

LABORATORIO DE MICROPROCESADORES

TRABAJO PRÁCTICO FINAL LIFE MONITOR

GRUPO 1

- ▶ Lisandro Alvarez
- ▶ Tomás González
 - ▶ Rocío Parra
- ▶ Gonzalo Reina



Diseño

Monitor de signos vitales

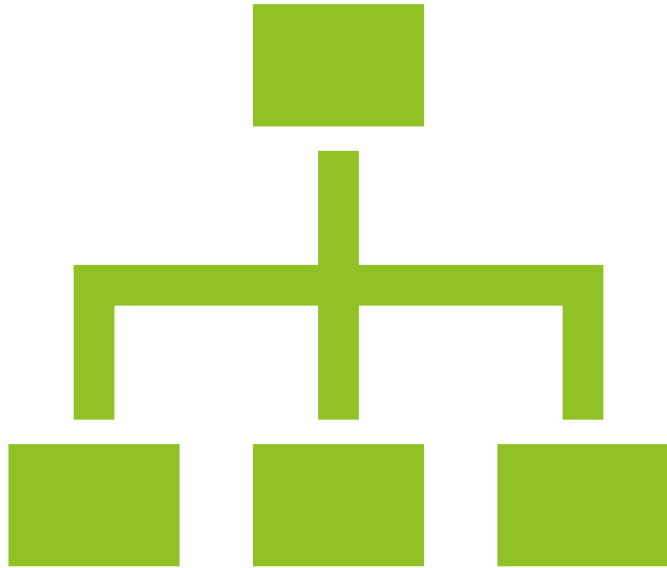
Objetivos

Requerimientos

- ▶ Medición de signos vitales
 - ▶ Saturación de oxígeno
 - ▶ Frecuencia cardíaca
 - ▶ Temperatura corporal
 - ▶ Electrocardiograma
- ▶ Información en tiempo real en aplicación de celular o tablet
- ▶ Alertas (pregrabadas en MP3) si alguna variable está fuera de rango

Herramientas disponibles

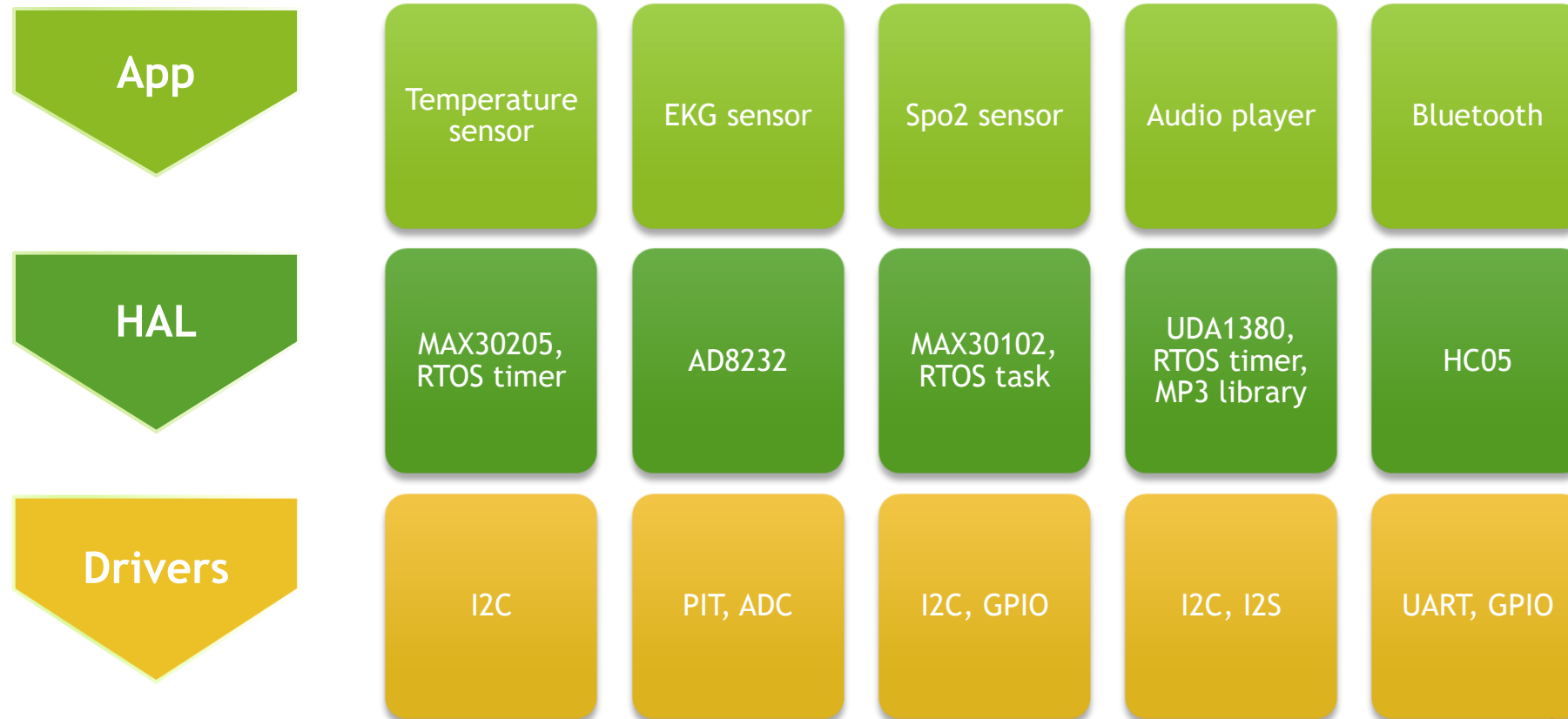
- ▶ Microprocesador FRDM-K64F
- ▶ ECG-AFE (AD8232)
- ▶ Sensor de temperatura (MAX30205)
- ▶ Oxímetro de pulso (MAX30102)
- ▶ Interface Bluetooth (HC05)
- ▶ I2S Codec (UDA1380)



