

Sistema embutido para a medição de distâncias e deteção de objetos a longo, médio e curto alcance.

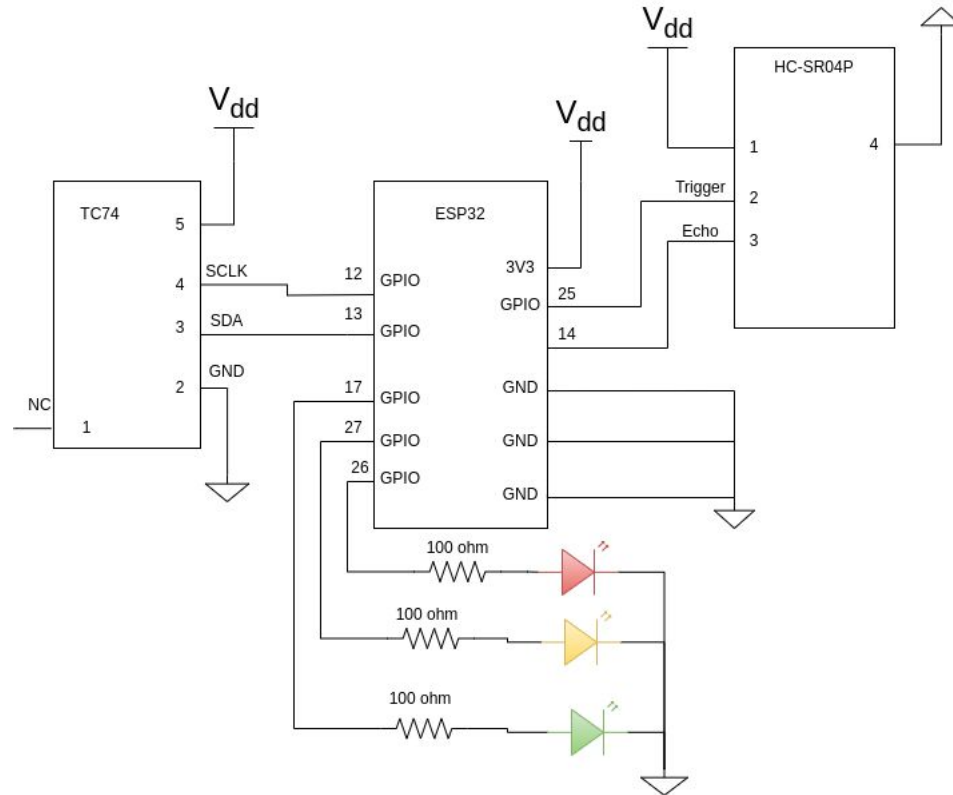
Arquiteturas para Sistemas Embutidos

Departamento de Eletrónica, Telecomunicações e Informática

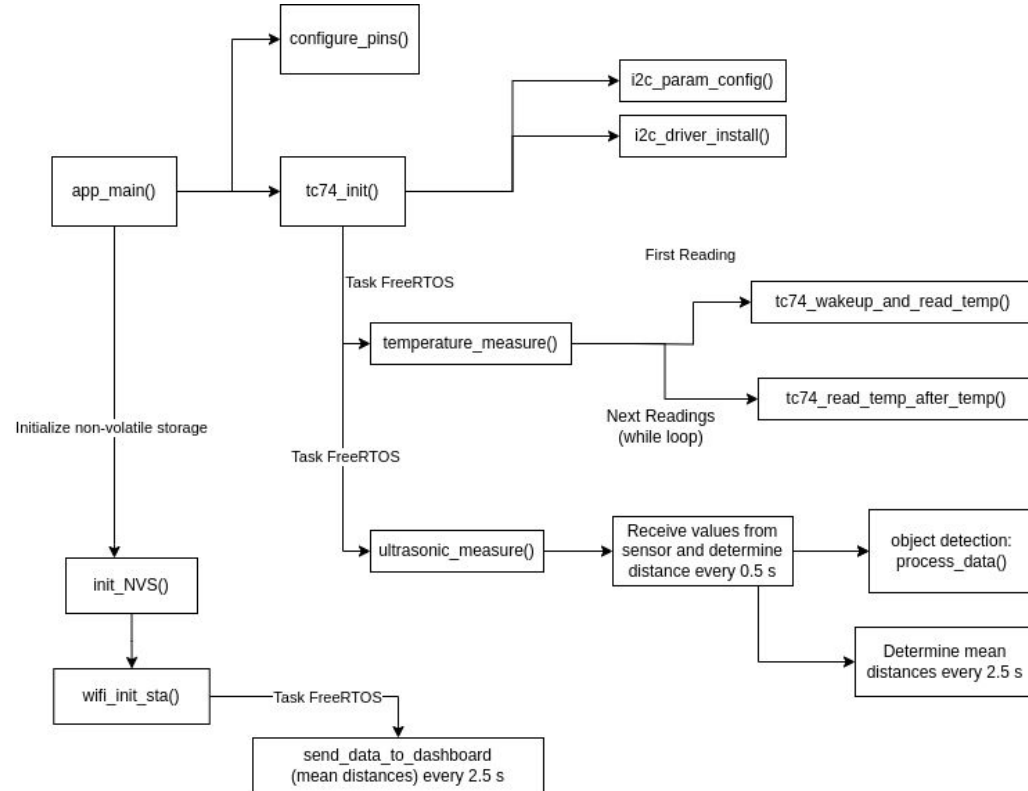
Mestrado Integrado em Engenharia de
Computadores e Telemática

