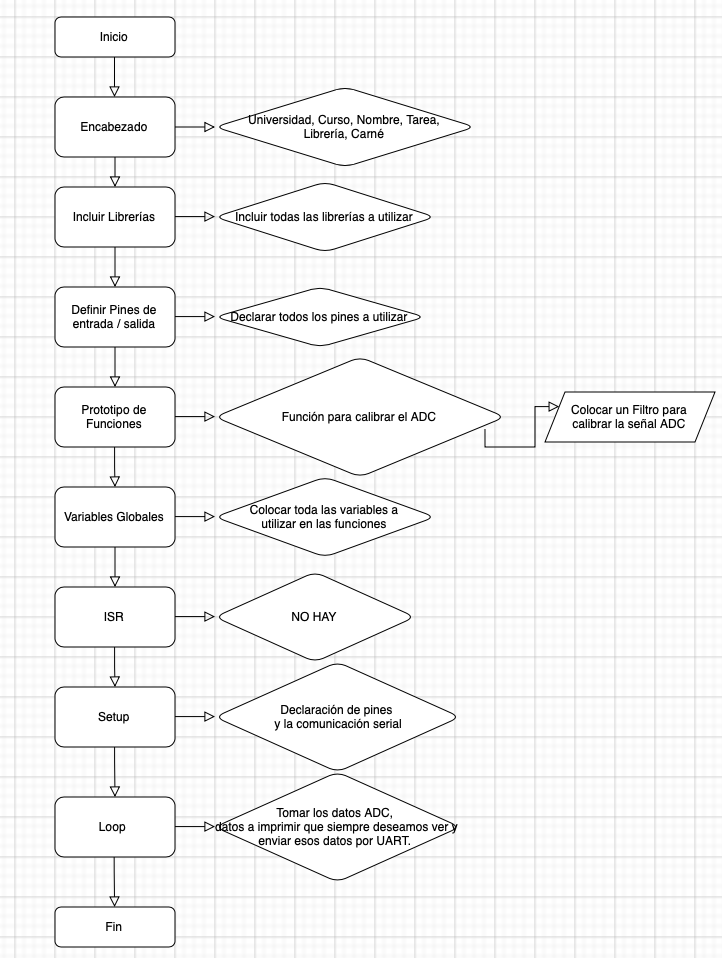
Ing. Jeremy Mejía 19846 Ingeniería en Audio, UVG

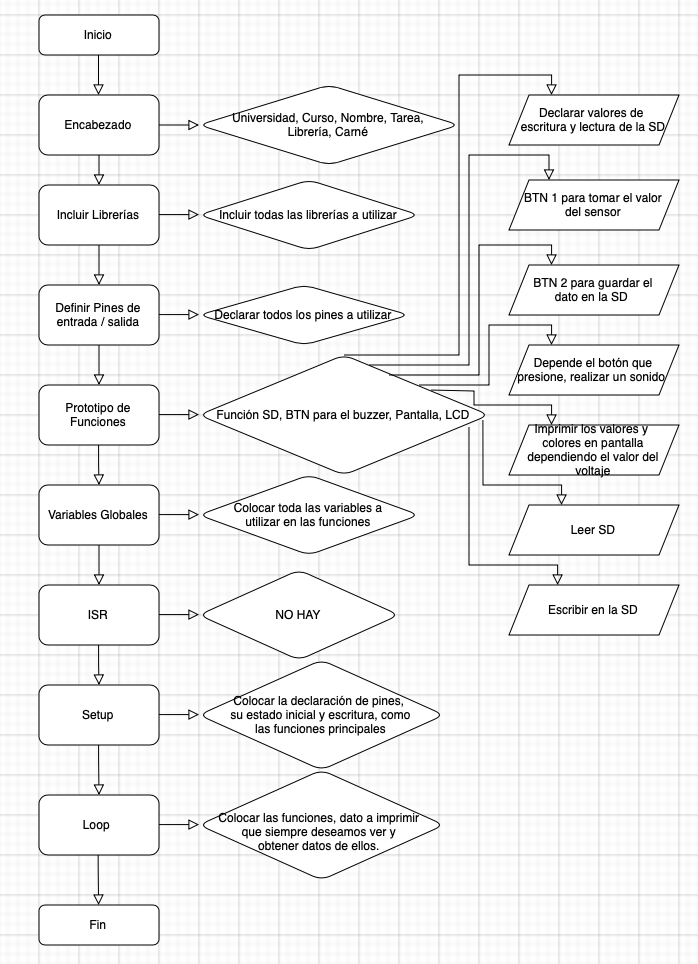
**Proyecto 2 – Micrófono Vúmetro**

**(Noviembre, 2021)**

**Diagrama de Flujo del ESP32**

****

**Diagrama de Flujo de la Tiva C**

****

**Análisis**

El proyecto 2 consiste en utilizar la comunicación serial entre el microcontrolador ESP32 y la TIVA C, para esto se utilizó un sensor; en mi caso un micrófono, para poder enviar los datos obtenidos desde el ESP32 a la TIVA C. Otros requitos funcionales del proyectos son que al momento de presionar los botónes, estos tengan la función de hacer la medición del sensor y el otro guardar ese valor, así también conectar un buzzer para tener una referencia auditiva de lo que está sucediendo y por último poder desplegar de manera gráfica en la pantala TFT el valor del sensor.

El micrófono fue conectado al ESP32 y dentro del código se calibró la señal ADC. El dato obtenido fue enviado por medio de la comunicación UART a la TIVA C, y este valor fue disignado por funciones para poder marcar la cantidad de intensidad por decibeles tanto en el monitor serial, como también en la pantalla TFT. En la pantalla TFT se muestra el deseño de un vúmetro, dependiendo de la intensidad del sonido, este se va mostrando en pantalla y regresa a cero. El botón 1 sirve para tomar el dato de la medición y el botón 2 es para guardar ese dato en la SD.