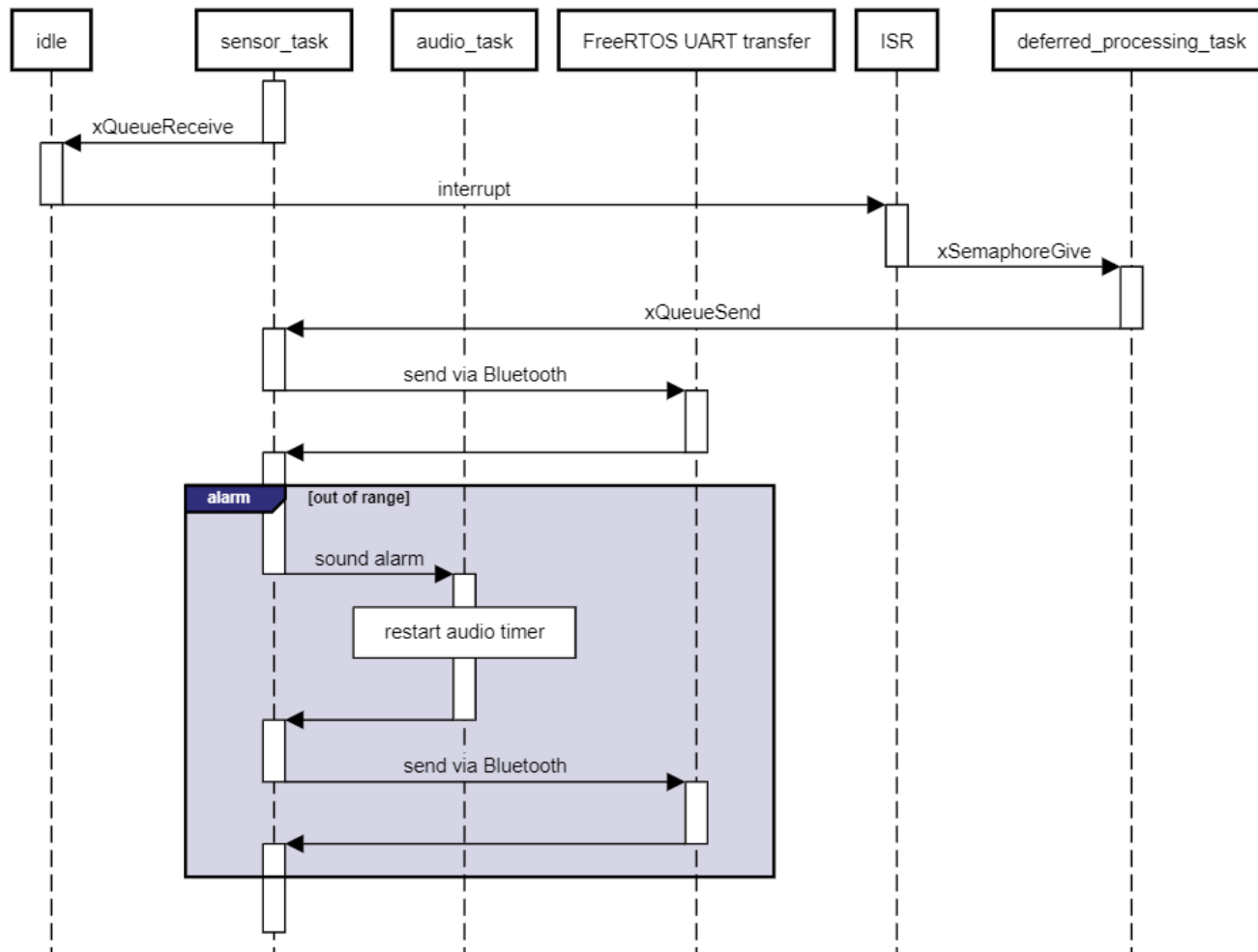
Implementación

Arquitectura de Firmware

División en módulos y jerarquía



Example app flow

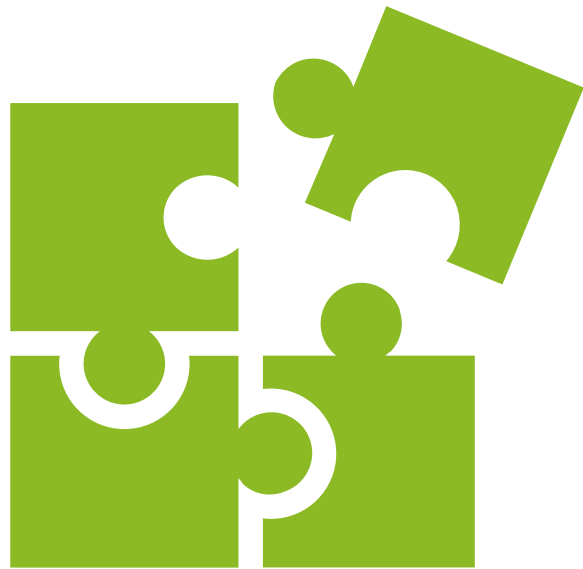


Tareas

- ▶ Tarea principal: recibir eventos de sensors y despacharlos, detector variables fuera de rango
- ▶ Audio task
- ▶ Transferencias UART, I2C.. manejadas por FreeRTOS
- ▶ Tarea de procesamiento diferido para oxímetro

Tasks e interrupciones

- ▶ SPO2_CALC_TASK_PRIORITY = 1
 - ▶ SPO2_TASK_PRIORITY = 2
 - ▶ SENSOR_TASK_PRIORITY = 3
 - ▶ configTIMER_TASK_PRIORITY = 4
 - ▶ AUDIO_TASK_PRIORITY = 5
-
- ▶ NVIC_SetPriority(I2C0_IRQn, 4);
 - ▶ NVIC_SetPriority(PORTB_IRQn, 4);
 - ▶ NVIC_SetPriority(ADC0_IRQn, 4);
 - ▶ NVIC_SetPriority(UART3_RX_TX_IRQn, 5);
 - ▶ NVIC_SetPriority(I2S0_Tx_IRQn, 5);