Carlos Costa (88755), Ana Rosa (98678)

Arquitetura de Hardware

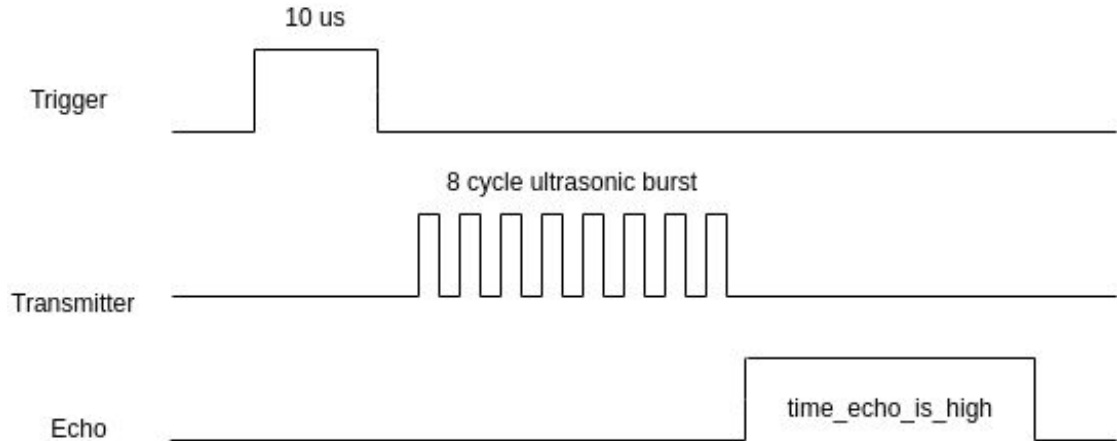
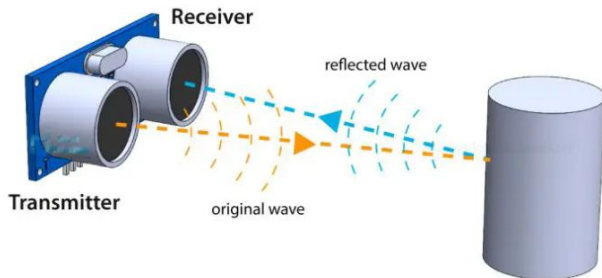


Arquitetura de Software



Obtenção da distância

- **Sinal Interno:** 10us impulso TTL
- O transdutor T converte o sinal elétrico em 8 pulsos com $f = 40\text{kHz}$.
- Se for detetado o objeto, o transdutor R recebe as ondas refletidas.
- Em função do tempo do echo determina-se a distância.