**Código ESP32**

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

// Universidad del Valle de Guatemala

// BE3015 - Electrónica Digital 2

// Jeremy Mejía

// Proyecto 2

// Arduino

// Carné: 19846

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

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

// Librerías

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

#include <Arduino.h>

#include "esp\_adc\_cal.h"

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

// Definición de Pines

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

//Pin del mic

#define Mic 13

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

// Prototipo de Funciones

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

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

// Variables Globales

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

int voltaje;

int adcRaw;

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

// Configuración

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

void setup() {

Serial.begin(115200);

Serial2.begin(115200);

pinMode(Mic, INPUT);

}

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

// Loop Principal

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

void loop() {

adcRaw = analogRead(Mic);

voltaje = ReadVoltage(adcRaw);

//voltaje = map(voltaje, 0, 4095, 1, 10);

Serial.println(voltaje);

Serial2.write(voltaje);

delay(100);

}

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

// Función Calibrar ADC

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

float ReadVoltage(int ADC\_Raw){ //Calibración del ADC del ESP32

esp\_adc\_cal\_characteristics\_t adc\_chars;

esp\_adc\_cal\_characterize(ADC\_UNIT\_1, ADC\_ATTEN\_11db, ADC\_WIDTH\_12Bit, 1100, &adc\_chars);

return (esp\_adc\_cal\_raw\_to\_voltage(ADC\_Raw, &adc\_chars));

}

**Código TIVA C**

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

**// Universidad del Valle de Guatemala**

**// BE3015 - Electrónica Digital 2**

**// Jeremy Mejía**

**// Proyecto 2**

**// Arduino**

**// Carné: 19846**

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

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

**// Librerías**

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

**//Pantalla**

**#include <stdint.h>**

**#include <stdbool.h>**

**#include <TM4C123GH6PM.h>**

**#include "inc/hw\_ints.h"**

**#include "inc/hw\_memmap.h"**

**#include "inc/hw\_types.h"**

**#include "driverlib/debug.h"**

**#include "driverlib/gpio.h"**

**#include "driverlib/interrupt.h"**

**#include "driverlib/rom\_map.h"**

**#include "driverlib/rom.h"**

**#include "driverlib/sysctl.h"**

**#include "driverlib/timer.h"**

**#include "bitmaps.h"**

**#include "font.h"**

**#include "lcd\_registers.h"**

**//SD**

**#include <SPI.h>**

**#include <SD.h>**

**#include "pitches.h"**

**//Definición de Pines**

**#define LCD\_RST PD\_0**

**#define LCD\_CS PD\_1**

**#define LCD\_RS PD\_2**

**#define LCD\_WR PD\_3**

**#define LCD\_RD PE\_1**

**int DPINS[] = {PB\_0, PB\_1, PB\_2, PB\_3, PB\_4, PB\_5, PB\_6, PB\_7};**

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

**// Definición de Pines**

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

**//Pines de los leds**

**#define ledR PF\_1**

**#define ledA PF\_2**

**#define ledV PF\_3**

**//Botones**

**#define NOTE\_C4\_1 260**

**#define btn1 PF\_4**

**#define btn2 PF\_0**

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

**// Prototipo de Funciones**

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

**void LCD\_Init(void);**

**void LCD\_CMD(uint8\_t cmd);**

**void LCD\_DATA(uint8\_t data);**

**void SetWindows(unsigned int x1, unsigned int y1, unsigned int x2, unsigned int y2);**

**void LCD\_Clear(unsigned int c);**

**void H\_line(unsigned int x, unsigned int y, unsigned int l, unsigned int c);**

**void V\_line(unsigned int x, unsigned int y, unsigned int l, unsigned int c);**

**void Rect(unsigned int x, unsigned int y, unsigned int w, unsigned int h, unsigned int c);**

**void FillRect(unsigned int x, unsigned int y, unsigned int w, unsigned int h, unsigned int c);**

**void LCD\_Print(String text, int x, int y, int fontSize, int color, int background);**

**void LCD\_Bitmap(unsigned int x, unsigned int y, unsigned int width, unsigned int height, unsigned char bitmap[]);**

**void LCD\_Sprite(int x, int y, int width, int height, unsigned char bitmap[], int columns, int index, char flip, char offset);**

**void BTN1(void);**

**void BTN2(void);**

**void Pantalla(void);**

**//SD**

**void writeSD(void);**

**void readSD(void);**

**extern uint8\_t fondo[];**

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

**// Variables Globales**

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

**//Variables del vúmetro**

**String T1 = "-22";**

**String T2 = "-16";**

**String T3 = "-12";**

**String T4 = "-6";**

**String T5 = "-3";**

**String T6 = "0";**

**String T7 = "3";**

**String T8 = "6";**

**String T9 = "10";**

**String T10 = "16";**

**String datodB;**

**File archivo;**

**//Valores globales**

**int valor = 0;**

**int dB = 0;**

**//Valores del Buzzer**

**int buzzerPin = PF\_1;**

**int melody[] = {**

**NOTE\_C4\_1, NOTE\_D4};**

**int melody2[] = {**

**NOTE\_D4, NOTE\_C4\_1};**

**int noteDurations[] = {**

**4, 2};**

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

**// Configuración**

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

**void setup() {**

**Serial.begin(115200);**

**Serial2.begin(115200);**

**while (!Serial) {**

**;**

**}**

**Serial.print("Initializing SD card...");**

**pinMode(PA\_3, OUTPUT);**

**SPI.setModule(0);**

**if (!SD.begin(PA\_3)) {**

**Serial.println("initialization failed!");**

**return;**

**}**

**Serial.println("initialization done.");**

**//Pines de Salida**

**pinMode(buzzerPin,OUTPUT);**

**pinMode(btn1, INPUT\_PULLUP);**

**pinMode(btn2, INPUT\_PULLUP);**

**readSD();**

**SysCtlClockSet(SYSCTL\_SYSDIV\_2\_5 | SYSCTL\_USE\_PLL | SYSCTL\_OSC\_MAIN | SYSCTL\_XTAL\_16MHZ);**

**Serial.begin(115200);**

**GPIOPadConfigSet(GPIO\_PORTB\_BASE, 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7, GPIO\_STRENGTH\_8MA, GPIO\_PIN\_TYPE\_STD\_WPU);**

