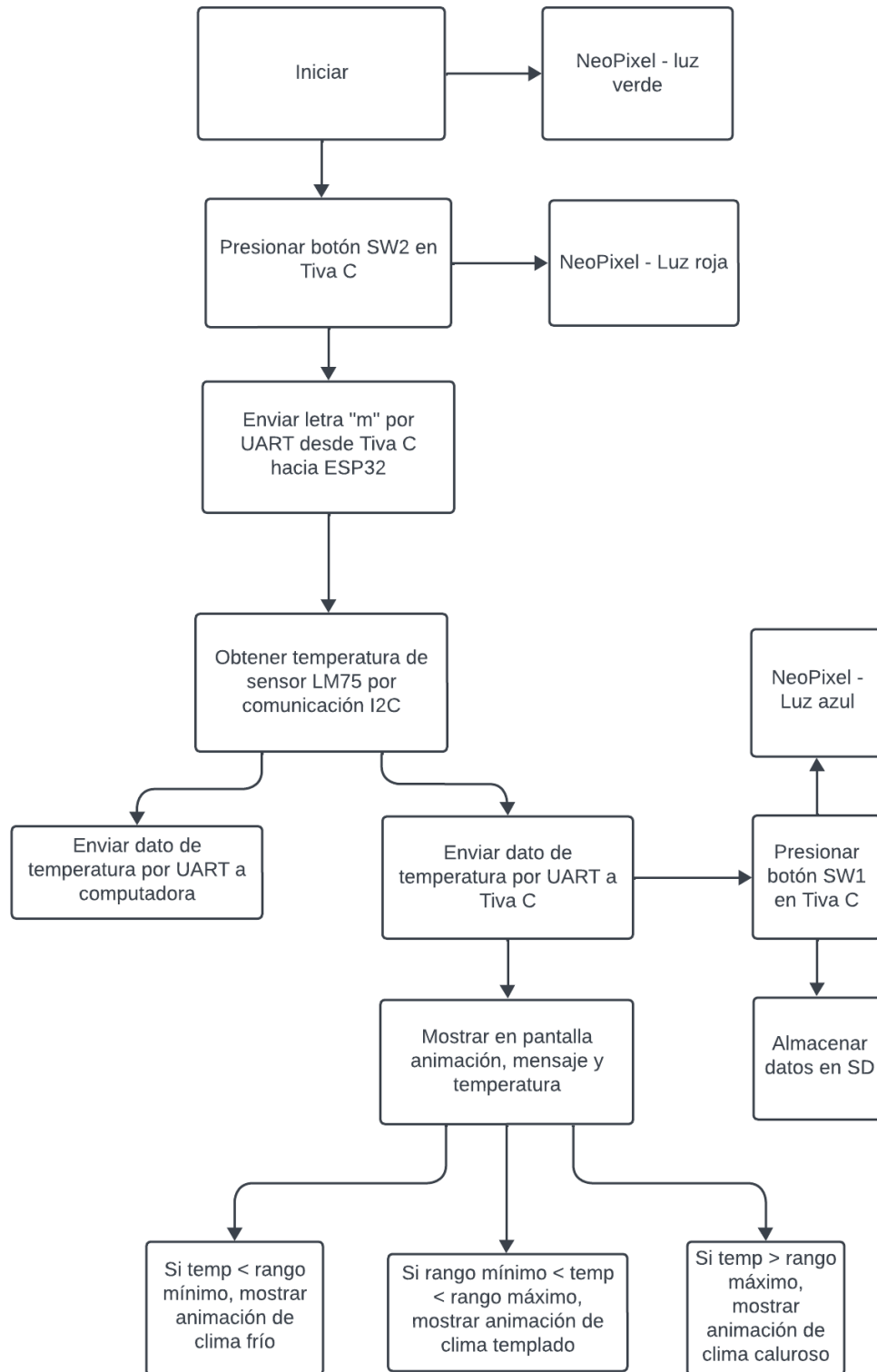


## Proyecto No 3. I2C y NeoPixel

### Diagrama de flujo



## **Pseudocódigos**

### **ESP32**

1. Comunicar el ESP32 con el sensor de temperatura LM75 por comunicación I2C.
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.
4. Enviar los datos tanto a la TIVA C como a la computadora por UART.
5. Encender un color correspondiente al envío de datos

### **TIVA C**

1. Establecer comunicación UART con el ESP32 y comunicación SPI con la SD. Además, conectar pantalla TFT y comunicar dos botones con los cuales se pueda dar indicaciones al microcontrolador.
2. Presionar un botón para obtener la lectura de temperatura desde el ESP32 y determinar en qué rango de temperatura se encuentra. Definir 3 rangos de temperatura a los que se les asignará una animación.
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 obtenido. A la vez, enviar un indicador al ESP32 para que pueda encender los leds con un color correspondiente al almacenamiento de los datos

## **Funciones y lógica del código**

Código ESP32: se inicia con adjuntar las librerías necesarias, entre estas la de LM75 para así poder usar las funciones del sensor y la de NeoPixel para poder utilizar los leds. Se crea el objeto del NeoPixel a utilizar y las variables necesarias para controlar los colores a usar. Luego es necesario inicializar los Serial que usaremos y el sensor. A continuación, al iniciar se encienden los leds en un patrón específico y se obtienen los valores de temperatura, y al recibir una letra específica desde la Tiva por medio de comunicación serial, se mostrará otro patrón de colores en los leds y se mostrará en la computadora dicho valor y además se enviará el dato por UART a la Tiva C.