Obtenção da distância



Speed Of Sound:

- Velocidade a que as ondas viajam.
- Depende da temperatura e do meio através do qual a onda sonora se está a propagar.

$$\begin{aligned}\text{speedOfSound} &= 331.5 + (0.6 * \text{temperature}) \text{ (m/s)} \\ &= (331.5 + (0.6 * \text{temperature})) * 100 \text{ (cm/s)} \\ &= ((331.5 + (0.6 * \text{temperature})) * 100) / 1000000 \text{ (cm/us)}\end{aligned}$$

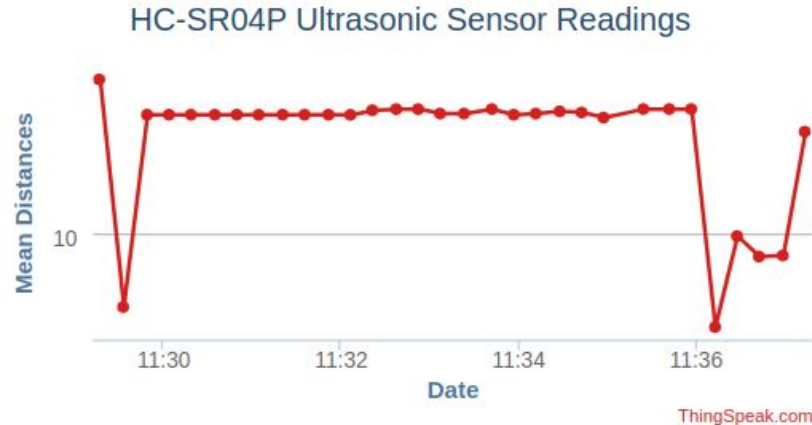
$$\text{distance} = (\text{speedOfSound} * \text{time_echo_is_high}) / 2$$

$$\text{max distance} = 4.50 \text{ cm}$$

$$\text{timeout} = ((4.50 * 2) / \text{speedOfSound}) + 1$$

Envio dos dados para a dashboard ThingSpeak

- A cada 0.1 s realiza-se a medição da temperatura ambiente em °C.
- A cada 0.5 s realiza-se a medição das distâncias.
- A cada 1.5 s insere-se a média do valor das distâncias determinadas, ou seja, após 3 medições é enviado a média para a dashboard.
- O envio dos dados é feito através da REST API do thingspeak por HTTP.



Ckecklist

Requisitos obrigatórios:

- A aplicação a executar no kit ESP32DevKitC tira partido do FreeRTOS. ✓
- Explorados os periféricos do ESP32, que fizerem sentido no contexto do projeto, incluindo aspetos de interrupções e DMA. ✓
- Os dados recolhidos do sensor e processados no ESP32 são apresentados num dashboard remoto. ✓
- Conectividade de rede (WiFi) para enviar os dados para a dashboard. ✓
- Disponibilizada ligação por terminal, independentemente do dashboard remoto. ✓
- Exploradas as várias funcionalidades das ferramentas de desenvolvimento, incluindo debug. ✓

Requisitos opcionais:

- Explorados os modos de baixo consumo energético do ESP32. ✗
- Suportadas atualizações remotas (Over-the-Air) do sistema. ✗
- Incluído um atuador. ✗
- Sistema de ficheiros para armazenar dados localmente. ✗



Baseline

- Para o envio dos dados para o dashboard, além da conexão wifi, os dados foram enviados através do módulo `esp_http_client` do espressif.
- Para o envio do sinal interno em microsegundos do esp32 para o pino trigger do sensor ultrasónico foi usado o módulo `rom/ets_sys.h` do espressif.

Avaliação

Ana Rosa: 60%

Carlos Costa: 40%

Bibliografia



Informação sensor HC-SR04P:

- <https://circuitdigest.com/microcontroller-projects/interfacing-hcsr04-ultrasonic-sensor-module-with-esp32>
- https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/?utm_content=cmp-true

Fontes código:

- <https://esp32tutorials.com/esp32-esp-idf-thingspeak-send-sensor-readings/>
- <https://esp32tutorials.com/hc-sr04-ultrasonic-sensor-esp32-esp-idf/#more-2275>
- <https://github.com/UncleRus/esp-idf-lib/tree/master/components/ultrasonic>
- <https://esp32tutorials.com/esp32-esp-idf-connect-wifi-station-mode-example/>
- https://github.com/espressif/esp-idf/blob/54576b7528b182256c027de86eb605a172bc2821/examples/protocols/esp_http_client/main/esp_http_client_example.c