Módulos

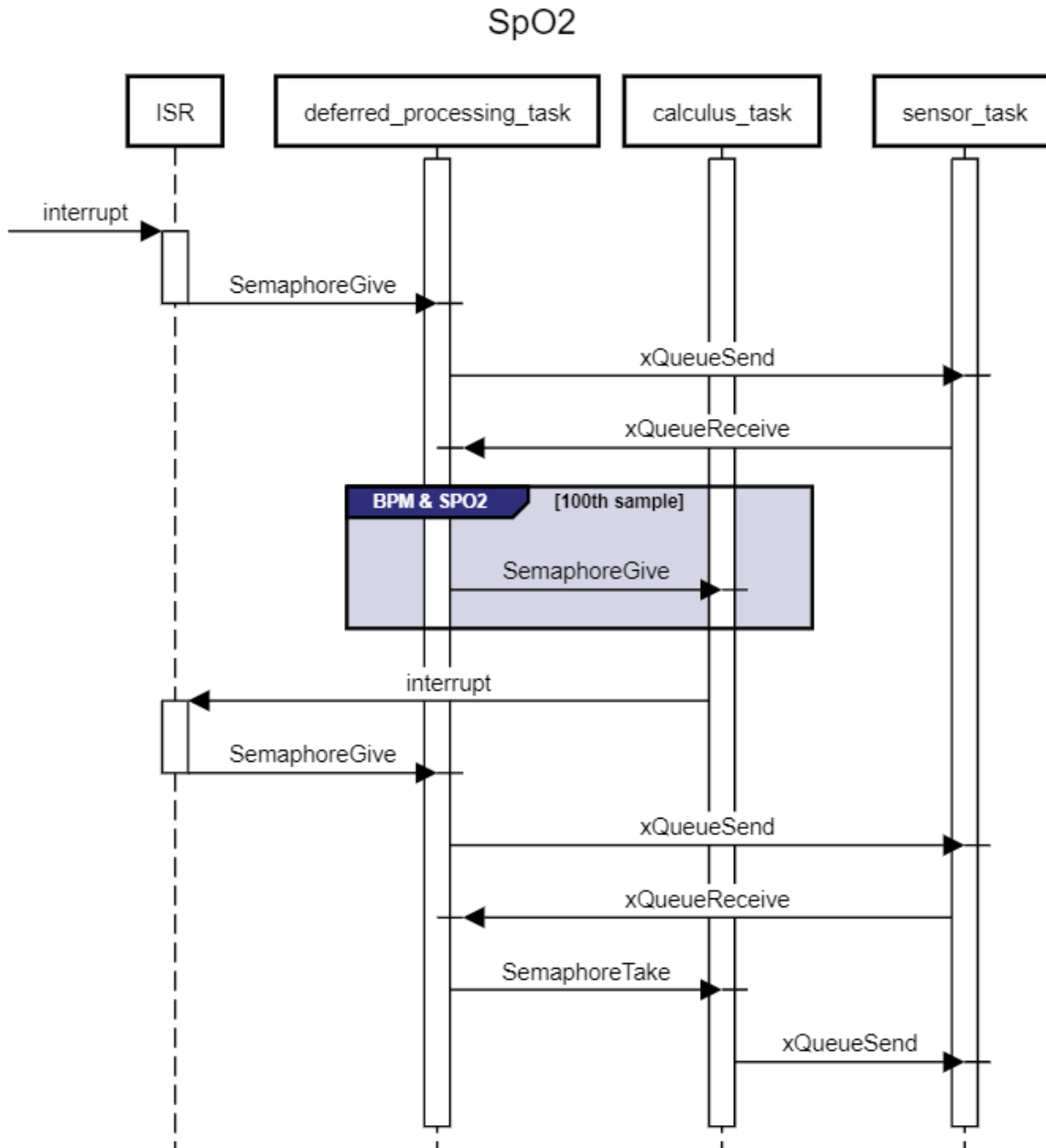
- ECG-AFE (AD8232)
- Sensor de temperatura (MAX30205)
- Oxímetro de pulso (MAX30102)
- Interface Bluetooth (HC05)
- Módulo I2S (UDA1380)

Sensor de temperatura (MAX30205)

- ▶ Lectura periódica triggereada por timer de FreeRTOS
- ▶ Transferencia I2C manejada por FreeRTOS (driver intermedio de desarrollo propio)
- ▶ Inmediatamente después de la lectura se envía el evento con la nueva muestra
- ▶ Frecuencia de sampleo: 1Hz (1s)

Sensor de ECG (AD8232)

