**NODEMCU\_laser\_LDR\_medida\_directa\_calculo\_distancia\_OLED**

/\*

El programa primero detecta la diferencia entre la señal con y sin Láser encendido, calculando un valor de corte como la semisuma.

Después . si la masa corta dos veces por el lado izquierdo de su recorrido el vector, al volver, cuenta el espaciado entre

el primer y tercer valle de los números, averiguando el período. Podemos comprobar con la fórmula: T = 2\*Pi\*RAIZ(longitud/g)

\*/

#include <SPI.h>

#include <Wire.h>

#include <Adafruit\_GFX.h> //Nucleo de la librería gráfica.

#include <Adafruit\_SSD1306.h> //Librería para pantallas OLED monocromas de 128x64 y 128x32

#include <Fonts/FreeMonoBoldOblique12pt7b.h>

#include <Fonts/FreeSerifBold9pt7b.h> // Biblioteca de fuentes: https://learn.adafruit.com/adafruit-gfx-graphics-library/using-fonts

#define pinBoton D5

#define pinLASER D8

// =============================================================================================

// IMPORTANTE: definición de pantalla, y RESET que funciona, sin tocar librería adafruit\_SSD1306

// =============================================================================================

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)

#define OLED\_RESET 0 // Reset pin # (or -1 if sharing Arduino reset pin)

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

// =============================================================================================

long tiempo = 0;

long tiempo1 = 0;

long diferencia = 0;

long tiempocorte1=0;

long tiempocorte2=0;

float voltage = 0;

float voltageAnterior = 0;

float valorCorte = 0;

int contar = 0;

boolean estado = false;

boolean estadoBoton = LOW;

const unsigned char PROGMEM logo [] = {

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0xF0, 0x00, 0x00, 0x07, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0xFC, 0x00, 0x00, 0x07, 0xF0, 0x00, 0x03, 0xF8, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFC, 0x00, 0x00, 0x07, 0xC0, 0x00, 0x00, 0xF8, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFE, 0x00, 0x00, 0x07, 0x80, 0x00, 0x00, 0x78, 0x00, 0x00,

0x00, 0x00, 0x00, 0x7E, 0x00, 0xFF, 0xFE, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00,

0x00, 0x00, 0x00, 0xFF, 0x01, 0xFF, 0xFE, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,

0x00, 0x00, 0x03, 0xFF, 0xC0, 0xFF, 0xFE, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,

0x00, 0x00, 0x03, 0xFF, 0xF0, 0xFF, 0xFF, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,

0x00, 0x00, 0x03, 0xFF, 0xF0, 0xFF, 0xFF, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,

0x00, 0x70, 0x03, 0xFF, 0xF0, 0x7F, 0xFF, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,

0x03, 0xFC, 0x07, 0xFF, 0xF8, 0x3F, 0xFF, 0x80, 0x00, 0x06, 0x03, 0x00, 0x30, 0x18, 0x00, 0x00,

0x03, 0xFE, 0x07, 0xFF, 0xF8, 0x1F, 0xFF, 0x80, 0x00, 0x06, 0x1F, 0xE1, 0xFE, 0x18, 0x00, 0x00,

0x0F, 0xFF, 0x07, 0xFF, 0xF8, 0x07, 0xFF, 0x80, 0x00, 0x06, 0x30, 0x33, 0x03, 0x18, 0x00, 0x00,

0x0F, 0xFF, 0xC7, 0xFF, 0xFC, 0xE3, 0xFF, 0xC0, 0x00, 0x06, 0x20, 0x12, 0x01, 0x18, 0x00, 0x00,

0x1F, 0xFF, 0xC3, 0xFF, 0xFC, 0xE0, 0x7F, 0xC0, 0x00, 0x06, 0x60, 0x0C, 0x01, 0x98, 0x00, 0x00,

0x1F, 0xFF, 0xC1, 0xFF, 0xFC, 0xFC, 0x1F, 0xC0, 0x00, 0x06, 0x63, 0x0C, 0x31, 0x98, 0x00, 0x00,

0x1F, 0xFF, 0xE0, 0xFF, 0xFE, 0x7F, 0x03, 0xE0, 0x00, 0x06, 0x60, 0x0C, 0x01, 0x98, 0x00, 0x00,

0x1F, 0xFF, 0xE0, 0x7F, 0xFE, 0x7F, 0xC0, 0xE0, 0x00, 0x06, 0x20, 0x12, 0x01, 0x18, 0x00, 0x00,

0x0F, 0xFF, 0xE0, 0x1F, 0xFE, 0x7F, 0xF8, 0x40, 0x00, 0x06, 0x30, 0x33, 0x03, 0x18, 0x00, 0x00,

0x0F, 0xFF, 0xF7, 0x8F, 0xFF, 0x3F, 0xFC, 0x00, 0x00, 0x06, 0x1C, 0xE1, 0xCE, 0x18, 0x00, 0x00,