**Serial.println("Inicio");**

**LCD\_Init();**

**LCD\_Clear(0x00);**

**FillRect(0, 0, 240, 320, 0x0000);**

**LCD\_Print(T1, 109, 298, 1, 0xffff, 0x0000);**

**LCD\_Print(T2, 109, 266, 1, 0xffff, 0x0000);**

**LCD\_Print(T3, 109, 234, 1, 0xffff, 0x0000);**

**LCD\_Print(T4, 113, 202, 1, 0xffff, 0x0000);**

**LCD\_Print(T5, 113, 170, 1, 0xffff, 0x0000);**

**LCD\_Print(T6, 121, 138, 1, 0xffff, 0x0000);**

**LCD\_Print(T7, 121, 106, 1, 0xffff, 0x0000);**

**LCD\_Print(T8, 121, 74, 1, 0xffff, 0x0000);**

**LCD\_Print(T9, 117, 42, 1, 0xffff, 0x0000);**

**LCD\_Print(T10, 117, 10, 1, 0xffff, 0x0000);**

**}**

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

**// Loop Principal**

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

**void loop() {**

**if(digitalRead(btn1) == LOW){**

**BTN1();**

**Pantalla();**

**}**

**if(digitalRead(btn2) == LOW){**

**BTN2();**

**writeSD();**

**}**

**if(Serial2.available() > 0){**

**valor = Serial2.read();**

**}**

**//Fila 1**

**FillRect(0, 288, 48, 31, 0x0000);**

**FillRect(49, 288, 48, 31, 0x0000);**

**FillRect(147, 288, 48, 31, 0x0000);**

**FillRect(196, 288, 48, 31, 0x0000);**

**//Fila 2**

**FillRect(0, 256, 48, 31, 0x0000);**

**FillRect(49, 256, 48, 31, 0x0000);**

**FillRect(147, 256, 48, 31, 0x0000);**

**FillRect(196, 256, 48, 31, 0x0000);**

**//Fila 3**

**FillRect(0, 224, 48, 31, 0x0000);**

**FillRect(49, 224, 48, 31, 0x0000);**

**FillRect(147, 224, 48, 31, 0x0000);**

**FillRect(196, 224, 48, 31, 0x0000);**

**//Fila 4**

**FillRect(0, 192, 48, 31, 0x0000);**

**FillRect(49, 192, 48, 31, 0x0000);**

**FillRect(147, 192, 48, 31, 0x0000);**

**FillRect(196, 192, 48, 31, 0x0000);**

**//Fila 5**

**FillRect(0, 160, 48, 31, 0x0000);**

**FillRect(49, 160, 48, 31, 0x0000);**

**FillRect(147, 160, 48, 31, 0x0000);**

**FillRect(196, 160, 48, 31, 0x0000);**

**//Fila 6**

**FillRect(0, 128, 48, 31, 0x0000);**

**FillRect(49, 128, 48, 31, 0x0000);**

**FillRect(147, 128, 48, 31, 0x0000);**

**FillRect(196, 128, 48, 31, 0x0000);**

**//Fila 7**

**FillRect(0, 96, 48, 31, 0x0000);**

**FillRect(49, 96, 48, 31, 0x0000);**

**FillRect(147, 96, 48, 31, 0x0000);**

**FillRect(196, 96, 48, 31, 0x0000);**

**//Fila 8**

**FillRect(0, 64, 48, 31, 0x0000);**

**FillRect(49, 64, 48, 31, 0x0000);**

**FillRect(147, 64, 48, 31, 0x0000);**

**FillRect(196, 64, 48, 31, 0x0000);**

**//Fila 9**

**FillRect(0, 32, 48, 31, 0x0000);**

**FillRect(49, 32, 48, 31, 0x0000);**

**FillRect(147, 32, 48, 31, 0x0000);**

**FillRect(196, 32, 48, 31, 0x0000);**

**//Fila 10**

**FillRect(0, 0, 48, 31, 0x0000);**

**FillRect(49, 0, 48, 31, 0x0000);**

**FillRect(147, 0, 48, 31, 0x0000);**

**FillRect(196, 0, 48, 31, 0x0000);**

**delay(100);**

**}**

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

**// Función ReadSD**

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

**void readSD(void){**

**archivo = SD.open("vúmetro.txt");**

**if (archivo) {**

**Serial.println("El archivo contiene lo siguiente:");**

**// read from the file until there's nothing else in it:**

**while (archivo.available()) {**

**Serial.write(archivo.read());**

**}**

**// close the file:**

**archivo.close();**

**} else {**

**// if the file didn't open, print an error:**

**Serial.println("error opening vúmetro.txt");**

**}**

**}**

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

**// Función WriteSD**

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

**void writeSD(void){**

**archivo = SD.open("vúmetro.txt", FILE\_WRITE);**

**if (archivo) {**

**Serial.print("Writing to vúmetro.txt...");**

**archivo.println(valor);**

**// close the file:**

**archivo.close();**

**Serial.println("done.");**

**} else {**

**// if the file didn't open, print an error:**

**Serial.println("error opening vúmetro.txt");**

**}**

**}**

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

**// Función BTN1**

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

**void BTN1(void){**

**for (int thisNote = 0; thisNote < 2; thisNote++) {**

**// to calculate the note duration, take one second**

**// divided by the note type.**

**//e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.**

**int noteDuration = 1000/noteDurations[thisNote];**

**tone(buzzerPin, melody[thisNote],noteDuration);**

**int pauseBetweenNotes = noteDuration + 50; //delay between pulse**

**delay(pauseBetweenNotes);**

**noTone(buzzerPin);**

**}**

**}**

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

**// Función BTN2**

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

**void BTN2(void){**

**for (int thisNote = 0; thisNote < 2; thisNote++) {**

**// to calculate the note duration, take one second**

**// divided by the note type.**

**//e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.**

**int noteDuration = 1000/noteDurations[thisNote];**

**tone(buzzerPin, melody2[thisNote],noteDuration);**

**int pauseBetweenNotes = noteDuration + 50; //delay between pulse**

**delay(pauseBetweenNotes);**

**noTone(buzzerPin);**

**}**

**}**

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