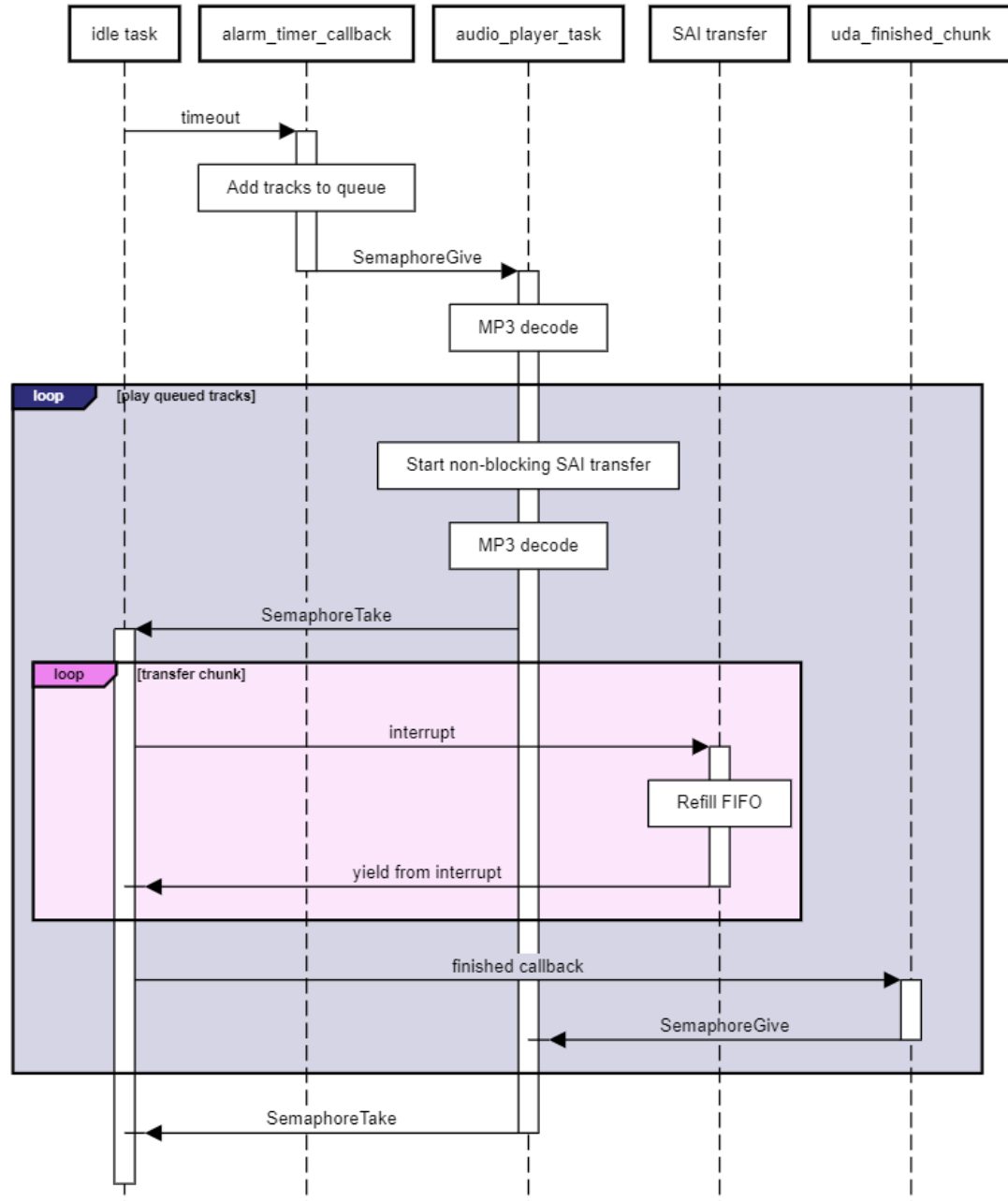
- ▶ Lectura periódica triggereada por PIT
- ▶ El callback de conversión finalizada del ADC se utiliza para enviar la nueva muestra
- ▶ Frecuencia de sampleo: 180Hz (~5ms)



Oxímetro de pulso (MAX30102)

- ▶ Lectura periódica de LEDs triggereada por GPIO
- ▶ spo2_task: enviar y guardar samples
- ▶ calculus_task: calcular y enviar BPM y spO2 (cada 100 samples)
- ▶ Frecuencia de muestreo: 25Hz (4ms)

