



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**título del TFG
Documentación Técnica**



Presentado por Enrique del Olmo Dominguez
en Universidad de Burgos — 15 de junio
de 2022

Cotutores: Alejandro Merino y Daniel Sarabia

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	iv
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
Apéndice B Especificación de Requisitos	5
B.1. Introducción	5
B.2. Catalogo de requisitos	6
B.3. Especificación de requisitos	8
Apéndice C Especificación de diseño	15
C.1. Introducción	15
C.2. Diseño de datos	15
C.3. diseño arquitectónico	16
Apéndice D Documentación técnica de programación	19
D.1. Introducción	19
D.2. Estructura de directorios	19
D.3. Manual del programador	20
D.4. Compilación, instalación y ejecución del proyecto	23
Apéndice E Documentación de usuario	25
E.1. Introducción	25

E.2. Requisitos de usuarios	25
E.3. Manual del usuario	26

Índice de figuras

C.1. T	16
D.1. Estructura de directorios	19
D.2.	21
D.3.	21
D.4.	22
D.5.	22
D.6.	23
D.7. U	23
D.8. Interfaz de la herramienta Packet Sender	24
E.1. Botones de la placa maestro	26

Índice de tablas

B.1. CU-1 Seleccionar SE	8
B.2. CU-2 Seleccionar SE	9
B.3. CU-3 Seleccionar SE	10
B.4. CU-4 Seleccionar SE	11
B.5. CU-5 Seleccionar SE	12
B.6. CU-6 Seleccionar SE	13

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En este apartado se describen los resultados obtenidos tras la realización del proyecto.

Se trataran dos temas principales. El primero corresponde a la planificación temporal del trabajo, en la cual veremos las tareas realizadas y los tiempos requeridos para llevarlas a cabo. También veremos el tiempo total de la realización del trabajo de fin de grado. Puesto que se esta utilizando una metodología Scrum algunos tiempos pueden no ajustarse lo suficiente a lo esperado sobretodo los de los primeros sprint realizados ya que en estos se debe llevar un proceso de medición de fuerzas y dificultades del trabajo a realizar. el segundo tema del que se va a desarrollar es la realización de un análisis de la viabilidad de un proyecto de estas características en la vida real. Para ello nos centraremos en analizar las variables económicas y legales. En cuanto a este ultimo punto veremos los tipos de licencias que se pueden utilizar en este tipo de software

A.2. Planificación temporal

Como ya hemos comentado la planificación del proyecto se llevo a cabo siguiendo la metodología Scrum que esta basada entre otras cosas, en el uso de sprints para racionar y organizar la carga de tareas.

Según esta metodología estos sprints deben durar entre 1 y 3 semanas. En este caso y puesto que mi situación personal ha sido distinta a lo largo de este proceso la duración de estos sprints no ha sido siempre la misma. Esto

se debe a que mis obligaciones han ido variando a medida que avanzaba el cuatrimestre. Al principio de este, durante los 3 primeros Sprint y comienzos del cuarto disponía de bastante mas tiempo para su realización por lo que era capaz de meter mas horas al día, posteriormente empecé a cursar las practicas curriculares con un total de seis horas diarias lo que dificultó seguir con el mismo ritmo de horas dedicadas al TFG. Si que se ha tratado de que los sprints estuvieran formados por aproximadamente el mismo numero de horas. Estas horas están expuestas en el repositorio de Git Hub y cada tarea tiene asignado un tag. Cada uno de estos Tags es un numero que relaciona 1 a 1 la dificultad con las horas para su realización, es decir, una dificultad de ocho equivale a aproximadamente 8 horas de trabajo. Como es lógico cuanta mayor es la dificultad mas se rompe esta relación, puesto que al ser mas complicado siempre suele alargarse el tiempo de realización de la tarea. El proyecto se desarrollo en los siguientes Sprints:

Sprint 1: Investigación y familiarización (Software, Hardware y Herramientas)

Sprint 2: Envío y Recibo de paquetes TCP/IP

Sprint 3: Investigación y primeros pasos con periféricos (Comunicación uart, I2C)

Sprint 4: Investigación sobre Wifi. Uso de potenciómetros para gestionar motores

Sprint 5: Periféricos (sensor de temperatura y pantalla) y uso real en una empresa

Sprint 6: Memoria, presentación y últimos Retos

En los siguientes apartados obtendremos mas información sobre como fue el desarrollo de cada uno de estos Sprints.

