

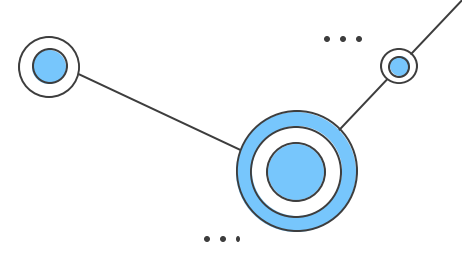
# TC74 Thermal Sensor

## Arquiteturas para Sistemas Embutidos

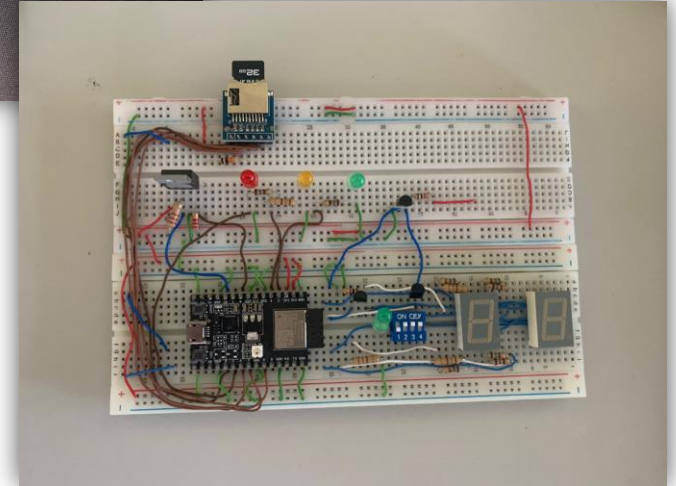
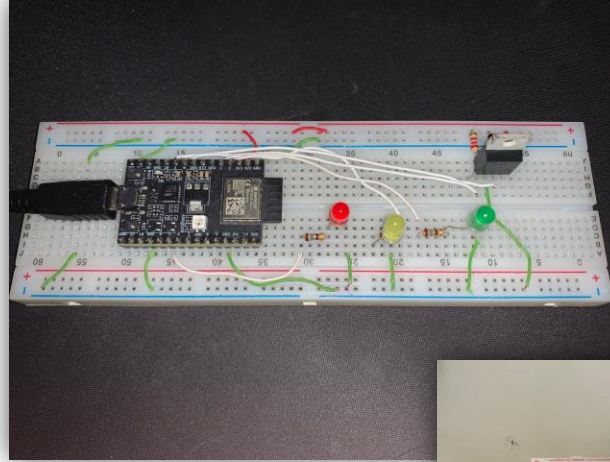
Adalberto Jr.- 105589

Nuno Ferreira- 121758

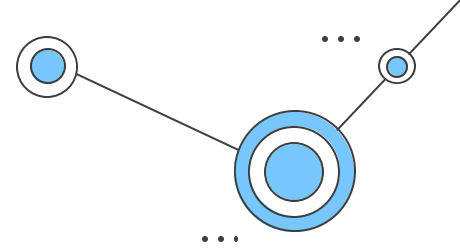
# Components



- ESP32
- Sensor de temperatura TC74
- 3 LEDs
- SD card



# TC74 Sensor



- O TC74 é um sensor que comunica com outros dispositivos por meio do protocolo de comunicação I2C.
- O sensor TC74 pode medir temperaturas na faixa dos  $-40^{\circ}\text{C}$  a  $+125^{\circ}\text{C}$ , tornando-o adequado para uma variedade de aplicações.
- Este sensor é capaz de medir a temperatura com alta precisão, geralmente com uma resolução de  $0,5^{\circ}\text{C}$ .



# Project Objectives

- **Ler a temperatura:** utilizar o sensor de temperatura TC74 para medir com precisão a temperatura ambiente.
- **Acender os LEDs com base na temperatura:** À medida que a temperatura aumenta, os LEDs com cores mais quentes vão-se acendendo, proporcionando uma representação visual da temperatura do sensor.
- **Dashboard:** Criar um dashboard que mostre os valores da temperatura utilizando uma web-based API.
- **Sistema de Ficheiros:** Guardar os dados coletados num cartão SD.

# Connecting the sensor

- Para a leitura da temperatura conectamos o pino VCC do sensor com o pino 3.3V do ESP32.
- Em seguida, conectamos o pino SDA e o pino SCLK do sensor ao GPIO\_0 e ao GPIO\_10 do ESP32, respetivamente.
- Por fim, foi feita a ligação do pino de ground do sensor ao ground do ESP32.

