



GUIA DE TRABAJO PROYECTO SALVA

[SUBTÍTULO DEL DOCUMENTO]



1. ARTIK-020

Para la programación del ARTIK 020 he utilizado el programa [Simplicity Studio](#) de Samsung. La versión de SDK del programa final está hecha con el *Gecko SDK Suite: Bluetooth 2.8.0.0, Flex 2.2.1.0, Kernel, Micrium, OS*, según el entorno, la versión 2.2.1.

Para la depuración de mensajes abriendo un terminal en el puerto serie que esté conectado el ARTIK. Yo utilizaba *Realterm* para el ARTIK 020 y *Putty* para el ARTIK 520, pero puede servir cualquiera.

En cuanto a la versión final del programa está en el github [ARTIK-020](#). Comentarios de este programa:

- *Advertisement.h*: en este fichero se configuran los parámetros de los intervalos de descubrimiento. Está puesto al mínimo para la prueba, pero comentado tiene la opción que se iba a configurar finalmente. Para más información de este parámetro mirar en la documentación [cmd le gap set adv parameter](#). En *Advertisement.c* se implementa las funciones relacionadas con el proceso de descubrimiento.
- *App.c*: aquí se implementa la mayor parte del programa. Sus funciones son:
 - *App_init*: inicializa los dispositivos con los nombres y los parámetros
 - *App_handler*: implementa la máquina de estados que se describe en la memoria para gestionar los eventos *bluetooth*. También incluye la gestión de los modos de energía. Es la función más importante del programa y la que se encarga de hacer todo el trabajo. En la [memoria](#), en la sección 5.1 se explica todo en detalle.
 - *Pinta_Pantalla*: actualiza la pantalla del EDP.
- *Connected.h*: se establecen los intervalos de conexión. Están al mínimo y estos no se deberían de tocar, ya que es la configuración óptima.
- *Main.c*: inicializamos el programa y todo lo necesario de hardware. Para activar o desactivar la opción de *debug* por puerto serie comentar/descomentar función "RETARGET_SerialInit()". En principio se dejó sin usar para la versión final.
- *Spp.c* y *Spp.h*: en estos ficheros definimos los parámetros, funciones y estructuras que se van a encargar de almacenamiento de imágenes. En principio está hecho para un solo tamaño de imagen, pero si se quieren añadir más, se deberá de implementar en estos ficheros para seguir con la estructura del programa.

2. ARTIK 520

Para la programación del ARTIK 520 utilizamos el [ARTIK IDE](#) de Samsung. La versión del SDK es la *ARTIK 5 SDK for UBUNTU v1.6*. El toolchain es el *gcc-arm-linux-gnueabihf v5.41-2017.05*.

Para la comunicación del dispositivo lo hicimos con puerto serie (*Putty*). Implementamos los dos programas principales, el de conexión y envío de imagen por un lado y el de escaneo:

[ARTIK-BLE-CONNECT](#)

En SEND-ARTIK5.c está todo el código y básicamente se implementa la máquina de estados que se detalla en la [memoria](#) en la sección 5.2.2. Comentarios principales de este programa:

- El *SPP_SERVICE* y *SPP_DATA_CHAR* son los UUIDs asignados en el ARTIK020 para ese servicio y esa característica. Estos no varían.
- *Remote_address* será la dirección MAC del dispositivo al que tendremos que mandar la imagen.

[ARTIK-BLE-SCAN](#)

Se encarga de escanear los dispositivos bluetooth y mostrarlos por la salida *stdout*. En la memoria [memoria](#) sección 5.2.1 se detalla todo lo importante de este programa.

3. Como probar

1. Clonar repositorios y cargar en los dispositivos (ARTIK-020 y ARTIK-520).
2. Una vez funcionados se recomienda probar la conexión al ARTIK-020 con el móvil, para asegurarnos de que se pueda conectar.
3. En el ARTIK-520 ejecutar primero la búsqueda de dispositivos.
4. Por ultimo ejecutar el script de envío de imagen.