Audio player

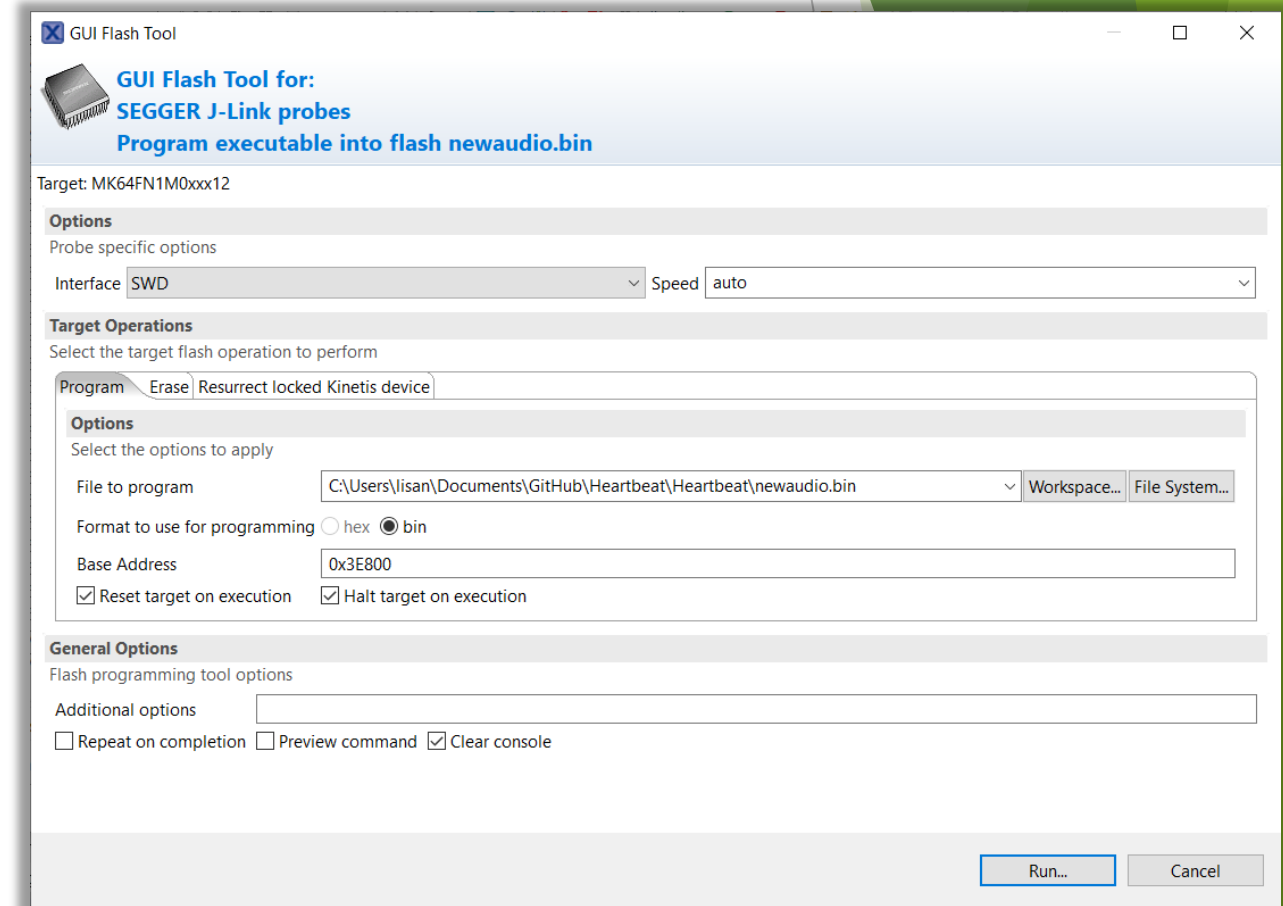


Audio playback

- Verificación periódica del estado de los signos vitales para reproducir audios (timer FreeRTOS)
- Reproducción basada en interrupciones

Programación de memoria Flash

- Utilización de GUI Flash Tool de MCUXpresso
- Visualización del uso de memoria del programa mediante herramientas de MCUXpresso para determinar sectores disponibles.



Transferencia HC05

- ▶ BT_com_send_meas:
 - ▶ Recibe evento de sensor (source y value)
 - ▶ Genera paquete:
 - ▶ [Header][Tag][Length][Data]
- ▶ BT_com_set_alarm
 - ▶ Recibe fuente de la alarma y booleano (set o reset)
 - ▶ Genera paquete:
 - ▶ [Header][Tag: 'A'][Length: 2][Data]
 - ▶ Data: [1 byte source][1 byte 'S' o 'R']

```
bt_com_state_t BT_com_send_meas(sensor_event_t sens_ev){
    sens_ev.value = (float)sens_ev.value;
    bt_com_state_t success = BT_COM_FAILURE;
    if (BT_com_is_connected()){
        int i = 0;
        // header
        for (i = 0; i < HEADER_LEN; i++){
            buffer[i] = (uint8_t)'F';
        }
        //tag
        buffer[i++] = BT_com_get_tag(sens_ev.type);
        // length
        buffer[i++] = (uint8_t)sizeof(sens_ev.value);
        // value
        memcpy(&buffer[i], &(sens_ev.value), sizeof(sens_ev.value));
        // UART Transmission to HC05.
        UART_RTOS_Send(&UART3_rtos_handle, buffer, i + sizeof(sens_ev.value));
        success = BT_COM_SUCCESS;
    }
    else{
        success = BT_COM_FAILURE;
    }
    return success;
}
```