Sprint 1

Este Sprint esta compuesto por 10 tareas que en total suman 43 puntos. Duró aproximadamente dos semanas. El comienzo fue complicado por la gran cantidad de conceptos y funcionamientos del hardware que debí investigar, además de la dificultad de familiarizarse con dos IDEs diferentes. Las acciones principales fueron:

Elección, descarga y familiarización del entorno de desarrollo

Completar las practicas propuestas por los Cotutores

Descarga de otros programas para documentación, organización y comunicación

Primeros pasos con lwIP

Primeros pasos con RTOS

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este apéndice se exponen los requisitos del software creado y la funcionalidad de este. Para ello se da una descripción completa de las funciones que debe realizar el programa y de como un usuario podría utilizarlo. Se definen los pasos y opciones para utilizar correctamente el hardware del sistema empujado y que cumpla así, con su función. Por otro lado, se definen también se definen los requisitos no funcionales. Para la exposición de las siguientes secciones se utilizaran, por motivo de simplicidad, algunas abreviaciones:

Sistema empujado o embebido: SE

Software del sistema empujado: SW del SE

Requisito de interfaces externas: RE

Requisito funcional: RF

Requisito no funcional: RNF

Casos de Uso: CU

Objetivos generales

En esta sección vamos a ver el funcionamiento del SE:

-

Clases de usuario y características

Para la utilización del sistema empotrado no se necesita ningún tipo de conocimiento técnico por lo que cualquier persona interesada puede usarlo sin problemas. La mayor dificultad radica en haber conectado todos los componentes en red y conocer los casos de uso y los pasos para utilizar la placa maestro. Ya que junto con el software se proporciona la memoria que describe el desarrollo, además de varios apéndices con información complementaria el usuario podrá ser conocedor del correcto funcionamiento del SE. En concreto, los usuarios pueden consultar el apéndice con el manual de usuario que cuenta con indicaciones para el uso del software.

B.2. Catalogo de requisitos

Requisitos externos

- **RE-2 Acceso a la red** El SE tiene que ser capaz de usar Ethernet. RE de alta prioridad.
- **RE-3 Transmisiones en red** El SE tiene que ser capaz de usar TCP/IP. RE de alta prioridad.
- **RE-3 Uso de Botones** Los seis botones que utiliza la placa mas el shield deben poderse utilizar correctamente. RE de alta prioridad.
- **RE-4 Uso de dispositivos ADC** Tanto el sensor de temperatura como los potenciómetros deben realizar adecuadamente sus lecturas. RE de alta prioridad
- **RE-5 Comunicaciones con periféricos** Las comunicaciones I2C y uart se deben de poder utilizar RE de alta prioridad.

Requisitos Funcionales

- **RF-1 Recepción de comandos** El SE tiene que recibir comandos transmitidos mediante paquetes TCP con destino a su correspondiente dirección IP y puerto TCP abierto. RF de alta prioridad.
- **RF-2 Identificación de comandos** El SE tiene que se capaz de identificar los comandos recibidos para poder ejecutar las acciones correctas. RF de alta prioridad

- **RF-3 Movimiento de los motores**
- **RF-4 Obtención de la velocidad de los motores**
- **RF-5 Obtención de la temperatura**
- **RF-6 Envío de comandos**
- **RF-7 Parada de ambos motores**

Requisitos No Funcionales

- **RNF-1 hardware del SE** El SE tiene que ser desarrollado con una placa de desarrollo FRDM-K64F. RNF de alta prioridad.
- **RNF-2 Rendimiento del SE** El SE tiene que ser capaz de realizar las acciones indicadas por el usuario sin demora. RNF de media prioridad.
- **RNF-3 Seguridad del SE** El SE tiene que asegurar que sus componentes no presentan riesgos eléctricos al usuario. RNF de alta prioridad.
- **RNF-4 Calidad del SW** El SW tiene que garantizar cierto nivel de calidad, tales como incluir comentarios que faciliten su comprensión para un mantenimiento o portabilidad posteriores. RNF de media prioridad.
- **RNF-5 Usabilidad del SE** La utilización del sistema
-

B.3. Especificación de requisitos

Diagrama de Casos de Uso

Casos de Uso

CU-1	Seleccionar SE
Versión	1.0
Fecha	2022-06
Requisitos asociados	RF-3
Descripción	Fijar la velocidad del Motor A
Precondición	Se debe tener conexión en red entre las placas Todos los elementos del sistema deben estar conectados
Acciones	<ol style="list-style-type: none"> 1. El usuario ajusta el potenciómetro 1 según sus necesidades. 2. El usuario pulsa el botón 1 para enviar el comando. 3. Se corrobora mediante la pantalla LCD y los leds de la placa maestro que se ha enviado el comando.
Postcondición	El motor comienza a moverse
Excepciones	<ul style="list-style-type: none"> ■ Si alguno de los elementos del sistema no esta bien conectado o no se dispone de corriente no funcionará ■ La lectura del potenciómetro puede no realizarse bien en algunas ocasiones.
Importancia	Alta
Comentarios	Los cuatro leds de la placa shield se iran encendiendo a medida que avance el envio del comando

Tabla B.1: CU-1 Seleccionar SE

CU-2	Seleccionar SE
Versión	1.0
Fecha	2022-06
Requisitos asociados	RF-3
Descripción	Fijar la velocidad del Motor B
Precondición	Se debe tener conexión en red entre las placas Todos los elementos del sistema deben estar conectados
Acciones	<ol style="list-style-type: none"> 1. El usuario ajusta el potenciómetro 2 según sus necesidades. 2. El usuario pulsa el botón 2 para enviar el comando. 3. Se corrobora mediante la pantalla LCD y los leds de la placa maestro que se ha enviado el comando.
Postcondición	El motor comienza a moverse
Excepciones	<ul style="list-style-type: none"> ■ Si alguno de los elementos del sistema no esta bien conectado o no se dispone de corriente no funcionará ■ La lectura del potenciómetro puede no realizarse bien en algunas ocasiones.
Importancia	Alta
Comentarios	Los cuatro leds de la placa shield se iran encendiendo a medida que avance el envio del comando

