**Memoria del Proyecto**

**1. Descripción del proyecto**

El proyecto utiliza un sensor ultrasónico para medir distancias entre 30 cm y 3 metros. Según la distancia detectada, el sistema enciende LEDs (rojo, amarillo o verde) para indicar la proximidad, activa un buzzer con diferentes frecuencias de pitido y muestra la distancia en la consola mediante UART.

Además, el sistema permite silenciar y reactivar el buzzer al pulsar la tecla 'M' en el teclado. Las distancias se dividen en tres rangos: el LED rojo indica la distancia más corta, el amarillo una intermedia y el verde la mayor. El pitido del buzzer también varía en velocidad dependiendo de la distancia.

**2. FSM: Descripción de la Interacción entre las Tareas**

**Estados:**

* **Espera**: El sistema espera eventos provenientes del sensor de ultrasonido o del teclado.
* **Leds+Buzzer**: Las mediciones se procesan para determinar la acción correspondiente (encendido de LEDs y frecuencia del buzzer).
* **UART:** El dato de la distancia se envía a través de la UART para poder visualizarlo en la consola.
* **Activa/desactiva Buzzer**: Al presionar la tecla M del teclado se desactiva o activa el buzzer.

**Eventos:**

* **Evento1**: Distancia detectada por el sensor.
* **Evento2**: Presión de la tecla M.
* **Evento3**: Acción completada.

**3. Descripción de las Tareas**

**3.1. Tarea de Medición (Task01)**

Mide la distancia y la usa para controlar LEDs y el buzzer. Configura el sensor de ultrasonido para medir distancias en intervalos periódicos. Dependiendo de los valores obtenidos del sensor de distancia manda diferentes interacciones a los LEDs y controla la frecuencia del pitido del buzzer. Clasifica la distancia en cuatro rangos:

* Distancia <= 50cm → +LEDs Rojosy buzzer 10Hz
* Distancia <= 100cm → +LEDs Amarillos y buzzer 5Hz
* Distancia <= 150cm → LEDs Verdes y buzzer 2Hz
* Distancia > 150 → LEDs y buzzer apagados

**3.2. Tarea de Comunicación (Task02)**

Envía las distancias medidas y los estados actuales a la consola UART. El estado del buzzer se controla mediante la funcion callback de recepción de la UART. Alterna entre los estados de silencio y activación dependiendo de la tecla M o m.

**4. Sincronización y Comunicación entre Tareas**

Las tareas funcionan en paralelo. Estas comparten datos pero no existen condiciones bloqueantes. Sin embargo, por precaución, hemos optado por usar mutexes que protegen el acceso a Distancia\_sensor para evitar condiciones de carrera entre StartTask01 y StartTask02

**5. Descripción de las Comunicaciones**

**5.1. Comunicación UART (Serial)**

La comunicación serial es bidireccional. Permite enviar los datos de distancia y el estado del sistema al host, al igual que recibir datos desde la terminal.

Se utiliza HAL\_UART\_Transmit para enviar datos en formato serial al exterior, como las mediciones de distancia. Es una comunicación síncrona en la que los datos se envían carácter por carácter.

Usa HAL\_UART\_Receive\_IT y callbacks (HAL\_UART\_RxCpltCallback) para manejar datos recibidos de forma asíncrona, como comandos enviados por un usuario. Esto permite una respuesta en tiempo real a los comandos sin bloquear la ejecución principal del sistema.

**Formato de los datos enviados**:

"Distancia: %lu.%02lu cm\r\n", distancia\_entera, distancia\_decimal

**6. Descripción del Programa Interfaz en el Host**

El programa de interfaz en el host se ejecuta en un sistema basado en PC para visualizar y controlar el sistema.

**Características:**

* **Consola UART**:

Muestra en tiempo real la distancia detectada por el sensor de ultrasonido.

* **Comandos**:

‘M’ o ‘m’(teclado): Silencia o activa el buzzer.

**repositorio de Git:** <https://github.com/operezdearrilu/TRE_Proyecto.git>