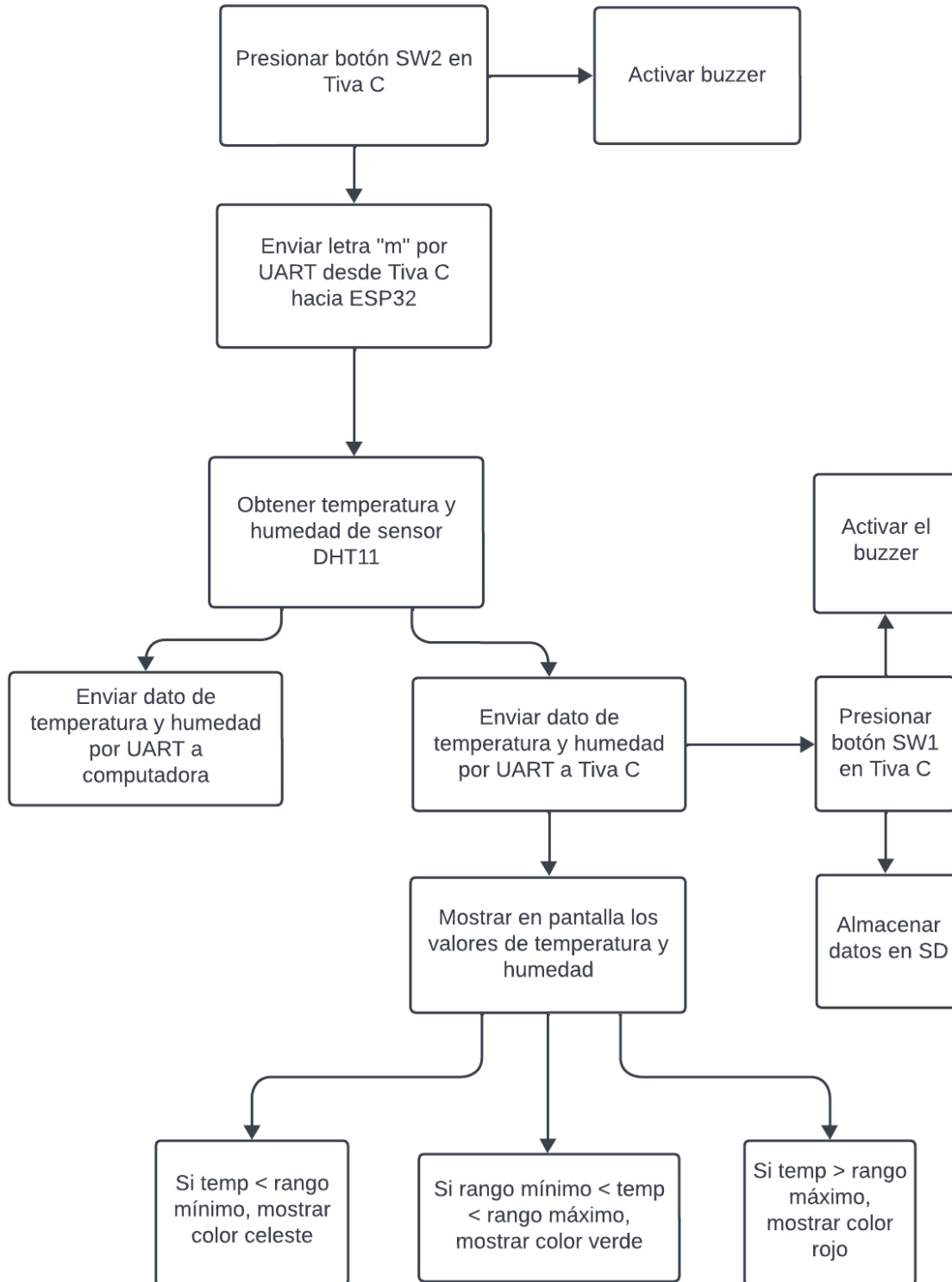


Proyecto No 2. SPI, TFT y SD

Diagrama de flujo



Pseudocódigos

ESP32

1. Comunicar el ESP32 con el sensor de temperatura y humedad DHT11 por medio de una lectura digital.
2. Establecer comunicación UART entre ESP32 y TIVA C y entre ESP32 y computadora.
3. Recibir un indicador desde la TIVA C cuando se presione un botón para poder obtener un valor de temperatura y de humedad.
4. Enviar los datos tanto a la TIVA C como a la computadora