**// Función BTN2**

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

**void Pantalla(void){**

**dB = valor;**

**if(dB > 10 && dB <= 20){**

**analogWrite(ledR, 0);**

**analogWrite(ledA, 0);**

**analogWrite(ledV, 255);**

**datodB = "Intensidad: -22dB";**

**//Fila 1**

**FillRect(0, 288, 48, 31, 0x07E0);**

**FillRect(49, 288, 48, 31, 0x07E0);**

**FillRect(147, 288, 48, 31, 0x07E0);**

**FillRect(196, 288, 48, 31, 0x07E0);**

**Serial.println("Intensidad: -22dB");**

**delay(1000);**

**}**

**else if(dB > 20 && dB <= 25){**

**analogWrite(ledR, 0);**

**analogWrite(ledA, 0);**

**analogWrite(ledV, 255);**

**datodB = "Intensidad: -16dB";**

**//Fila 1**

**FillRect(0, 288, 48, 31, 0x07E0);**

**FillRect(49, 288, 48, 31, 0x07E0);**

**FillRect(147, 288, 48, 31, 0x07E0);**

**FillRect(196, 288, 48, 31, 0x07E0);**

**//Fila 2**

**FillRect(0, 256, 48, 31, 0x07E0);**

**FillRect(49, 256, 48, 31, 0x07E0);**

**FillRect(147, 256, 48, 31, 0x07E0);**

**FillRect(196, 256, 48, 31, 0x07E0);**

**Serial.println("Intensidad: -16dB");**

**delay(1000);**

**}**

**else if(dB > 25 && dB <= 30 ){**

**analogWrite(ledR, 0);**

**analogWrite(ledA, 0);**

**analogWrite(ledV, 255);**

**datodB = "Intensidad: -12dB";**

**//Fila 1**

**FillRect(0, 288, 48, 31, 0x07E0);**

**FillRect(49, 288, 48, 31, 0x07E0);**

**FillRect(147, 288, 48, 31, 0x07E0);**

**FillRect(196, 288, 48, 31, 0x07E0);**

**//Fila 2**

**FillRect(0, 256, 48, 31, 0x07E0);**

**FillRect(49, 256, 48, 31, 0x07E0);**

**FillRect(147, 256, 48, 31, 0x07E0);**

**FillRect(196, 256, 48, 31, 0x07E0);**

**//Fila 3**

**FillRect(0, 224, 48, 31, 0x07E0);**

**FillRect(49, 224, 48, 31, 0x07E0);**

**FillRect(147, 224, 48, 31, 0x07E0);**

**FillRect(196, 224, 48, 31, 0x07E0);**

**Serial.println("Intensidad: -12dB");**

**delay(1000);**

**}**

**else if (dB > 30 && dB <= 35){**

**analogWrite(ledR, 0);**

**analogWrite(ledA, 0);**

**analogWrite(ledV, 255);**

**datodB = "Intensidad: -6dB";**

**//Fila 1**

**FillRect(0, 288, 48, 31, 0x07E0);**

**FillRect(49, 288, 48, 31, 0x07E0);**

**FillRect(147, 288, 48, 31, 0x07E0);**

**FillRect(196, 288, 48, 31, 0x07E0);**

**//Fila 2**

**FillRect(0, 256, 48, 31, 0x07E0);**

**FillRect(49, 256, 48, 31, 0x07E0);**

**FillRect(147, 256, 48, 31, 0x07E0);**

**FillRect(196, 256, 48, 31, 0x07E0);**

**//Fila 3**

**FillRect(0, 224, 48, 31, 0x07E0);**

**FillRect(49, 224, 48, 31, 0x07E0);**

**FillRect(147, 224, 48, 31, 0x07E0);**

**FillRect(196, 224, 48, 31, 0x07E0);**

**//Fila 4**

**FillRect(0, 192, 48, 31, 0x07E0);**

**FillRect(49, 192, 48, 31, 0x07E0);**

**FillRect(147, 192, 48, 31, 0x07E0);**

**FillRect(196, 192, 48, 31, 0x07E0);**

**Serial.println("Intensidad: -6dB");**

**delay(1000);**

**}**

**else if (dB > 35 && dB <= 40){**

**analogWrite(ledR, 0);**

**analogWrite(ledA, 0);**

**analogWrite(ledV, 255);**

**datodB = "Intensidad: -3dB";**

**//Fila 1**

**FillRect(0, 288, 48, 31, 0x07E0);**

**FillRect(49, 288, 48, 31, 0x07E0);**

**FillRect(147, 288, 48, 31, 0x07E0);**

**FillRect(196, 288, 48, 31, 0x07E0);**

**//Fila 2**

**FillRect(0, 256, 48, 31, 0x07E0);**

**FillRect(49, 256, 48, 31, 0x07E0);**

**FillRect(147, 256, 48, 31, 0x07E0);**

**FillRect(196, 256, 48, 31, 0x07E0);**

**//Fila 3**

**FillRect(0, 224, 48, 31, 0x07E0);**

**FillRect(49, 224, 48, 31, 0x07E0);**

**FillRect(147, 224, 48, 31, 0x07E0);**

**FillRect(196, 224, 48, 31, 0x07E0);**

**//Fila 4**

**FillRect(0, 192, 48, 31, 0x07E0);**

**FillRect(49, 192, 48, 31, 0x07E0);**

**FillRect(147, 192, 48, 31, 0x07E0);**

**FillRect(196, 192, 48, 31, 0x07E0);**

**//Fila 5**

**FillRect(0, 160, 48, 31, 0x07E0);**

**FillRect(49, 160, 48, 31, 0x07E0);**

**FillRect(147, 160, 48, 31, 0x07E0);**

**FillRect(196, 160, 48, 31, 0x07E0);**

**Serial.println("Intensidad: -3dB");**

**delay(1000);**

**}**

**else if (dB > 40 && dB <= 45){**

**analogWrite(ledR, 255);**

**analogWrite(ledA, 0);**

**analogWrite(ledV, 255);**

**datodB = "Intensidad: 0dB";**

**//Fila 1**

**FillRect(0, 288, 48, 31, 0x07E0);**

**FillRect(49, 288, 48, 31, 0x07E0);**

**FillRect(147, 288, 48, 31, 0x07E0);**

**FillRect(196, 288, 48, 31, 0x07E0);**

**//Fila 2**

**FillRect(0, 256, 48, 31, 0x07E0);**

**FillRect(49, 256, 48, 31, 0x07E0);**

**FillRect(147, 256, 48, 31, 0x07E0);**

**FillRect(196, 256, 48, 31, 0x07E0);**

**//Fila 3**

**FillRect(0, 224, 48, 31, 0x07E0);**

**FillRect(49, 224, 48, 31, 0x07E0);**

**FillRect(147, 224, 48, 31, 0x07E0);**

**FillRect(196, 224, 48, 31, 0x07E0);**

**//Fila 4**

**FillRect(0, 192, 48, 31, 0x07E0);**

**FillRect(49, 192, 48, 31, 0x07E0);**

**FillRect(147, 192, 48, 31, 