# INTRODUCCIÓN

En este documento se explicará la estructura de los dos proyectos de software utilizados en el trabajo.

# PROYECTOS UTILIZADOS

En el trabajo se han utilizado 2 proyectos de software diferentes, cada uno para cada una de las plataformas hardware utilizadas, la plataforma de la Coockie y el kit de evaluación de STM. Estos dos proyectos se han llamado Coockie (proyecto software que corre en la plataforma Coockie como su propio nombre indica para el control del módulo KTWM102) y UART (proyecto software que corre en la plataforma STM con el que se maneja los dos Dongles de Evaluación KTDG102).

# PROYECTO UART (STM)

Como ya se ha mencionado en 2 PROYECTOS UTILIZADOS, este proyecto se encarga de la comunicación UART con los Dongles y del proceso de unión a la red Thread de estos. Este proyecto se ha generado con STM32CubeMX para la configuración del microcontrolador y se ha programado y compilado con Keil µVision 5.

La estructura del código utilizada, en cuanto a ficheros y dependencias entre ellos es la indicada a continuación en Ilustración 1:

Diagrama

Descripción generada automáticamente

Ilustración 1 Estructura de Ficheros Proyecto Dongles

## DRIVERS STM

Estos drivers son los ficheros .c y .h que se incluyen por defecto al generar el proyecto con la configuración de puertos de comunicación, relojes, conversores ADC, salidas y entradas a utilizar en el proyecto, etc que se haya elegido para el propio microcontrolador. Incluyen las funciones, estructuras y variables necesarias para el uso de las diferentes funcionalidades del microcontrolador.

## COMANDOS.H

Como su propio nombre indica, en este fichero, se introducen los comandos básicos a enviar a los Dongles. Se introdujo este documento para evitar cargar demasiado el fichero principal al ser comandos que, en su mayoría, no van a ser modificados salvo los correspondientes a la configuración de la red elegida.

Por ello, para los comandos de configuración de red, se deja en este fichero el comando sin ninguna PAYLOAD, la cual se definirá en main.c, fichero el cual se encargará de unir posteriormente comando y payload correspondiente antes de gestionar su transmisión. Esto hace así más accesible una modificación de la configuración sin llegar a mezclar comandos y código principal en un mismo fichero.

## COBS.H

En este fichero se definen las estructuras, variables y funciones necesarias para la codificación y decodificación COBS de los comandos a transmitir y las respuestas que provengan de los Dongles. Para la codificación, se ha utilizado una función para microcontroladores como Arduino, mientras que para la decodificación se ha podido adaptar al entorno del proyecto la función desarrollada por Kirale Technologies.

Las dependencias de esta librería son: *stding.h, stdio.h, stdlib.h y string.h*.

## LIBRERÍAS MAIN.H E INTYPES.H

Estas librerías son las que incluyen en los ficheros COBS.h y comandos.h de los tipos de variables a utilizar en el proyecto, desde los tipos de datos básicos hasta las variables para los puertos UART.

## MAIN.C

Este fichero es el fichero principal, donde se ejecutará el programa o se gestionarán las llamadas a los procesos correspondientes para el funcionamiento correcto del proyecto.

También, se definen las funciones principales del proyecto, incluyendo la configuración e inicialización del microcontrolador y la gestión de la transmisión y la recepción de los comandos y sus respuestas. Esto último implica también la gestión de llamadas a la decodificación y codificación de los comandos y respuestas, las funciones destinadas a la lectura y envío de datos, el cálculo del checksum. También define y llama a las funciones encargadas de enviar la secuencia de comandos al correspondiente Dongle para su configuración y unión a la red.

Después de la configuración de los nodos y estén unidos a la red, se configura la IP de los Dongles y se abre un socket en un puerto UDP, y finalmente se ejecuta en bucle una rutina de envío de mensajes UDP desde los Dongles a otros nodos de la red, para lo cual habrá que saber previamente las direcciones IP de los nodos y configurarlas correctamente en los comandos de envío de mensajes UDP (SendHelloX) para un correcto envío.

Para las direcciones IP, si se han configurado los nodos o se van a configurar los nodos de igual manera que los Dongles, con una IP conocida previamente, configurar envíos con esas IP. En caso de no conocer previamente las direcciones IP, se deberá esperar a ejecutar el programa a que los demás nodos estén en la red, pudiendo ver sus direcciones IP tanto en la herramienta KiTools de PC, como en el Panel de Administración Web proporcionado por el Border Router.

