#include <Adafruit\_LiquidCrystal.h>

Adafruit\_LiquidCrystal pantallaLCD(0); // Pantalla LCD

const int pinEntrada = A0;

const int botonComienzo = 2;

const int botonDetalles = 3;

bool estadoAdquisicion = false; // Estado de adquisición de datos

int valorAnalogico = 0;

float nivelVoltaje = 0.0;

float frecuenciaMedida = 0.0;

unsigned long tiempoPrevio = 0;

unsigned long tiempoActualizado = 0;

float periodoLectura = 0.0;

unsigned long ultimaLectura = 0; // Control de tiempo de lectura

// Almacenar los valores de la señal

float \*muestras; // muestras de la señal

int cantidadMuestras = 100;

void inicializarPantalla() {

pantallaLCD.begin(16, 2); // LCD de 16x2

pantallaLCD.setCursor(0, 0);

pantallaLCD.print(" Preparando... ");

delay(2000); // Espera 2 segundos

pantallaLCD.setCursor(0, 1);

pantallaLCD.print(" Todo en orden ");

muestras = new float[cantidadMuestras]; // memoria para las muestras

}

void adquirirDatos() {

unsigned long tiempoActualizado = millis();

if (tiempoActualizado - ultimaLectura >= 1000) {

valorAnalogico = analogRead(pinEntrada);

nivelVoltaje = (valorAnalogico \* 5.0) / 1023.0;

periodoLectura = (tiempoActualizado - tiempoPrevio) / 1000.0; // En segundos

if (periodoLectura > 0) {

frecuenciaMedida = 1.0 / periodoLectura;

}

tiempoPrevio = tiempoActualizado;

pantallaLCD.clear();

pantallaLCD.setCursor(0, 0);

pantallaLCD.print("Amplitud: ");

pantallaLCD.print(nivelVoltaje, 2); // 2 decimales para el voltaje

pantallaLCD.setCursor(0, 1);

pantallaLCD.print("Frecuencia: ");

pantallaLCD.print(frecuenciaMedida, 2); // 2 decimales para la frecuencia

ultimaLectura = tiempoActualizado;

}

}

void mostrarFormaOnda() {

pantallaLCD.clear();

pantallaLCD.setCursor(0, 0);

leerMuestras();

String tipoOnda = determinarForma();

pantallaLCD.print(" Tipo de onda: ");

pantallaLCD.setCursor(0, 1);

pantallaLCD.print(tipoOnda);

delay(500);

}

void leerMuestras() {

for (int i = 0; i < cantidadMuestras; i++) {

muestras[i] = analogRead(pinEntrada) \* (5.0 / 1023.0); // Lee y convierte a voltaje

}

}

String determinarForma() {

bool ondaTriangular = true;

bool ondaCuadrada = true;

bool ondaSenoidal = true;

for (int i = 1; i < cantidadMuestras - 1; i++) {

if (!((muestras[i-1] < muestras[i] && muestras[i] > muestras[i+1]) || (muestras[i-1] > muestras[i] && muestras[i] < muestras[i+1]))) {

ondaTriangular = false;

break;

}

}

for (int i = 1; i < cantidadMuestras - 1; i++) {

if (!(muestras[i] == muestras[i-1] || muestras[i] == 0 || muestras[i] == 5)) {

ondaCuadrada = false;

break;

}

}

int cambiosDePendiente = 0;

for (int i = 1; i < cantidadMuestras; i++) {

if ((muestras[i-1] < muestras[i] && muestras[i] > muestras[i+1]) || (muestras[i-1] > muestras[i] && muestras[i] < muestras[i+1])) {

cambiosDePendiente++;

}

}

if (cambiosDePendiente > 5) ondaSenoidal = false;

if (ondaTriangular)

return "Triangular";

else if (ondaCuadrada)

return "Cuadrada";

else if (ondaSenoidal)

return "Senoidal";

else

return "Desconocida";

}

void setup() {

inicializarPantalla();

pinMode(pinEntrada, INPUT);

pinMode(botonComienzo, INPUT);

pinMode(botonDetalles, INPUT);

}

void loop() {

bool inicioPresionado = digitalRead(botonComienzo) == HIGH;

bool infoPresionado = digitalRead(botonDetalles) == HIGH;

if (inicioPresionado) {

pantallaLCD.clear();

pantallaLCD.setCursor(0, 0);

pantallaLCD.print(" Analizando... ");

estadoAdquisicion = !estadoAdquisicion; // Cambia el estado de adquisición

delay(500);

pantallaLCD.clear();

if (estadoAdquisicion) {

pantallaLCD.setCursor(0, 0);

pantallaLCD.print(" Tomando");

pantallaLCD.setCursor(0, 1);

pantallaLCD.print(" Muestras ");

} else {

pantallaLCD.setCursor(0, 0);

pantallaLCD.print(" En pausa... ");

}

}

if (estadoAdquisicion) {

adquirirDatos();

}

if (infoPresionado) {

mostrarFormaOnda();

}

}