PROYECTO COMPUTO INTEGRADO

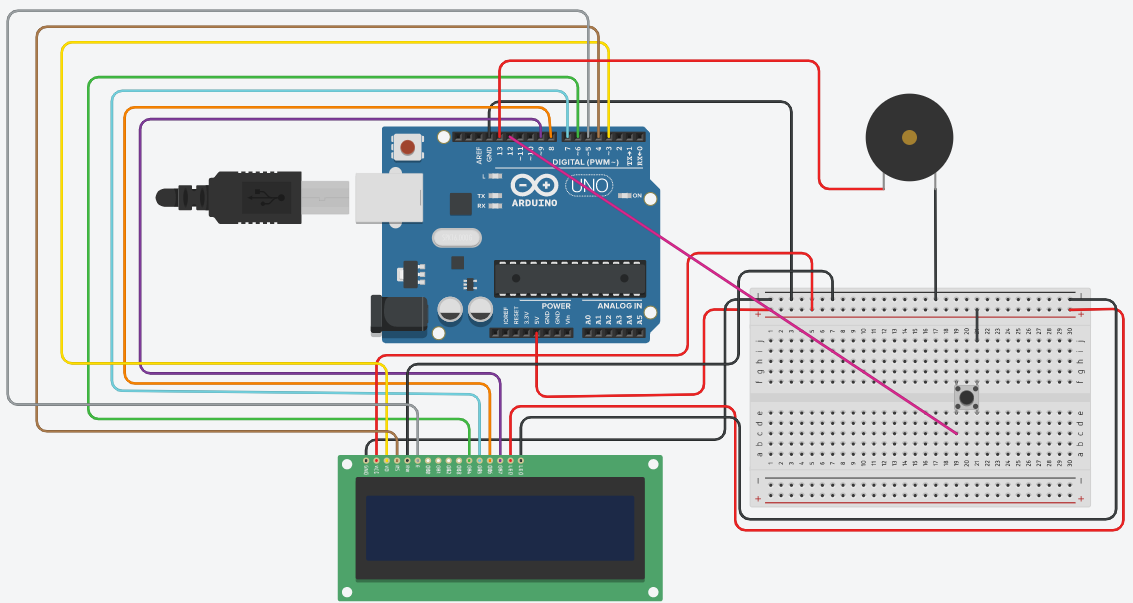
COMPONENTES

* 1 Pantalla LCD de 16x2, Display LCD de 16x2
* 1 ProtoBoard de 400 puntos
* 1 Pila Alcalina De 9 Volts
* 1 ARDUINO UNO REV3
* 1 Buzzer Activo
* 1 Push button

Librerias

* 1. LiquidCrystal
* 2. Wire.h

Circuito



Código fuente

#include <LiquidCrystal.h>

#include <Wire.h>

// Intensidad de contraste y configuracion de LCD

int V0 **=** 3**;**

int RS **=** 4**;**

int E **=** 5**;**

//Salidas digitales necesarias para mostrar informacion en LCD

int D4 **=** 6**;**

int D5 **=** 7**;**

int D6 **=** 8**;**

int D7 **=** 9**;**

// Creacion de instancia de clase de LiquidCrystal para enviar la decodificacion del codigo morce a la pantalla LCD

LiquidCrystal lcd**(**RS**,** E**,** D4**,** D5**,** D6**,** D7**);**

// Clave de señal del pin 12 a GND 7

#define CLAVE 12

// Timbre desde el pin 13 hasta GND 8

#define BUZZER 13

// Configuracion inicial

void setup**()**

**{**

pinMode**(**CLAVE**,** INPUT\_PULLUP**);**

pinMode**(**BUZZER**,** OUTPUT**);**

analogWrite**(**V0**,** 50**);**

lcd**.**begin**(**16**,** 2**);**

lcd**.**setCursor**(**0**,** 0**);**

**}**

//Declaracion de variables globales

float duracionDePulsacion **=** 200.0**;**

boolean PrevS **=** **false;**

long tiempo1**,** tiempo2**;**

boolean S;

String codigoMorse = "";

int x = 0;

int y = 0;

// Ejecucion de decodificacion

void loop()

{

S = !digitalRead(CLAVE);

if (S)

{

if (S != PrevS)

{

tiempo1 = millis();

DecodificadorPause(tiempo2);

}

digitalWrite(BUZZER, HIGH);

}

else

{

if (S != PrevS)

{

tiempo2 = millis();

Decodificador(tiempo1);

}

digitalWrite(BUZZER, LOW);

}

if (abs(millis() - tiempo2) > duracionDePulsacion \* 10)

{

DecodificadorPause(tiempo2);

}

PrevS = S;

}

// Metodo que recibe el tiempo de duracion de cada una de las pulsaciones para determinar el código morse que se introduce.

void Decodificador(long tiempoDeInicio)

{

char pulsacion = '?';

long tiempo = abs(millis() - tiempoDeInicio); // tiempos de duración de la señal (un punto o un guión)

float duracionDelPunto = duracionDePulsacion / 3.0;

if (tiempo <= 2)

return;

if (tiempo <= duracionDelPunto)

{

pulsacion = '.';

}

else if (tiempo > duracionDePulsacion)

{

pulsacion = '-';

}

else if ((tiempo > (duracionDePulsacion + duracionDelPunto) / 1.9) && tiempo <= duracionDePulsacion)

{

pulsacion = '-';

}

else

pulsacion = '.';

if (pulsacion == '-')

{

if (tiempo > duracionDePulsacion)

duracionDePulsacion++;

if (tiempo < duracionDePulsacion)

duracionDePulsacion--;

}

else if (pulsacion == '.')

{

if (tiempo > duracionDePulsacion / 3.0)

duracionDePulsacion++;

if (tiempo < duracionDePulsacion / 3.0)

duracionDePulsacion--;

}

codigoMorse += pulsacion;

}

// Pausa el decodificador e ingresa un espacio despues de cada elemento del alfabento que se muestra en pantalla

void DecodificadorPause(long tiempoDeInicio)

{

if (codigoMorse == "")

return;

char pulsacion = '?';

long tiempo = abs(millis() - tiempoDeInicio);

if (tiempo > duracionDePulsacion - duracionDePulsacion / 40)

{

Decodificador();

}

if (tiempo > duracionDePulsacion \* 2)

{

Decodificador();

lcd.print("");

}

}

// Compara las pulsaciones del usuario

void Decodificador()

{

static String morseArray[] = {

".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..", ".---", "-.-", ".-..", "--", "-.", "---", ".--.", "--.-",

".-.", "...", "-", "..-", "...-", ".--", "-..-", "-.--", "--..","E" };

int i = 0;

while (morseArray[i] != "E")

{

if (morseArray[i] == codigoMorse)

{

lcd.print((char)('A' + i));

x++;

break;

}

i++;

}

if (morseArray[i] == "E")

{

//Entra cuando el elemento morse no existe dentro del arreglo anteriormente definido

}

ActualizaCursor();

codigoMorse = "";

}

// Actualiza la posicion del cursor previniendo que los datos salgan de la pantalla.

void ActualizaCursor()

{

lcd.setCursor(x, y);

//Valida que el cursor no se exeda de los caracteres maximos que puede mostrar la pantalla en X

if (x >= 16)

{

x = 0;

y++;

}

//Valida que el cursor no se exeda de los caracteres maximos que puede mostrar la pantalla en Y y limpia la pantalla

if (y >= 2)

{

lcd.clear();

x = y = 0;

}

}