# PROYECTO COOCKIE

Este proyecto es el utilizado en la plataforma Coockie, como ya se ha mencionado en 2 PROYECTOS UTILIZADOS. Este proyecto ha sigo creado, programado y compilado con Keil µVision 3. Para la carga del fichero en la plataforma Coockie se ha utilizado el programa Windows Serial Downloader for ADuC8XX MicroConverters proporcionado por Analog Devices.

La estructura del código utilizada, en cuanto a ficheros y dependencias entre ellos es la indicada a continuación en Ilustración 2:

Diagrama

Descripción generada automáticamente

Ilustración 2 Estructura de Ficheros Proyecto Coockie

## START\_AD.A51

Este fichero es un código de arranque, que se ejecuta nada más arrancar el dispositivo, ya sea mediante el encendido como mediante un reset. Es un fichero por defecto que no ha de ser modificado salvo para la modificación de alguna de las configuraciones, como puede ser activar o desactivar la memoria RAM Extendida, etc.

## ADUC841.H

Este fichero gestiona la configuración de los registros del microcontrolador utilizado en la Coockie. No se recomienda su modificación. Para modificar algún valor se recomienda hacerse desde otro fichero solamente cuando se necesita. Así se evita el poder perder en cierta manera fichero con la configuración y valores de registros por defecto.

## COMANDOS.H

Como su propio nombre indica, en este fichero, se introducen los comandos básicos a enviar al módulo KTWM102. Se introdujo este documento para evitar cargar demasiado el fichero principal al ser comandos que, en su mayoría, no van a ser modificados.

También se incluyen en este fichero la definición de los tipos de variables uint8\_t, uint16\_t, int16\_t y \_Bool, puesto que, por defecto, en este entorno no vienen definidos ni implementados. Se incluyen en este fichero por motivos de optimización, evitando generar ficheros extra para su definición y el definirlos en varios ficheros.

## COBS.H

En este fichero se definen las estructuras, variables y funciones necesarias para la codificación y decodificación COBS de los comandos a transmitir y las respuestas que provengan del módulo KTWM102.

Al igual que lo visto en PROYECTO DONGLES, para la codificación, se ha utilizado una función para microcontroladores como Arduino, mientras que para la decodificación se ha podido adaptar al entorno del proyecto la función desarrollada por Kirale Technologies. Sin embargo, se ha añadido la función de codificación desarrollada por Kirale, la cual no está adaptada ni implementada al completo, pero se implementará en líneas futuras de desarrollo de este proyecto

La dependencia única de este fichero es la librería *comandos.h,* esto se debe a la definición de los tipos de variables necesarios en dicha librería, permitiendo así disponer de ellos en esta librería.

## MAIN.C

Este fichero es el fichero principal, donde se ejecutará el programa o se gestionarán las llamadas a los procesos correspondientes para el funcionamiento correcto del proyecto.

También, se definen las funciones principales del proyecto, incluyendo la configuración e inicialización del microcontrolador y la gestión de la transmisión y la recepción de los comandos y sus respuestas. Esto último implica también la gestión de llamadas a la decodificación y codificación de los comandos y respuestas, las funciones destinadas a la lectura y envío de datos, el cálculo del checksum. También define y llama a las funciones encargadas de enviar la secuencia de comandos al módulo KTWM102 para su configuración y unión a la red y de gestionar la recepción de sus respuestas.

Tras la unión del nodo a la red, se entra en un bucle, en el cuál se procede a transmitir mensajes UDP a un nodo de la red cada 5 segundos. Para esto, deberá configurarse, antes de ejecutar el programa, la dirección IP a la que se desea transmitir.

Este fichero incluye al proyecto la librería *stdio.h*  como dependencia, la cuál implementa las funciones para la escritura y lectura de datos en el puerto UART, pudiendo realizar así su transmisión y su recepción. A su vez, se integran los ficheros *ADuC841.h* y *COBS.h* como dependencias, integrando a su vez *comandos.h* con este último. Esto permite a *main.c* utilizar las funciones COBS a la vez que usar los tipos de variables necesarios y los comandos a enviar al módulo KTWM102, al igual que modificar los registros del microcontrolador en caso de ser necesario.