0x0F, 0xFF, 0xF7, 0xC0, 0xFF, 0x3F, 0xFF, 0x80, 0x00, 0x06, 0x07, 0x80, 0x78, 0x18, 0x00, 0x00,

0x07, 0xFF, 0xF7, 0xF8, 0x1F, 0x3F, 0xFF, 0xC0, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,

0x03, 0xFF, 0xFB, 0xFE, 0x0F, 0x1F, 0xFF, 0xE0, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,

0x00, 0xFF, 0xFB, 0xFF, 0x03, 0x9F, 0xFF, 0xF0, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,

0x08, 0xFF, 0xFB, 0xFF, 0xE0, 0x1F, 0xFF, 0xF8, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,

0x0E, 0x0F, 0xFD, 0xFF, 0xF8, 0x0F, 0xFF, 0xF8, 0x00, 0x06, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,

0x0F, 0x03, 0xFD, 0xFF, 0xFF, 0x0F, 0xFF, 0xFC, 0x00, 0x07, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00,

0x0F, 0xF0, 0xFD, 0xFF, 0xFF, 0x0F, 0xFF, 0xFC, 0x00, 0x07, 0x80, 0x00, 0x00, 0x78, 0x00, 0x00,

0x07, 0xF8, 0x1C, 0xFF, 0xFF, 0x8F, 0xFF, 0xFC, 0x00, 0x07, 0xC0, 0x00, 0x00, 0xF8, 0x00, 0x00,

0x07, 0xFF, 0x04, 0xFF, 0xFF, 0xC7, 0xFF, 0xF8, 0x00, 0x07, 0xF8, 0x00, 0x07, 0xF8, 0x00, 0x00,

0x07, 0xFF, 0xE0, 0xFF, 0xFF, 0xE7, 0xFF, 0xF0, 0x00, 0x07, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x00,

0x03, 0xFF, 0xE0, 0xFF, 0xFF, 0xE7, 0xFF, 0x00, 0x00, 0x07, 0xFF, 0xFF, 0xFF, 0xF8, 0x00, 0x00,

0x03, 0xFF, 0xFE, 0x7F, 0xFF, 0xF3, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x03, 0xFF, 0xFE, 0x7F, 0xFF, 0xF3, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x01, 0xFF, 0xFE, 0x7F, 0xFF, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x01, 0xFF, 0xFF, 0x3F, 0xFF, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x01, 0xFF, 0xFF, 0x3F, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x01, 0xFF, 0xFF, 0x3F, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x01, 0x80, 0x06, 0x00, 0x00, 0x00, 0x00,

0x00, 0xFF, 0xFF, 0x9F, 0xC0, 0x00, 0x07, 0x00, 0x00, 0x03, 0xC0, 0x0F, 0x00, 0x00, 0x00, 0x00,

0x00, 0xFF, 0xFF, 0x9C, 0x00, 0x00, 0x1F, 0x80, 0x00, 0x03, 0xC0, 0x0F, 0x00, 0x00, 0x00, 0x00,

0x00, 0xFF, 0xFF, 0x80, 0x00, 0x00, 0x3F, 0x80, 0x00, 0x01, 0x80, 0x06, 0x00, 0x00, 0x00, 0x00,

0x00, 0x7F, 0xFF, 0x00, 0x00, 0x00, 0x63, 0x80, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x7F, 0xF8, 0x00, 0x00, 0x00, 0xE2, 0x00, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x7F, 0x80, 0x00, 0x00, 0x00, 0xE0, 0x0F, 0x8F, 0xE7, 0x0E, 0x1C, 0x71, 0xC7, 0x39, 0xC0,

0x00, 0x3C, 0x00, 0x00, 0x00, 0x00, 0xE0, 0x1F, 0xBF, 0xFF, 0x0E, 0x7D, 0xF3, 0x9F, 0xFF, 0xC0,

0x00, 0x20, 0x00, 0x00, 0x00, 0x00, 0xF0, 0x3B, 0xBF, 0x7F, 0x7F, 0xFD, 0xF3, 0x9F, 0xFF, 0xC0,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFC, 0x3F, 0x8E, 0x06, 0x1C, 0x18, 0x63, 0x87, 0xBD, 0xC0,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7E, 0x7F, 0x1C, 0x0E, 0x1C, 0x38, 0xE3, 0x0E, 0x73, 0x80,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3E, 0x7C, 0x1C, 0x0E, 0x1C, 0x38, 0xE7, 0x0E, 0x73, 0x80,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1E, 0x78, 0x1C, 0x0E, 0x18, 0x38, 0xE7, 0x0E, 0x73, 0x80,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0E, 0x71, 0x9C, 0x0C, 0xF9, 0xB3, 0xEF, 0x6E, 0x73, 0xB0,

0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x0E, 0x73, 0xB8, 0x1F, 0xFF, 0xFF, 0xFF, 0xFC, 0xE3, 0xE0,