En el caso que se reciba el indicador de que se guardará un dato, se mostrará un patrón diferente de colores en los leds.

```
// Librerías
#include <Arduino.h>
#include <Temperature_LM75_Derived.h>
#include <Adafruit_NeoPixel.h>

#define CIRCLE 25
#define CIRCLE_LEDS 24
#define BRIGHTNESS 255

// Declare objects to control the NeoPixels
Adafruit_NeoPixel circle(CIRCLE_LEDS, CIRCLE, NEO_GRB + NEO_KHZ800);

// Variables globales
int valorSensor = 0;
uint8_t red = 0;
uint8_t green = 0;
uint8_t blue = 0;

Generic_LM75 temperature;
```

```
// Configuración
void setup()
{
  Serial.begin(115200);
  Serial2.begin(115200);
  Wire.begin();
  // Ensure all LED arrays are clear when we start
  circle.begin();
  circle.clear();
  circle.setBrightness(BRIGHTNESS);

  Serial.println("Inicio exitoso!");
  red = 0;
  green = 255;
  blue = 0;
  circle.clear();

  for (int i = CIRCLE_LEDS - 1; i >= 0; i--)
  {
    circle.setPixelColor(i, red, green, blue);
    circle.show();
    delay(50);
  }
}
```

```

// Loop principal
void loop()
{
    // Obtener los valores de temperatura
    float t = temperature.readTemperatureC();
    // Si no se pudo obtener los valores, mostrar un mensaje de advertencia
    if (isnan(t))
    {
        Serial.println(F("No se pudo leer del sensor LM75!"));
        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
        if (bufferTiva == 'm')
        {
            red = 255;
            green = 0;
            blue = 0;
            circle.clear();

```

```

//Iluminar en circulo
for (int i = 0; i < CIRCLE_LEDS; i++)
{
    //Mostrar colores correspondientes en neopixel
    circle.setPixelColor(i, red, green, blue);
    circle.show();
    delay(40);
    blue = blue + 12;
    red = red - 12;
}
delay(1000);
circle.clear();
circle.show();
Serial.println("\nEnviando temperatura");
Serial.print(F("Temperatura: "));
Serial.print(t);
Serial.println(F("°C "));
String temp = String(t);
Serial2.println(temp);
delay(300);
}

```

```

//Si guardan los datos en la SD, se recibe el valor x
//en ese caso, iluminar del color correspondiente
if (bufferTiva == 'x')
{
    red = 0;
    green = 0;
    blue = 255;
    circle.clear();
    //Iluminar en circulo
    for (int i = 0; i < CIRCLE_LEDS; i++)
    {
        circle.setPixelColor(i, red, green, blue);
        circle.show();
        delay(30);
        green += 8;
    }
    delay(1000);
    // Clear down the circle LEDs and start again
    circle.clear();
    circle.show();
}
}

```

Código 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 serán agregadas en imagen ya que es bastante extenso, cualquier duda consultar código). Además, se definió 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 tile[];
extern uint8_t bombette[];
extern uint8_t sol[];
extern uint8_t nube[];
extern uint8_t kooper[];

```

En la configuración inicial se definieron todos los aspectos de la pantalla que estarían desde el inicio, estos son los textos, el color de fondo y la animación del suelo.

```

//Mostrar mensajes e imágenes iniciales en la LCD
String text1 = "Bienvenido";
String text2 = "Mide la temperatura por favor!";
LCD_Print(text1, 5, 45, 2, 0x00,0x3EDE);
LCD_Print(text2, 5, 85, 1, 0x00,0x3EDE);

//LCD_Bitmap(unsigned int x, unsigned int y, unsigned
for(int x = 0; x <319; x++){
    LCD_Bitmap(x, 207, 16, 16, tile);
    LCD_Bitmap(x, 223, 16, 16, tile);
    x += 15;
}

```

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. Si se ha presionado el botón SW2, se envía la letra "m" al ESP32.

```

// Obtener valores de los botones
buttonState2 = digitalRead(buttonPin2);
buttonState1 = digitalRead(buttonPin1);

//Presionar boton para recibir datos
if (buttonState2 == 0) {
    presionado2 = 1;
}
if(buttonState2 == 1 && presionado2 == 1){
    Serial.println("recibiendo");
    Serial2.println("m");
    delay(300);
    presionado2 = 0;
}

```

Si el serial se encuentra disponible, se lee el valor de temperatura enviados desde el ESP32, este se almacena como un String. El valor de temperatura se castea a float para poder verificar en qué rango de temperatura se encuentra y luego se muestra la animación correspondiente al rango de temperatura, un mensaje y el valor.