TIVA C

1. Establecer comunicación UART con el ESP32 y comunicación SPI con la SD. Además, conectar pantalla TFT y buzzer y comunicar dos botones con los cuales se pueda dar indicaciones al microcontrolador.
2. Presionar un botón para obtener la lectura de temperatura y humedad desde el ESP32 y determinar en qué rango de temperatura se encuentra. Definir 3 rangos de temperatura a los que se les asignará un color, celeste, verde y rojo. A la vez, generar un sonido por medio del buzzer.
3. Mostrar los valores obtenidos en pantalla y dependiendo del rango de temperatura en que se encuentra, mostrar un color respectivo en la pantalla TFT.
4. Presionar un segundo botón para poder almacenar en la SD el valor de temperatura y humedad obtenidos. A la vez, generar un sonido por medio del buzzer (los sonidos de ambos botones deberán ser diferentes).

Funciones y lógica del código

Código ESP32: se inicia con adjuntar las librerías necesarias, entre estas la de DHT para así poder usar las funciones del sensor digital. Luego es necesario inicializar los Serial que usaremos y el sensor. A continuación, se obtienen los valores de temperatura y humedad, y al recibir una letra específica desde la Tiva por medio de comunicación serial, se mostrará en la computadora ambos valores y además estos se convertirán a String y se concatenarán para poder enviar el dato por UART a la Tiva C.

```

#include <Arduino.h>
#include "DHT.h"
#define DHTPIN 4
#define DHTTYPE DHT11

//Variables globales
int valorSensor = 0;

DHT dht(DHTPIN, DHTTYPE);

//Configuración
void setup()
{
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));
  Serial2.begin(115200);

  dht.begin();
}

```

```

float h = dht.readHumidity();
float t = dht.readTemperature();
//Si no se pudo obtener los valores, mostrar un mensaje de advertencia
if (isnan(h) || isnan(t))
{
  Serial.println(F("Failed to read from DHT sensor!"));
  return;
}

//Cuando haya comunicaci3n serial
if (Serial2.available() > 0)
{
  //leer el mensaje proveniente de la Tiva C
  char bufferTiva = Serial2.read();
  //Si este corresponde a la letra m, enviar los datos de temperatura y humedad
  //concatenados en un string separado por una coma
  if (bufferTiva == 'm')
  {
    Serial.println("\nEnviando temperatura y humedad");
    Serial.print(F("Temperatura: "));
    Serial.print(t);
    Serial.println(F("°C "));
    Serial.print(F("Humedad: "));
    Serial.print(h);
    Serial.print(F("%"));
    String temp = String(t);
    String hum = String(h);
    String myString = temp + ',' + hum;
    Serial2.println(myString);
    delay(300);
  }
}

```

C3digo Tiva C: En este caso se agregaron todas las librerías para poder utilizar la pantalla LCD con las funciones que se nos proporcionaron en clase (no ser3n agregadas en imagen ya que es bastante extenso, cualquier duda consultar c3digo). Adem3s, se definieron las notas que se usarían para las melodías del buzzer y el archivo que se abriría para almacenar los datos en la SD.

Para poder evitar que la memoria de la pantalla se llene, se utilizó la función PROGMEM para almacenar los gráficos que se deseaban mostrar y se agregaron como variables externas.

```
extern uint8_t humedad[];
extern uint8_t termometro[];
```