# Reading Temperatures

- Para a leitura da temperatura conectamos o pino VCC do sensor com o pino 3.3V do ESP32.
- Em primeiro lugar, é inicializado o sensor, e em seguida o mesmo é acordado.
- Depois disso, esperamos que o sensor acorde e ficamos á espera que as temperaturas estejam prontas. Quando tiveram fazemos a leituras das mesmas.

```
esp_err_t tc74_wakeup_and_read_temp(i2c_master_dev_handle_t sensorHandle, uint8_t* pTemp)
{
    esp_err_t err = tc74_wakeup(sensorHandle);
    if(err != ESP_OK)
        return err;

    while (!tc74_is_temperature_ready(sensorHandle))
    {
        vTaskDelay(10);
    }
    return tc74_read_temp_after_cfg(sensorHandle, pTemp);
}
```

# File System(SPI)

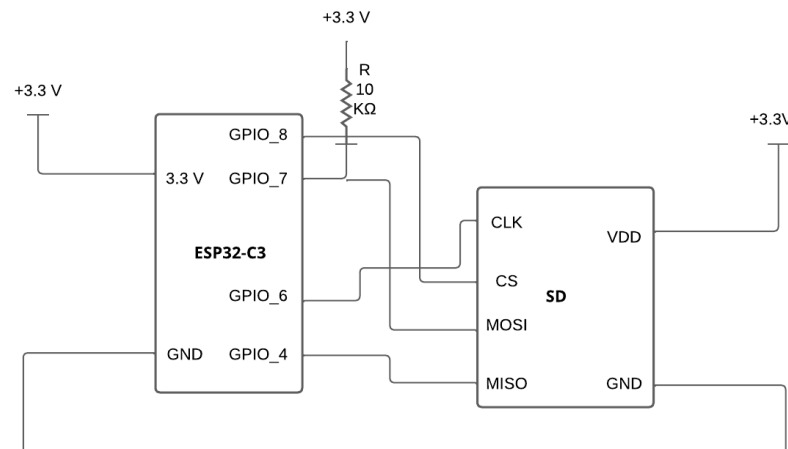
SD: 32GB

Formatado em FATFS

Adaptação do código dos exemplos

**OBS:** Ativar a opção:  
`CONFIG_EXAMPLE_FORMAT_IF_MOUNT_FAILED`

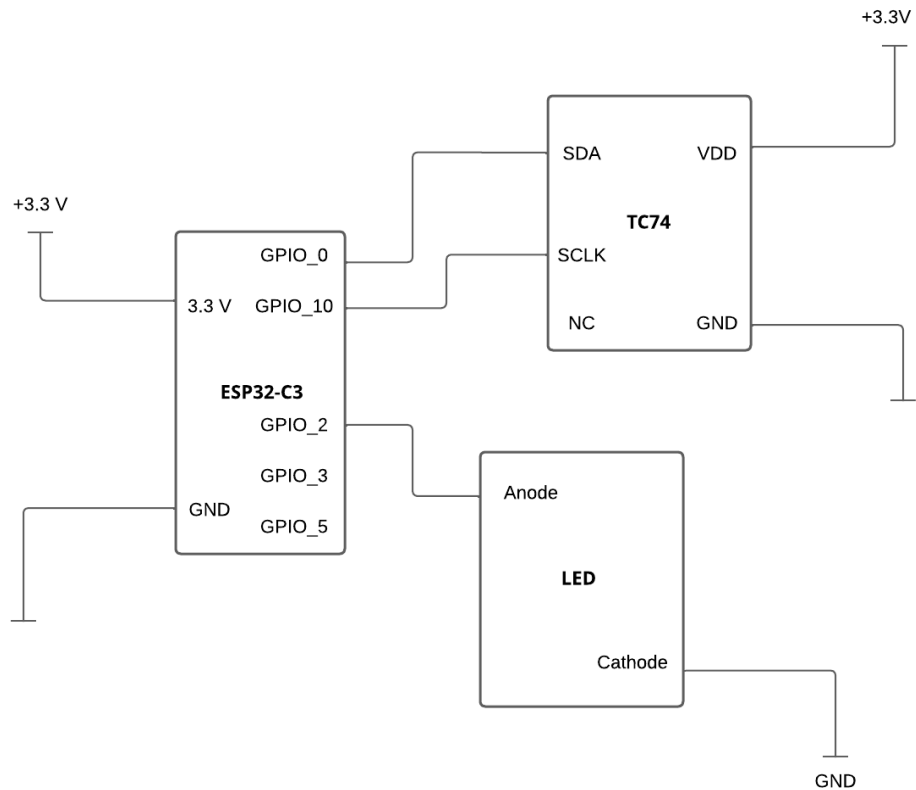
Escreve o valor lido, a data e a hora que  
houve variação na temperatura.



```
I (8910) Projeto: File written
I (9020) Projeto: File written
I (9130) Projeto: File written
```

```
Temp: 26 Date: Thu Jan 1 00:00:08 1970!
Temp: 28 Date: Thu Jan 1 00:00:08 1970!
Temp: 27 Date: Thu Jan 1 00:00:08 1970!
```