0x07E0);**

**FillRect(196, 192, 48, 31, 0x07E0);**

**//Fila 5**

**FillRect(0, 160, 48, 31, 0x07E0);**

**FillRect(49, 160, 48, 31, 0x07E0);**

**FillRect(147, 160, 48, 31, 0x07E0);**

**FillRect(196, 160, 48, 31, 0x07E0);**

**//Fila 6**

**FillRect(0, 128, 48, 31, 0x0FFF);**

**FillRect(49, 128, 48, 31, 0x0FFF);**

**FillRect(147, 128, 48, 31, 0x0FFF);**

**FillRect(196, 128, 48, 31, 0x0FFF);**

**Serial.println("Intensidad: 0 dB");**

**delay(1000);**

**}**

**else if (dB > 45 && dB <= 50){**

**analogWrite(ledR, 255);**

**analogWrite(ledA, 0);**

**analogWrite(ledV, 255);**

**datodB = "Intensidad: 3dB";**

**//Fila 1**

**FillRect(0, 288, 48, 31, 0x07E0);**

**FillRect(49, 288, 48, 31, 0x07E0);**

**FillRect(147, 288, 48, 31, 0x07E0);**

**FillRect(196, 288, 48, 31, 0x07E0);**

**//Fila 2**

**FillRect(0, 256, 48, 31, 0x07E0);**

**FillRect(49, 256, 48, 31, 0x07E0);**

**FillRect(147, 256, 48, 31, 0x07E0);**

**FillRect(196, 256, 48, 31, 0x07E0);**

**//Fila 3**

**FillRect(0, 224, 48, 31, 0x07E0);**

**FillRect(49, 224, 48, 31, 0x07E0);**

**FillRect(147, 224, 48, 31, 0x07E0);**

**FillRect(196, 224, 48, 31, 0x07E0);**

**//Fila 4**

**FillRect(0, 192, 48, 31, 0x07E0);**

**FillRect(49, 192, 48, 31, 0x07E0);**

**FillRect(147, 192, 48, 31, 0x07E0);**

**FillRect(196, 192, 48, 31, 0x07E0);**

**//Fila 5**

**FillRect(0, 160, 48, 31, 0x07E0);**

**FillRect(49, 160, 48, 31, 0x07E0);**

**FillRect(147, 160, 48, 31, 0x07E0);**

**FillRect(196, 160, 48, 31, 0x07E0);**

**//Fila 6**

**FillRect(0, 128, 48, 31, 0x0FFF);**

**FillRect(49, 128, 48, 31, 0x0FFF);**

**FillRect(147, 128, 48, 31, 0x0FFF);**

**FillRect(196, 128, 48, 31, 0x0FFF);**

**//Fila 7**

**FillRect(0, 96, 48, 31, 0x0FFF);**

**FillRect(49, 96, 48, 31, 0x0FFF);**

**FillRect(147, 96, 48, 31, 0x0FFF);**

**FillRect(196, 96, 48, 31, 0x0FFF);**

**Serial.println("Intensidad: 3dB");**

**delay(1000);**

**}**

**else if (dB > 50 && dB <= 55){**

**analogWrite(ledR, 255);**

**analogWrite(ledA, 0);**

**analogWrite(ledV, 255);**

**datodB = "Intensidad: 6dB";**

**//Fila 1**

**FillRect(0, 288, 48, 31, 0x07E0);**

**FillRect(49, 288, 48, 31, 0x07E0);**

**FillRect(147, 288, 48, 31, 0x07E0);**

**FillRect(196, 288, 48, 31, 0x07E0);**

**//Fila 2**

**FillRect(0, 256, 48, 31, 0x07E0);**

**FillRect(49, 256, 48, 31, 0x07E0);**

**FillRect(147, 256, 48, 31, 0x07E0);**

**FillRect(196, 256, 48, 31, 0x07E0);**

**//Fila 3**

**FillRect(0, 224, 48, 31, 0x07E0);**

**FillRect(49, 224, 48, 31, 0x07E0);**

**FillRect(147, 224, 48, 31, 0x07E0);**

**FillRect(196, 224, 48, 31, 0x07E0);**

**//Fila 4**

**FillRect(0, 192, 48, 31, 0x07E0);**

**FillRect(49, 192, 48, 31, 0x07E0);**

**FillRect(147, 192, 48, 31, 0x07E0);**

**FillRect(196, 192, 48, 31, 0x07E0);**

**//Fila 5**

**FillRect(0, 160, 48, 31, 0x07E0);**

**FillRect(49, 160, 48, 31, 0x07E0);**

**FillRect(147, 160, 48, 31, 0x07E0);**

**FillRect(196, 160, 48, 31, 0x07E0);**

**//Fila 6**

**FillRect(0, 128, 48, 31, 0x0FFF);**

**FillRect(49, 128, 48, 31, 0x0FFF);**

**FillRect(147, 128, 48, 31, 0x0FFF);**

**FillRect(196, 128, 48, 31, 0x0FFF);**

**//Fila 7**

**FillRect(0, 96, 48, 31, 0x0FFF);**

**FillRect(49, 96, 48, 31, 0x0FFF);**

**FillRect(147, 96, 48, 31, 0x0FFF);**

**FillRect(196, 96, 48, 31, 0x0FFF);**

**//Fila 8**

**FillRect(0, 64, 48, 31, 0x0FFF);**

**FillRect(49, 64, 48, 31, 0x0FFF);**

**FillRect(147, 64, 48, 31, 0x0FFF);**

**FillRect(196, 64, 48, 31, 0x0FFF);**

**Serial.println("Intensidad: 6dB");**

**delay(1000);**

**}**

**else if (dB > 55 && dB <= 60){**

**analogWrite(ledR, 255);**

**analogWrite(ledA, 0);**

**analogWrite(ledV, 0);**

**datodB = "Intensidad: 10dB";**

**//Fila 1**

**FillRect(0, 288, 48, 31, 0x07E0);**

**FillRect(49, 288, 48, 31, 0x07E0);**

**FillRect(147, 288, 48, 31, 0x07E0);**

**FillRect(196, 288, 48, 31, 0x07E0);**

**//Fila 2**

**FillRect(0, 256, 48, 31, 0x07E0);**

**FillRect(49, 256, 48, 31, 0x07E0);**

**FillRect(147, 256, 48, 31, 0x07E0);**

**FillRect(196, 256, 48, 31, 0x07E0);**

**//Fila 3**

**FillRect(0, 224, 48, 31, 0x07E0);**

**FillRect(49, 224, 48, 31, 0x07E0);**

**FillRect(147, 224, 48, 31, 0x07E0);**

**FillRect(196, 224, 48, 31, 0x07E0);**

**//Fila 4**

**FillRect(0, 192, 48, 31, 0x07E0);**

**FillRect(49, 192, 48, 31, 0x07E0);**

**FillRect(147, 192, 48, 31, 0x07E0);**

**FillRect(196, 192, 48, 31, 0x07E0);**

**//Fila 5**

**FillRect(0, 160, 48, 31, 0x07E0);**

**FillRect(49, 160, 48, 31, 0x07E0);**

**FillRect(147, 160, 48, 31, 0x07E0);**

**FillRect(196, 160, 48, 31, 0x07E0);**

**//Fila 6**

**FillRect(0, 128, 48, 31, 0x0FFF);**

**FillRect(49, 128, 48, 31, 0x0FFF);**

**FillRect(147, 128, 48, 31, 0x0FFF);**

**FillRect(196, 128, 48, 31, 0x0FFF);**

**//Fila 7**

**FillRect(0, 96, 48, 31, 0x0FFF);**

**FillRect(49, 96, 48, 31, 0x0FFF);**

**FillRect(147, 96, 48, 31, 0x0FFF);**

**FillRect(196, 