0x00, 0x00, 0x00, 0x00, 0x00, 0x1C, 0x1C, 0x7F, 0x38, 0x1F, 0x3E, 0x7D, 0xFF, 0xDC, 0xE7, 0xC0,

0x00, 0x00, 0x00, 0x00, 0x00, 0x1E, 0x3C, 0x7E, 0x38, 0x1E, 0x3C, 0x79, 0xE7, 0x1C, 0xE3, 0x80,

0x00, 0x00, 0x00, 0x00, 0x00, 0x1F, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x1F, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

};

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// the setup routine runs once when you press reset:

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void setup() {

display.begin(SSD1306\_SWITCHCAPVCC, 0x3C); // //Inicializa pantalla con en la dirección 0x3D para la conexión I2C.

pinMode(pinBoton,INPUT);

pinMode(pinLASER,OUTPUT);

// initialize serial communication at 9600 bits per second:

Serial.begin(115200);

digitalWrite(pinLASER,LOW); // El láser empieza, lógicamente, apagado

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Enciendo la pantalla

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

display.fillScreen(0); //Limpiamos la pantalla

// Logo

display.drawBitmap(0,0,logo,128,64,1); // drawBitmap (int16\_t x, int16\_t y, uint8\_t \* mapa de bits, int16\_t w, int16\_t h, uint16\_t color)

// https://www.brainy-bits.com/create-arduino-array-from-pictures/

display.display(); //Refrescamos la pantalla para visualizarlos

delay(4000); // 4 segundos

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// the loop routine runs over and over again forever:

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void loop() {

if (valorCorte<=0) {vcorte();} // Si el valor de corte no está establecido, lo establece

// Encender el laser o no

int estadoBotonAhora = digitalRead(pinBoton);

if (estadoBotonAhora==HIGH && estadoBoton==LOW) {

estado = estado xor 1;

display.fillScreen(0);

display.setFont();

display.setTextSize(2); // Tamaño

display.setTextColor(1,0); // Color (1,0) ó 0,1 invertido

display.setCursor(0,5); // Posiciones x e y

display.println("Voltaje de corte: ");

display.setCursor(0,5+18); // Posiciones x e y

display.println(String (valorCorte)+" V");

display.display();

delay(500);

}

estadoBoton = estadoBotonAhora;

digitalWrite(pinLASER,estado);

// Fin de encender el laser

tiempo = millis();

// read the input on analog pin 0:

int sensorValue = analogRead(A0);

// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):

float voltage = sensorValue;

String voltaje = String(voltage);

voltaje.replace(".",",");

// Serial.println(String(tiempo)+" ; "+voltaje+" ; corte: "+String(valorCorte));

// Reconocimiento del paso

**if (estado==HIGH && voltage<valorCorte && voltageAnterior>=valorCorte) { // Si se sucede un paso hacia abajo por el valor de Corte**

**contar = contar + 1;**

**if (contar==1) { // Primer paso**

**tiempocorte1 = tiempo;**

Serial.println("=========================================");

Serial.println ("Valor de corte: "+String(valorCorte));

Serial.println("Primer paso en el valor de tiempo (ms): "+String(float(tiempocorte1)));

} **else if (contar==3 && tiempocorte1>0) { // Tercer paso**

**diferencia = tiempo-tiempocorte1;**

**Serial.println("T= "+String( float(diferencia) /1000)+"s; L="+longitud(diferencia)+" cm");**

Serial.println("=========================================");

// Pantalla OLED

display.fillScreen(0);

display.setFont();

display.setTextSize(2); // Tamaño

display.setTextColor(1,0); // Color (1,0) ó 0,1 invertido

display.setCursor(0,10); // Posiciones x e y

display.println("T="+String( float(diferencia) /1000)+"s");

display.setCursor(0,10+16);

display.println("T="+String(diferencia)+"ms");

display.setCursor(0,10+16\*2);

display.println("L="+longitud(diferencia)+"cm");

display.display();

// Refresco tiempo

contar = 0;

tiempocorte1=0;

} // Fin de la condición que el conteo sea par

}

voltageAnterior = voltage;

// delay(1);

// Serial.println(voltage);

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// funciones

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**// Funcion que devuelve la longitud**

**String longitud(long periodo) {**

**// perido viene en milisegundos, convertir a segundos.**

**float calculo = 100\*9.8\*pow(((periodo/1000.0) / (2.0 \* 3.141596254)),2.0) ;**

**// devuelve la longitud del péndulo en centímetros**

**String cadena = String(calculo);**

**cadena.replace(".",",");**

**return cadena;**

**}**

// Funcion que reconoce el valor de corte

void vcorte() {

// Al principio establece el valor de corte. Una vez establecido, esta rutina no se vuelve a ejecutar

digitalWrite(pinLASER,LOW);

valorCorte = analogRead(A0);

delay(10);

digitalWrite(pinLASER,HIGH);

delay(10);

valorCorte = (valorCorte +analogRead(A0))/2; // La semisuma

digitalWrite(pinLASER,estado); }