```

//Leer valores de temperatura
while (Serial2.available()){
  delay(10);
  if (Serial2.available()>0){
    char c = Serial2.read();
    inByte +=c;
  }
}
if(inByte.length()>0){
  Serial.println(inByte);
  //Separar valores
  temp = inByte.substring(0,5);
  //Variable en la que se almacenarán los datos para poder enviar a la SD
  enviarArchivo = inByte;
  //Reiniciar variable para volver a recibir datos

  //Preparar variables para mostrarlas en pantalla
  inByte = "";
  temp = temp+"*C";
  tempFloat = temp.toFloat();
  Serial.println("Temperatura en float: ");
  Serial.println(tempFloat);
  String msjCalor = "Que calor! Corre del sol";
  String msjTemplado = "Esta templado! Vamos a caminar";
  String msjFrio = "Que frio! Mira esa nube!";

  //Calor
  if (tempFloat > 27.90){
    //Limpiar pantalla
    LCD_Clear(0x3EDE);
    //Mostrar cuadros
    for(int x = 0; x <319; x++){
      LCD_Bitmap(x, 207, 16, 16, tile);
      LCD_Bitmap(x, 223, 16, 16, tile);
      x += 15;
    }
    delay(500);
    //Mostrar animacion
    for (int x = 0; x<320-40; x++){
      delay(15);
      int anim2 = (x/10)%3; //3 porque son 3 columnas
      int anim3 = (x/11)%1;
      LCD_Sprite(x,160,40,40,bombette,3,anim2,0,1);
      V_line(x-1, 160,40,0x3EDE);
      LCD_Sprite(x-40,160,40,40,sol,1,anim3,0,1);
      V_line(x-41, 160,40,0x3EDE);
    }
    delay(500);
    LCD_Print(msjCalor, 55, 55, 1.99, 0x00,0x3EDE);
  }
}

```

```

//templado
else if (tempFloat<27.90 && tempFloat > 26.90){
    //Limpiar pantalla
    LCD_Clear(0x3EDE);
    for(int x = 0; x <319; x++){
        LCD_Bitmap(x, 207, 16, 16, tile);
        LCD_Bitmap(x, 223, 16, 16, tile);
        x += 15;
    }
    delay(500);
    //Mostrar animacion
    for (int x = 0; x<320-40; x++){
        delay(15);
        int anim2 = (x/10)%3;
        int anim3 = (x/11)%4; //4 porque son 4 columnas
        LCD_Sprite(x,160,40,40,bombette,3,anim2,0,1);
        V_line(x-1, 160,40,0x3EDE); //Agregamos una línea detrás de bombette
        LCD_Sprite(x-40,160,40,40,kooper,4,anim3,0,1);
        V_line(x-40, 160,40,0x3EDE);
    }
    delay(500);
    LCD_Print(msjTemplado, 55, 55, 1.99, 0x00,0x3EDE);
}

//frio
else if (tempFloat<26.90){
    //Limpiar pantalla
    LCD_Clear(0x3EDE);
    for(int x = 0; x <319; x++){
        LCD_Bitmap(x, 207, 16, 16, tile);
        LCD_Bitmap(x, 223, 16, 16, tile);
        x += 15;
    }
    delay(500);
    //Mostrar animacion
    for (int x = 0; x<320-40; x++){
        delay(15);
        int anim2 = (x/10)%3;
        int anim3 = (x/11)%1;
        LCD_Sprite(x,160,40,40,bombette,3,anim2,0,1);
        V_line(x-1, 160,40,0x3EDE);
        LCD_Sprite(x-40,160,40,40,nube,1,anim3,0,1);
        V_line(x-40, 160,40,0x3EDE);
    }
    delay(500);
    LCD_Print(msjFrio, 65, 55, 1.99, 0x00,0x3EDE);
}

//Mostrar valor de temperatura
Serial.println(temp);
LCD_Print(temp, 80, 75, 1.99, 0x00,0x3EDE);
delay(300);

```



Si se presiona el botón SW1, se abre el documento en la SD con el nombre "proyecto.txt" para escritura y se almacena el último valor de temperatura obtenido, además, se envía la letra "x" al ESP32 para así encender los leds según el color correspondiente

```
//Presionar boton para almacenar en la SD
if (buttonStatel == 0) {
    presionadol = 1;
}
if(buttonStatel == 1 && presionadol == 1){
    Serial.println("Guardando");
    Serial2.println("x");
    delay(300);
    archivo = SD.open("proyecto.txt", FILE_WRITE);
    if(archivo){
        //Subir datos a SD
        Serial.println("Subiendo a SD: ");
        Serial.println(enviarArchivo);
        archivo.println(enviarArchivo);
        archivo.close();
        Serial.println("Listo!");
    }
    else{
        Serial.println("error opening proyecto.txt");
    }
    presionadol = 0;
}
```

### Link de Github

<https://github.com/ingebor/Proyecto3Digital2.git>

### Link de Youtube

<https://youtu.be/FHRPiwmnaD8>