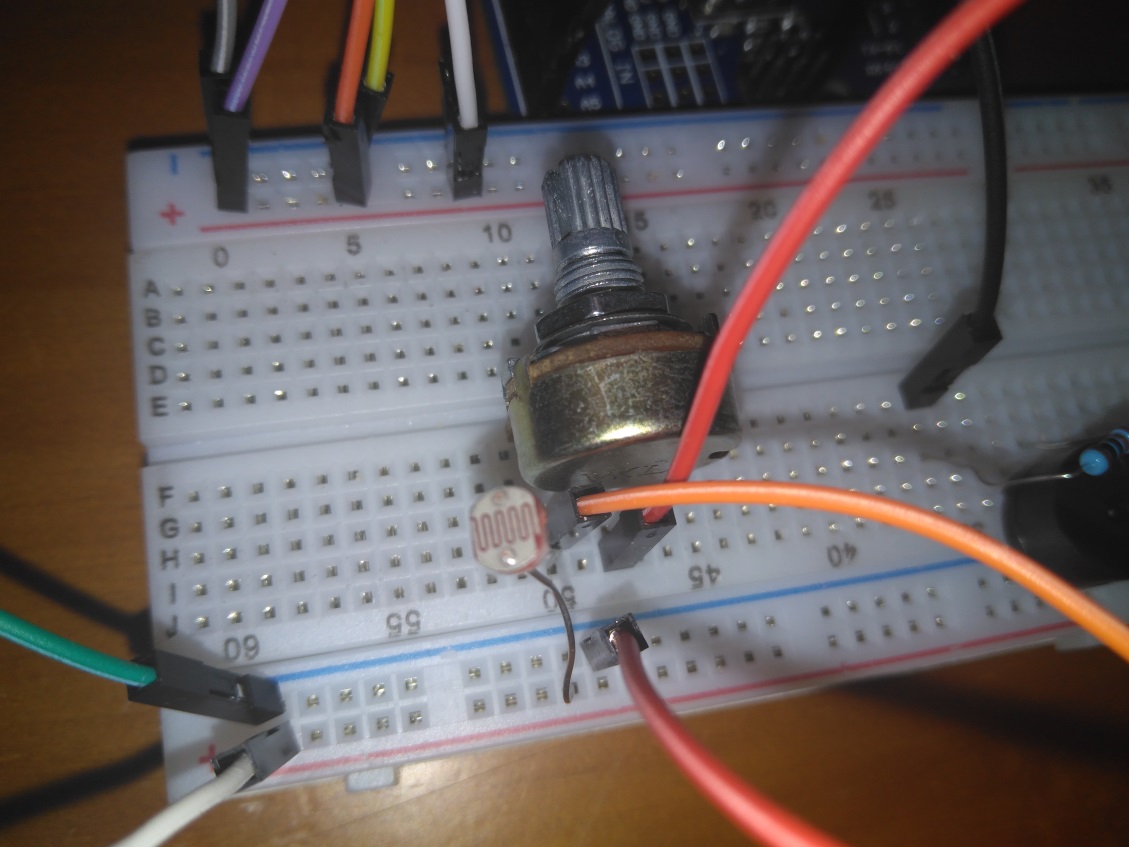
1. Relé con láser



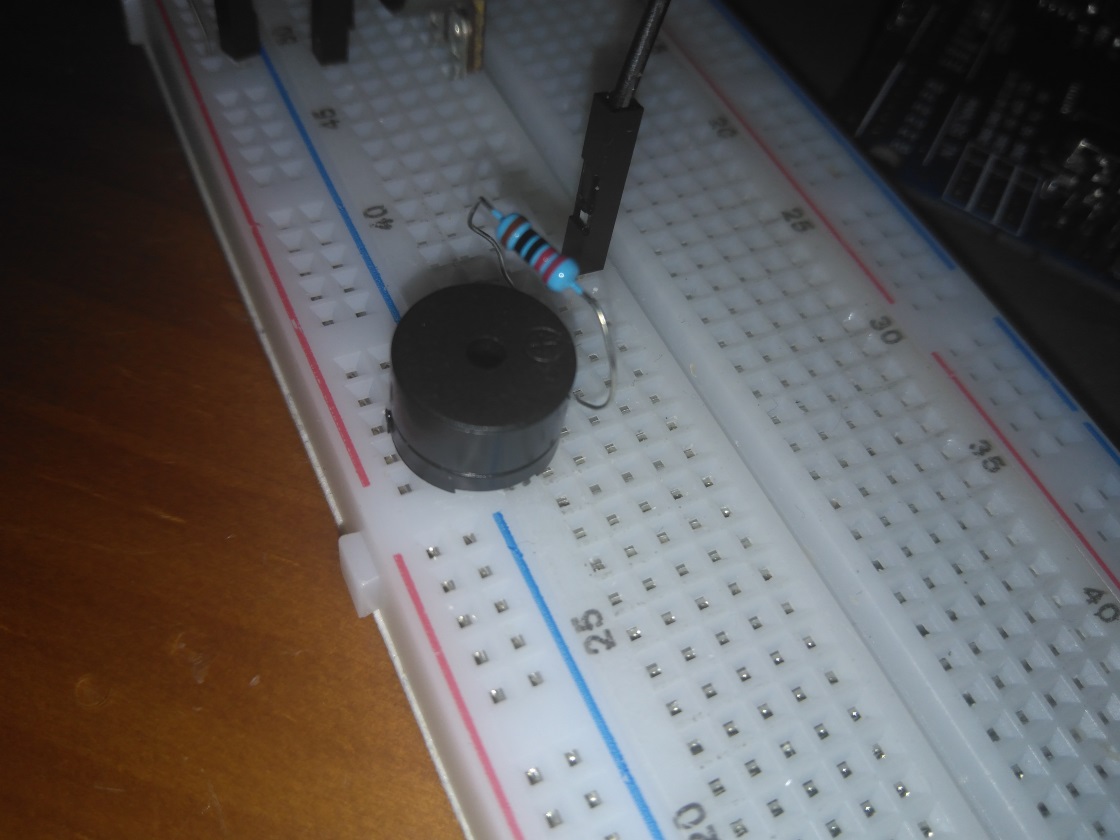
1. Keypad



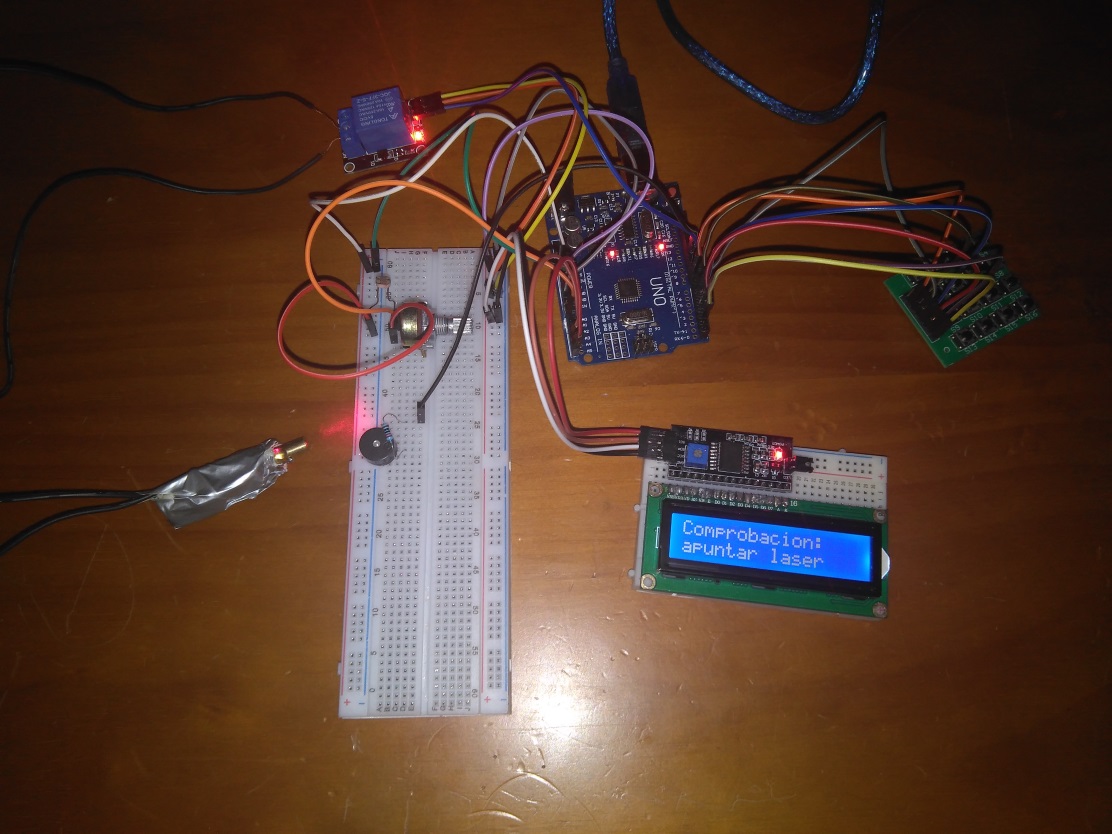
1. LDR y potenciómetro



1. Zumbador



1. Montaje total



Bibliografía

Arduino: curso práctico de formación

Óscar Torrente Artero

Enlaces empleados

**Librerías:**

<https://playground.arduino.cc/Code/Password#Download>

<https://playground.arduino.cc/Code/Keypad>

<https://www.arduino.cc/en/Reference/Wire>

<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>

**Información:**

<https://www.prometec.net/teclados-matriciales/>