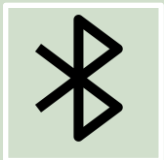
```
bt_com_state_t BT_com_set_alarm(sensor_event_type_t source, bool set){
    bt_com_state_t success = BT_COM_FAILURE;
    if(BT_com_is_connected()){
        int i = 0;
        // header
        for (i = 0; i < HEADER_LEN; i++){
            buffer[i] = (uint8_t)'F';
        }
        //tag
        buffer[i++] = (uint8_t)'A';
        buffer[i++] = 2;
        // Message: Source + Set or Reset.
        buffer[i++] = BT_com_get_tag(source);
        buffer[i++] = (uint8_t)(set ? 'S' : 'R');
        // UART Transmission to HC05
        UART_RTOS_Send(&UART3_rtos_handle, buffer, i);
        success = BT_COM_SUCCESS;
    }
    else{
        success = BT_COM_FAILURE;
    }
    return success;
}
```

Heartbeat - Android App



Main Activity

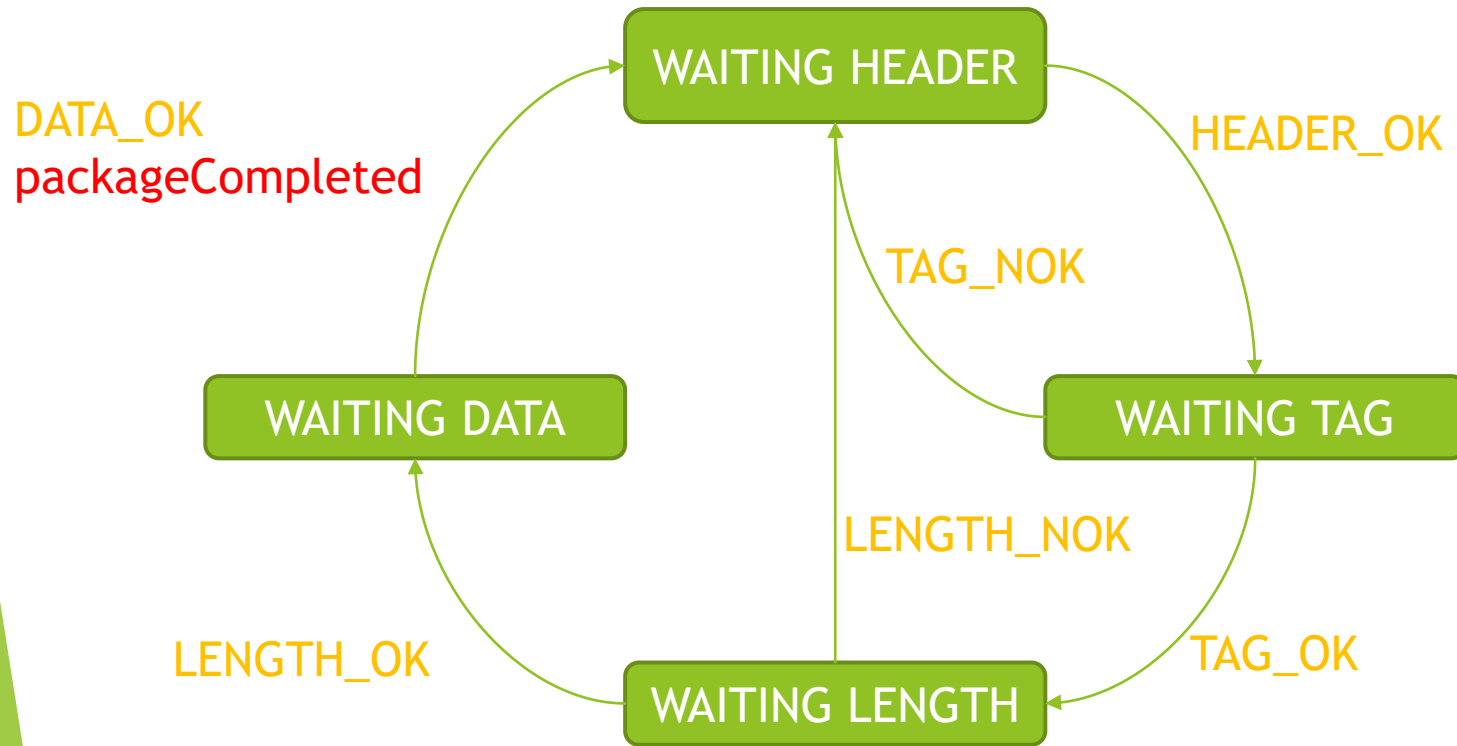
Control de elementos gráficos
Parseo de datos y alarmas
Instancia el thread de comunicación



Bluetooth Connection Service

Maneja la conexión con el dispositivo bluetooth
Parsea los datos recibidos y envía paquetes a Main Activity.