# General Diagram with a LED





# DashBoard

Adalberto / Nuno TC74

Sensor Temperatura

23°C

Increase Temperature by 10°C

Sleep

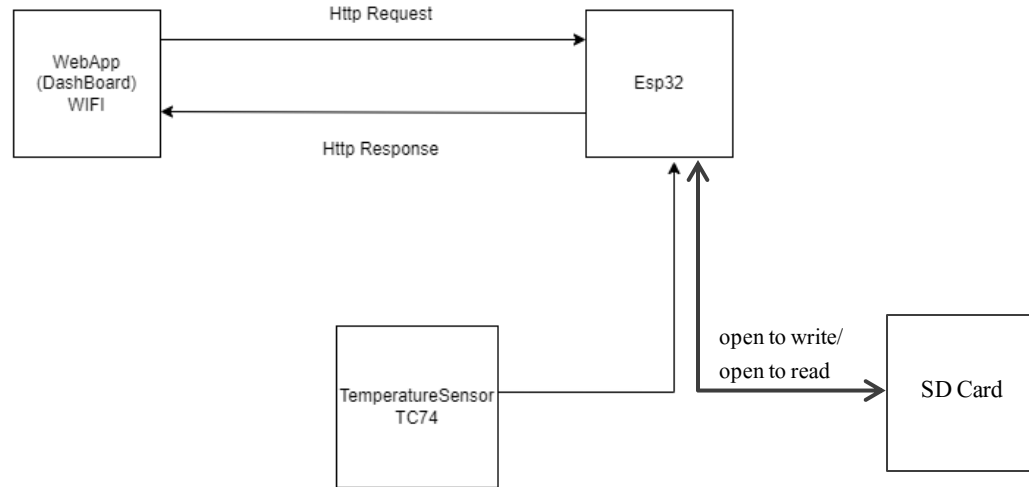
Wake Up

Para a criação da dashboard usámos a Biblioteca (**HTTP Server**) disponível no esp-idf e o frontend foi desenvolvido utilizando **HTML** e **CSS**. A dashboard mostra os valores da temperatura em tempo real, e permite a interação de um utilizador á distancia

A conexão à rede Wi-Fi é gerida pelas bibliotecas **esp\_wifi** e **esp\_event**, garantindo a comunicação contínua com o servidor HTTP.

1. /set\_temperature?temp=10
2. /temperature\_history
3. /sleep
4. /wakeup

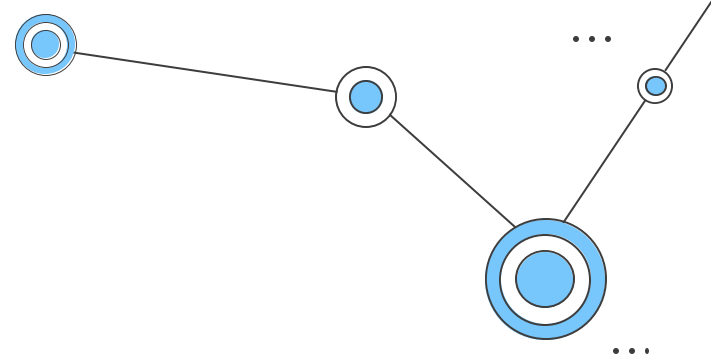
# Software Architecture



# References

- A **dashboard** foi baseada na biblioteca (HTTP Web Server) do esp32, e através do respetivos exemplos do express.
- O código de **leitura do sensor** desenvolvido por nós baseado nos exemplos da aula e no datasheet do sensor TC74.
- Ligação dos **LEDs** ao esp32, e o funcionamento deles foi baseado nas aulas práticas.
- Montagem do **cartão SD** baseado nas aulas teóricas, usando o exemplo do express.
- Montagem de todo o **circuito na placa branca** baseado em tudo o que aprendemos nas aulas.

# Future work



Usar sensor BME280

Implementar atualizações remotas do sistema (OTA)

Acertar a data e a hora do esp32  
(synchronize the time using an NTP server)

Implementar funcionalidades de acordo com os valores da temperatura.