96, 48, 31, 0x0FFF);**

**//Fila 8**

**FillRect(0, 64, 48, 31, 0x0FFF);**

**FillRect(49, 64, 48, 31, 0x0FFF);**

**FillRect(147, 64, 48, 31, 0x0FFF);**

**FillRect(196, 64, 48, 31, 0x0FFF);**

**//Fila 9**

**FillRect(0, 32, 48, 31, 0x003F);**

**FillRect(49, 32, 48, 31, 0x003F);**

**FillRect(147, 32, 48, 31, 0x003F);**

**FillRect(196, 32, 48, 31, 0x003F);**

**Serial.println("Intensidad: 10dB");**

**delay(1000);**

**}**

**else if (dB >= 60){**

**analogWrite(ledR, 255);**

**analogWrite(ledA, 0);**

**analogWrite(ledV, 0);**

**datodB = "Intensidad: 16dB";**

**//Fila 1**

**FillRect(0, 288, 48, 31, 0x07E0);**

**FillRect(49, 288, 48, 31, 0x07E0);**

**FillRect(147, 288, 48, 31, 0x07E0);**

**FillRect(196, 288, 48, 31, 0x07E0);**

**//Fila 2**

**FillRect(0, 256, 48, 31, 0x07E0);**

**FillRect(49, 256, 48, 31, 0x07E0);**

**FillRect(147, 256, 48, 31, 0x07E0);**

**FillRect(196, 256, 48, 31, 0x07E0);**

**//Fila 3**

**FillRect(0, 224, 48, 31, 0x07E0);**

**FillRect(49, 224, 48, 31, 0x07E0);**

**FillRect(147, 224, 48, 31, 0x07E0);**

**FillRect(196, 224, 48, 31, 0x07E0);**

**//Fila 4**

**FillRect(0, 192, 48, 31, 0x07E0);**

**FillRect(49, 192, 48, 31, 0x07E0);**

**FillRect(147, 192, 48, 31, 0x07E0);**

**FillRect(196, 192, 48, 31, 0x07E0);**

**//Fila 5**

**FillRect(0, 160, 48, 31, 0x07E0);**

**FillRect(49, 160, 48, 31, 0x07E0);**

**FillRect(147, 160, 48, 31, 0x07E0);**

**FillRect(196, 160, 48, 31, 0x07E0);**

**//Fila 6**

**FillRect(0, 128, 48, 31, 0x0FFF);**

**FillRect(49, 128, 48, 31, 0x0FFF);**

**FillRect(147, 128, 48, 31, 0x0FFF);**

**FillRect(196, 128, 48, 31, 0x0FFF);**

**//Fila 7**

**FillRect(0, 96, 48, 31, 0x0FFF);**

**FillRect(49, 96, 48, 31, 0x0FFF);**

**FillRect(147, 96, 48, 31, 0x0FFF);**

**FillRect(196, 96, 48, 31, 0x0FFF);**

**//Fila 8**

**FillRect(0, 64, 48, 31, 0x0FFF);**

**FillRect(49, 64, 48, 31, 0x0FFF);**

**FillRect(147, 64, 48, 31, 0x0FFF);**

**FillRect(196, 64, 48, 31, 0x0FFF);**

**//Fila 9**

**FillRect(0, 32, 48, 31, 0x003F);**

**FillRect(49, 32, 48, 31, 0x003F);**

**FillRect(147, 32, 48, 31, 0x003F);**

**FillRect(196, 32, 48, 31, 0x003F);**

**//Fila 10**

**FillRect(0, 0, 48, 31, 0x003F);**

**FillRect(49, 0, 48, 31, 0x003F);**

**FillRect(147, 0, 48, 31, 0x003F);**

**FillRect(196, 0, 48, 31, 0x003F);**

**Serial.println("Intensidad: 16dB");**

**delay(1000);**

**}**

**}**

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

**// Función para inicializar LCD**

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

**void LCD\_Init(void) {**

**pinMode(LCD\_RST, OUTPUT);**

**pinMode(LCD\_CS, OUTPUT);**

**pinMode(LCD\_RS, OUTPUT);**

**pinMode(LCD\_WR, OUTPUT);**

**pinMode(LCD\_RD, OUTPUT);**

**for (uint8\_t i = 0; i < 8; i++) {**

**pinMode(DPINS[i], OUTPUT);**

**}**

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

**// Secuencia de Inicialización**

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

**digitalWrite(LCD\_CS, HIGH);**

**digitalWrite(LCD\_RS, HIGH);**

**digitalWrite(LCD\_WR, HIGH);**

**digitalWrite(LCD\_RD, HIGH);**

**digitalWrite(LCD\_RST, HIGH);**

**delay(5);**

**digitalWrite(LCD\_RST, LOW);**

**delay(20);**

**digitalWrite(LCD\_RST, HIGH);**

**delay(150);**

**digitalWrite(LCD\_CS, LOW);**

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

**LCD\_CMD(0xE9); // SETPANELRELATED**

**LCD\_DATA(0x20);**

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

**LCD\_CMD(0x11); // Exit Sleep SLEEP OUT (SLPOUT)**

**delay(100);**

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

**LCD\_CMD(0xD1); // (SETVCOM)**

**LCD\_DATA(0x00);**

**LCD\_DATA(0x71);**

**LCD\_DATA(0x19);**

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

**LCD\_CMD(0xD0); // (SETPOWER)**

**LCD\_DATA(0x07);**

**LCD\_DATA(0x01);**

**LCD\_DATA(0x08);**

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

**LCD\_CMD(0x36); //(0x36) // (MEMORYACCESS)**

**LCD\_DATA(0x96);//(0xC8) (0xE8)0x40 | 0x80 | 0x20 | 0x08); // LCD\_DATA(0x19);**

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

**LCD\_CMD(0x3A); // Set\_pixel\_format (PIXELFORMAT)**

**LCD\_DATA(0x05); // color setings, 05h - 16bit pixel, 11h - 3bit pixel**

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

**LCD\_CMD(0xC1); // (POWERCONTROL2)**

**LCD\_DATA(0x10);**

**LCD\_DATA(0x10);**

**LCD\_DATA(0x02);**

**LCD\_DATA(0x02);**

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

**LCD\_CMD(0xC0); // Set Default Gamma (POWERCONTROL1)**

**LCD\_DATA(0x00);**

**LCD\_DATA(0x35);**

**LCD\_DATA(0x00);**

**LCD\_DATA(0x00);**

**LCD\_DATA(0x01);**

**LCD\_DATA(0x02);**

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

**LCD\_CMD(0xC5); // Set Frame Rate (VCOMCONTROL1)**

**LCD\_DATA(0x04); // 72Hz**

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