En la configuración inicial se definieron todos los aspectos de la pantalla que estarían desde el inicio, estos son los textos, las dos imágenes (termómetro y nube) y un recuadro el cual cambiaría de color para determinar en qué rango se encuentra la temperatura.

```
FillRect(35, 70, 150, 20, 0xF47C);
String text1 = "Temperatura";
String text3 = "Humedad";
LCD_Print(text1, 5, 45, 2, 0x00,0xffff);
LCD_Print(text3, 135, 160, 2, 0x00,0xffff);
//LCD_Sprite(int x, int y, int width, int height,

//LCD_Bitmap(unsigned int x, unsigned int y, unsigned int width, unsigned int height, unsigned char *bitmap)
LCD_Bitmap(200, 0, 120, 120, termometro);
LCD_Bitmap(0,120,120,120, humedad);
```

Luego en el menú principal se obtienen los valores de los botones de la Tiva y se utiliza una segunda variable para cada uno para así evitar el rebote de estos.

```
// read from port 1, send to port 0.
buttonState2 = digitalRead(buttonPin2);
buttonState1 = digitalRead(buttonPin1);

//Presionar boton para recibir datos
if (buttonState2 == 0) {
  presionado2 = 1;
}
if(buttonState2 == 1 && presionado2 == 1){
  Serial.println("recibiendo");
}
```

Si se ha presionado el botón SW2, se envía la letra "m" al ESP32 y se reproduce una melodía en el buzzer.

```

if(buttonState2 == 1 && presionado2 == 1){
    Serial.println("recibiendo");
    Serial2.println("m");
    delay(300);
    presionado2 = 0;
    for (int thisNote = 0; thisNote < 4; thisNote++) {

        // to calculate the note duration, take one second divided by the
        // note type. 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); // stop the tone playing
    }

    //Leer valores de temperatura y humedad
    while (Serial2.available()){
        delay(10);
        if (Serial2.available()>0){
            char c = Serial2.read();
            inByte +=c;
        }
    }
    if(inByte.length()>0){
        Serial.println(inByte);
        temp = inByte.substring(0,5);
        hum = inByte.substring(6,11);
        enviarArchivo = inByte;
        inByte = "";
        temp = temp+"*C";
        hum = hum+"%";
        tempFloat = temp.toFloat();
        Serial.println("Temperatura en float: ");
        Serial.println(tempFloat);
        if (tempFloat > 26.50){
            FillRect(35, 70, 150, 20, 0xF924);
        }
        else if (tempFloat<26.50 && tempFloat > 25.80){
            FillRect(35, 70, 150, 20, 0x5FAB);
        }
        else if (tempFloat<25.80){
            FillRect(35, 70, 150, 20, 0x57BC);
        }
        Serial.println(temp);
        LCD_Print(temp, 80, 75, 1.99, 0x00,0xffff);
        Serial.println(hum);
        LCD_Print(hum, 155, 190, 1.99, 0x00,0xffff);
        delay(300);
    }
}

```

Si el serial se encuentra disponible, se lee el valor de temperatura y humedad enviados desde el ESP32, estos se almacenan como un String (ya que vienen unidos) y luego se separan para tener cada valor individualmente. El valor de temperatura se castea a float para poder verificar en qué rango de temperatura se encuentra y luego se muestran ambos valores en la pantalla y, además, un recuadro alrededor del valor de la temperatura de un color correspondiente al rango.

Si se presiona el botón SW1, se abre el documento en la SD con el nombre "proyecto.txt" para escritura y se almacena los últimos valores de temperatura y humedad obtenidos, además, se reproduce una melodía en el buzzer.

```
//Presionar boton para almacenar en la SD
if (buttonState1 == 0) {
  presionado1 = 1;
}
if(buttonState1 == 1 && presionado1 == 1){
  archivo = SD.open("proyecto.txt", FILE_WRITE);
  if(archivo){
    for (int thisNote = 0; thisNote < 4; thisNote++) {

      // to calculate the note duration, take one second divided by the
      // note type. 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); // stop the tone playing

    }
    Serial.println("Subiendo a SD: ");
    Serial.println(enviarArchivo);
    archivo.println(enviarArchivo);
    archivo.close();
    Serial.println("Listo!");
  }
  else{
    Serial.println("error opening proyecto.txt");
  }
  presionado1 = 0;
}
```

Link de Github

<https://github.com/ingebor/Proyecto2Digital2.git>

Link de Youtube

<https://www.youtube.com/watch?v=1VOH8OTbX8I>