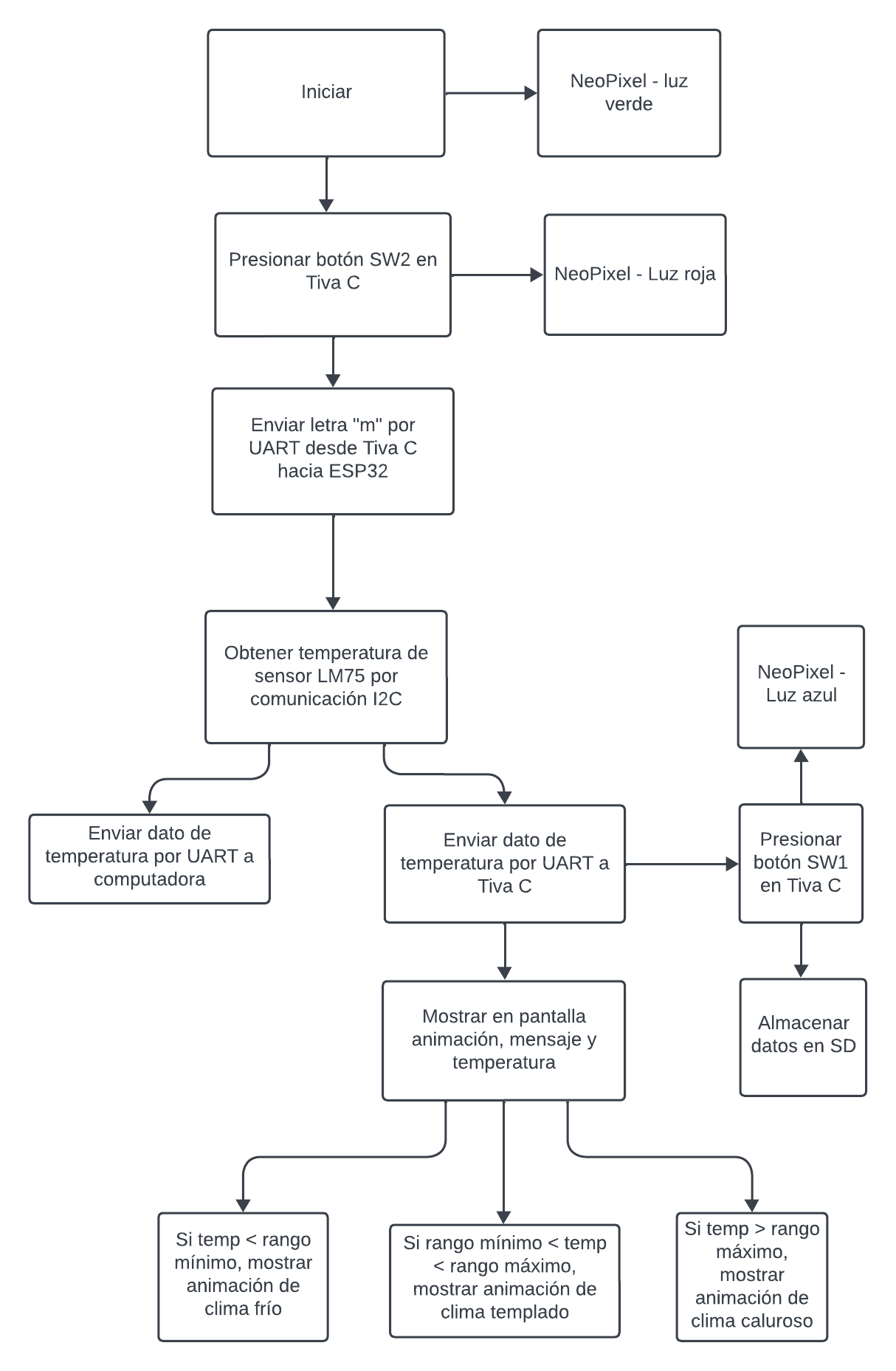
**Proyecto No 3. I2C y NeoPixel**

**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.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

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.

Texto, Carta

Descripción generada automáticamente

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.

Texto, Carta

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

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

Texto

Descripción generada automáticamente

**Link de Github**

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

**Link de Youtube**