**LCD\_CMD(0xD2); // Power Settings (SETPWRNORMAL)**

**LCD\_DATA(0x01);**

**LCD\_DATA(0x44);**

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

**LCD\_CMD(0xC8); //Set Gamma (GAMMASET)**

**LCD\_DATA(0x04);**

**LCD\_DATA(0x67);**

**LCD\_DATA(0x35);**

**LCD\_DATA(0x04);**

**LCD\_DATA(0x08);**

**LCD\_DATA(0x06);**

**LCD\_DATA(0x24);**

**LCD\_DATA(0x01);**

**LCD\_DATA(0x37);**

**LCD\_DATA(0x40);**

**LCD\_DATA(0x03);**

**LCD\_DATA(0x10);**

**LCD\_DATA(0x08);**

**LCD\_DATA(0x80);**

**LCD\_DATA(0x00);**

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

**LCD\_CMD(0x2A); // Set\_column\_address 320px (CASET)**

**LCD\_DATA(0x00);**

**LCD\_DATA(0x00);**

**LCD\_DATA(0x01);**

**LCD\_DATA(0x3F);**

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

**LCD\_CMD(0x2B); // Set\_page\_address 480px (PASET)**

**LCD\_DATA(0x00);**

**LCD\_DATA(0x00);**

**LCD\_DATA(0x01);**

**LCD\_DATA(0xE0);**

**// LCD\_DATA(0x8F);**

**LCD\_CMD(0x29); //display on**

**LCD\_CMD(0x2C); //display on**

**LCD\_CMD(ILI9341\_INVOFF); //Invert Off**

**delay(120);**

**LCD\_CMD(ILI9341\_SLPOUT); //Exit Sleep**

**delay(120);**

**LCD\_CMD(ILI9341\_DISPON); //Display on**

**digitalWrite(LCD\_CS, HIGH);**

**}**

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

**// Función para enviar comandos a la LCD - parámetro (comando)**

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

**void LCD\_CMD(uint8\_t cmd) {**

**digitalWrite(LCD\_RS, LOW);**

**digitalWrite(LCD\_WR, LOW);**

**GPIO\_PORTB\_DATA\_R = cmd;**

**digitalWrite(LCD\_WR, HIGH);**

**}**

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

**// Función para enviar datos a la LCD - parámetro (dato)**

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

**void LCD\_DATA(uint8\_t data) {**

**digitalWrite(LCD\_RS, HIGH);**

**digitalWrite(LCD\_WR, LOW);**

**GPIO\_PORTB\_DATA\_R = data;**

**digitalWrite(LCD\_WR, HIGH);**

**}**

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

**// Función para definir rango de direcciones de memoria con las cuales se trabajara (se define una ventana)**

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

**void SetWindows(unsigned int x1, unsigned int y1, unsigned int x2, unsigned int y2) {**

**LCD\_CMD(0x2a); // Set\_column\_address 4 parameters**

**LCD\_DATA(x1 >> 8);**

**LCD\_DATA(x1);**

**LCD\_DATA(x2 >> 8);**

**LCD\_DATA(x2);**

**LCD\_CMD(0x2b); // Set\_page\_address 4 parameters**

**LCD\_DATA(y1 >> 8);**

**LCD\_DATA(y1);**

**LCD\_DATA(y2 >> 8);**

**LCD\_DATA(y2);**

**LCD\_CMD(0x2c); // Write\_memory\_start**

**}**

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

**// Función para borrar la pantalla - parámetros (color)**

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

**void LCD\_Clear(unsigned int c) {**

**unsigned int x, y;**

**LCD\_CMD(0x02c); // write\_memory\_start**

**digitalWrite(LCD\_RS, HIGH);**

**digitalWrite(LCD\_CS, LOW);**

**SetWindows(0, 0, 319, 239); // 479, 319);**

**for (x = 0; x < 320; x++)**

**for (y = 0; y < 240; y++) {**

**LCD\_DATA(c >> 8);**

**LCD\_DATA(c);**

**}**

**digitalWrite(LCD\_CS, HIGH);**

**}**

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

**// Función para dibujar una línea horizontal - parámetros ( coordenada x, cordenada y, longitud, color)**

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

**void H\_line(unsigned int x, unsigned int y, unsigned int l, unsigned int c) {**

**unsigned int i, j;**

**LCD\_CMD(0x02c); //write\_memory\_start**

**digitalWrite(LCD\_RS, HIGH);**

**digitalWrite(LCD\_CS, LOW);**

**l = l + x;**

**SetWindows(x, y, l, y);**

**j = l;// \* 2;**

**for (i = 0; i < l; i++) {**

**LCD\_DATA(c >> 8);**

**LCD\_DATA(c);**

**}**

**digitalWrite(LCD\_CS, HIGH);**

**}**

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

**// Función para dibujar una línea vertical - parámetros ( coordenada x, cordenada y, longitud, color)**

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

**void V\_line(unsigned int x, unsigned int y, unsigned int l, unsigned int c) {**

**unsigned int i, j;**

**LCD\_CMD(0x02c); //write\_memory\_start**

**digitalWrite(LCD\_RS, HIGH);**

**digitalWrite(LCD\_CS, LOW);**

**l = l + y;**

**SetWindows(x, y, x, l);**

**j = l; //\* 2;**

**for (i = 1; i <= j; i++) {**

**LCD\_DATA(c >> 8);**

**LCD\_DATA(c);**

**}**

**digitalWrite(LCD\_CS, HIGH);**

**}**

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

**// Función para dibujar un rectángulo - parámetros ( coordenada x, cordenada y, ancho, alto, color)**

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

**void Rect(unsigned int x, unsigned int y, unsigned int w, unsigned int h, unsigned int c) {**

**H\_line(x , y , w, c);**

**H\_line(x , y + h, w, c);**

**V\_line(x , y , h, c);**

**V\_line(x + w, y , h, c);**

**}**

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