Parseo de Paquetes







```
while (true) {
    try {
        // Read 1 byte from input stream
        receivedByte = mmInStream.read();
        // parse byte & check if package is completed
        if(fsmReader.packageCompleted(receivedByte)) {
            tag = (char) fsmReader.getTag();
            data = fsmReader.getData();
            Intent newDataIntent = new Intent("newData");
            newDataIntent.putExtra("tag", tag);
            newDataIntent.putExtra("data", data);
            // Broadcast data to main activity
            LocalBroadcastManager.getInstance(mContext).
                sendBroadcast(newDataIntent);
        }
    } catch (IOException e) {
        Log.e(TAG, "write: Error reading Input Stream. "
            + e.getMessage() );
        break;
    }
}
```

Tiempo y memoria usados por tasks

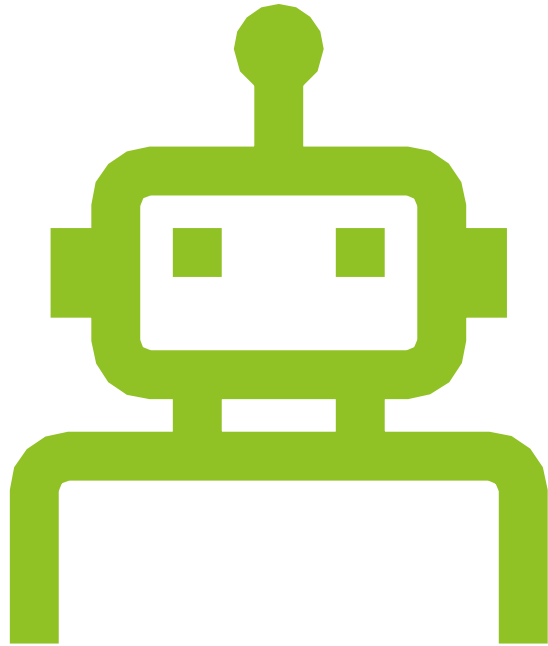
TCB#	Task Name	Task Handle	Task State	Priority	Stack Usage	Event Object	Runtime
> 1	sensor task	0x20008af0	<input type="checkbox"/> Suspended	3 (3)	548 B / 1016 B	Unknown (0x2000825c)	0x13f7 (1,2%)
> 2	IDLE	0x20008e18	 Running	0 (0)	88 B / 352 B		0x63291 (97,0%)
> 3	Tmr Svc	0x20009178	 Blocked	4 (4)	472 B / 712 B	TmrQ (Rx)	0xadc (0,7%)
> 4	spo2 task	0x20009640	<input type="checkbox"/> Suspended	2 (2)	464 B / 1016 B	Unknown (0x2000a0b4)	0x5d7 (0,4%)
> 5	hr spo2 task	0x2000a008	<input type="checkbox"/> Suspended	1 (1)	1,21 kB / 2,3 kB	Unknown (0x2000a10c)	0x4b (0,0%)
> 6	audio player task	0x2000a628	<input type="checkbox"/> Suspended	4 (4)	584 B / 1,12 kB	Unknown (0x2000a164)	0xb70 (0,7%)

Uso de memoria Flash y SRAM

Region	Start address	End address	Size	Free	Used	Usage (%)
 SRAM_UPPER	0x20000000	0x20030000	192 KB	117,56 KB	74,44 KB	38,77%
 PROGRAM_FLASH	0x0	0x100000	1 MB	913,45 KB	110,55 KB	10,80%
 SRAM_LOWER	0x1fff0000	0x20000000	64 KB	64 KB	0 B	0,00%
 FLEX_RAM	0x14000000	0x14001000	4 KB	4 KB	0 B	0,00%

Bibliografía y recursos

- ▶ [Helix Library](#)
- ▶ [New signal processing methodology for obtaining heart rate and SpO2 data from the MAX30102 sensor manufactured by MAXIM Integrated Products, Inc.](#)
- ▶ Freescale SDK
- ▶ FreeRTOS
- ▶ [Tutorial: Using Runtime Statistics with Amazon FreeRTOS V10](#)
- ▶ [Mastering FreeRTOS Real Time Kernel](#)



¡Gracias por
su atención!