**// Función para dibujar un rectángulo relleno - parámetros ( coordenada x, cordenada y, ancho, alto, color)**

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

**/\*void FillRect(unsigned int x, unsigned int y, unsigned int w, unsigned int h, unsigned int c) {**

**unsigned int i;**

**for (i = 0; i < h; i++) {**

**H\_line(x , y , w, c);**

**H\_line(x , y+i, w, c);**

**}**

**}**

**\*/**

**void FillRect(unsigned int x, unsigned int y, unsigned int w, unsigned int h, unsigned int c) {**

**LCD\_CMD(0x02c); // write\_memory\_start**

**digitalWrite(LCD\_RS, HIGH);**

**digitalWrite(LCD\_CS, LOW);**

**unsigned int x2, y2;**

**x2 = x + w;**

**y2 = y + h;**

**SetWindows(x, y, x2 - 1, y2 - 1);**

**unsigned int k = w \* h \* 2 - 1;**

**unsigned int i, j;**

**for (int i = 0; i < w; i++) {**

**for (int j = 0; j < h; j++) {**

**LCD\_DATA(c >> 8);**

**LCD\_DATA(c);**

**//LCD\_DATA(bitmap[k]);**

**k = k - 2;**

**}**

**}**

**digitalWrite(LCD\_CS, HIGH);**

**}**

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

**// Función para dibujar texto - parámetros ( texto, coordenada x, cordenada y, color, background)**

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

**void LCD\_Print(String text, int x, int y, int fontSize, int color, int background) {**

**int fontXSize ;**

**int fontYSize ;**

**if (fontSize == 1) {**

**fontXSize = fontXSizeSmal ;**

**fontYSize = fontYSizeSmal ;**

**}**

**if (fontSize == 2) {**

**fontXSize = fontXSizeBig ;**

**fontYSize = fontYSizeBig ;**

**}**

**char charInput ;**

**int cLength = text.length();**

**Serial.println(cLength, DEC);**

**int charDec ;**

**int c ;**

**int charHex ;**

**char char\_array[cLength + 1];**

**text.toCharArray(char\_array, cLength + 1) ;**

**for (int i = 0; i < cLength ; i++) {**

**charInput = char\_array[i];**

**Serial.println(char\_array[i]);**

**charDec = int(charInput);**

**digitalWrite(LCD\_CS, LOW);**

**SetWindows(x + (i \* fontXSize), y, x + (i \* fontXSize) + fontXSize - 1, y + fontYSize );**

**long charHex1 ;**

**for ( int n = 0 ; n < fontYSize ; n++ ) {**

**if (fontSize == 1) {**

**charHex1 = pgm\_read\_word\_near(smallFont + ((charDec - 32) \* fontYSize) + n);**

**}**

**if (fontSize == 2) {**

**charHex1 = pgm\_read\_word\_near(bigFont + ((charDec - 32) \* fontYSize) + n);**

**}**

**for (int t = 1; t < fontXSize + 1 ; t++) {**

**if (( charHex1 & (1 << (fontXSize - t))) > 0 ) {**

**c = color ;**

**} else {**

**c = background ;**

**}**

**LCD\_DATA(c >> 8);**

**LCD\_DATA(c);**

**}**

**}**

**digitalWrite(LCD\_CS, HIGH);**

**}**

**}**

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

**// Función para dibujar una imagen a partir de un arreglo de colores (Bitmap) Formato (Color 16bit R 5bits G 6bits B 5bits)**

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

**void LCD\_Bitmap(unsigned int x, unsigned int y, unsigned int width, unsigned int height, unsigned char bitmap[]) {**

**LCD\_CMD(0x02c); // write\_memory\_start**

**digitalWrite(LCD\_RS, HIGH);**

**digitalWrite(LCD\_CS, LOW);**

**unsigned int x2, y2;**

**x2 = x + width;**

**y2 = y + height;**

**SetWindows(x, y, x2 - 1, y2 - 1);**

**unsigned int k = 0;**

**unsigned int i, j;**

**for (int i = 0; i < width; i++) {**

**for (int j = 0; j < height; j++) {**

**LCD\_DATA(bitmap[k]);**

**LCD\_DATA(bitmap[k + 1]);**

**//LCD\_DATA(bitmap[k]);**

**k = k + 2;**

**}**

**}**

**digitalWrite(LCD\_CS, HIGH);**

**}**

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

**// Función para dibujar una imagen sprite - los parámetros columns = número de imagenes en el sprite, index = cual desplegar, flip = darle vuelta**

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

**void LCD\_Sprite(int x, int y, int width, int height, unsigned char bitmap[], int columns, int index, char flip, char offset) {**

**LCD\_CMD(0x02c); // write\_memory\_start**

**digitalWrite(LCD\_RS, HIGH);**

**digitalWrite(LCD\_CS, LOW);**

**unsigned int x2, y2;**

**x2 = x + width;**

**y2 = y + height;**

**SetWindows(x, y, x2 - 1, y2 - 1);**

**int k = 0;**

**int ancho = ((width \* columns));**

**if (flip) {**

**for (int j = 0; j < height; j++) {**

**k = (j \* (ancho) + index \* width - 1 - offset) \* 2;**

**k = k + width \* 2;**

**for (int i = 0; i < width; i++) {**

**LCD\_DATA(bitmap[k]);**

**LCD\_DATA(bitmap[k + 1]);**

**k = k - 2;**

**}**

**}**

**} else {**

**for (int j = 0; j < height; j++) {**

**k = (j \* (ancho) + index \* width + 1 + offset) \* 2;**

**for (int i = 0; i < width; i++) {**

**LCD\_DATA(bitmap[k]);**

**LCD\_DATA(bitmap[k + 1]);**

**k = k + 2;**

**}**

**}**

**}**

**digitalWrite(LCD\_CS, HIGH);**

**}**

**Link a GitHub:** [**https://github.com/Jeremy-Mejia/Proyecto-2-Digital.git**](https://github.com/Jeremy-Mejia/Proyecto-2-Digital.git)

**Link del Video:** [**https://youtu.be/2UroSGFwkIQ**](https://youtu.be/2UroSGFwkIQ)