Tabla B.2: CU-2 Seleccionar SE

CU-3	Seleccionar SE
Versión	1.0
Fecha	2022-06
Requisitos asociados	RF-7
Descripción	Parada de emergencia de ambos motores
Precondición	Se debe tener conexión en red entre las placas Todos los elementos del sistema deben estar conectados
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa el botón 3 para enviar el comando. 2. Se corrobora mediante la pantalla LCD y los leds de la placa maestro que se ha enviado el comando.
Postcondición	Ambos motores se paran
Excepciones	<ul style="list-style-type: none"> ■ Si alguno de los elementos del sistema no está bien conectado o no se dispone de corriente no funcionará ■ La lectura del potenciómetro puede no realizarse bien en algunas ocasiones.
Importancia	Alta
Comentarios	Los cuatro leds de la placa shield se encenderán a la vez

Tabla B.3: CU-3 Seleccionar SE

CU-4	Seleccionar SE
Versión	1.0
Fecha	2022-06
Requisitos asociados	RF-5
Descripción	El usuario solicita la temperatura ambiente
Precondición	Se debe tener conexión en red entre las placas Todos los elementos del sistema deben estar conectados
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa el botón 4. 2. Se corrobora mediante la pantalla LCD y los leds de la placa maestro que se ha enviado el comando.
Postcondición	La temperatura se muestra en la pantalla de la placa esclavo 1
Excepciones	<ul style="list-style-type: none"> ■ Si alguno de los elementos del sistema no está bien conectado o no se dispone de corriente no funcionará ■ Las dos primeras lectura del sensor de temperatura son de calibración.
Importancia	Alta
Comentarios	Los cuatro leds de la placa shield se encenderán a la vez

Tabla B.4: CU-4 Seleccionar SE

CU-5	Seleccionar SE
Versión	1.0
Fecha	2022-06
Requisitos asociados	RF-4
Descripción	Obtener velocidad del Motor A
Precondición	Se debe tener conexión en red entre las placas Todos los elementos del sistema deben estar conectados
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa el botón 5 para enviar el comando. 2. Se corrobora mediante la pantalla LCD y los leds de la placa maestro que se ha enviado el comando.
Postcondición	Se muestra por la pantalla LCD la velocidad del motor A
Excepciones	<ul style="list-style-type: none"> ■ Si alguno de los elementos del sistema no está bien conectado o no se dispone de corriente no funcionará
Importancia	Alta
Comentarios	Los cuatro leds de la placa shield se encenderán al mismo tiempo

Tabla B.5: CU-5 Seleccionar SE

CU-6	Seleccionar SE
Versión	1.0
Fecha	2022-06
Requisitos asociados	RF-4
Descripción	Obtener velocidad del Motor B
Precondición	Se debe tener conexión en red entre las placas Todos los elementos del sistema deben estar conectados
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa el botón 6 para enviar el comando. 2. Se corrobora mediante la pantalla LCD y los leds de la placa maestro que se ha enviado el comando.
Postcondición	Se muestra por la pantalla LCD la velocidad del motor B
Excepciones	<ul style="list-style-type: none"> ■ Si alguno de los elementos del sistema no está bien conectado o no se dispone de corriente no funcionará
Importancia	Alta
Comentarios	Los cuatro leds de la placa shield se encenderán al mismo tiempo

Tabla B.6: CU-6 Seleccionar SE

Apéndice C

Especificación de diseño

C.1. Introducción

En el apéndice anterior se detallaron los requisitos del software. El siguiente paso es mostrar las especificaciones de diseño, en este apéndice se explicarán los motivos para las soluciones tomadas en relación al diseño de este programa. El diseño del software recoge las cuestiones relativas a la gestión de los datos y su presentación, a la división de las partes del código en otros módulos y como se estructura su funcionamiento. En definitiva se expone como se estructura el software arquitectónicamente.

Ámbito del software

En este proyecto el usuario interactúa con la placa shield para enviar distintas comandos a los otros sistemas empuotrados.

C.2. Diseño de datos

La base del proyecto es la comunicación entre sistemas empuotrados. En esta comunicación se envían comandos que serán recibidos y gestionados por otros sistemas embebidos. Los comandos son cadenas de caracteres que se dividirán y pasarán por diferentes filtros hasta llegar a la acción que el SE debe realizar. Veamos el proceso en más detalle.

Transferencia de los comandos.

///los comandos nunca van a fallar porque se configuran directamente mediante la programación La comunicación entre placas para enviar y recibir comandos se implementa siguiendo el modelo TCP Transmission Control Protocol. Este modelo se encarga de realizar la conexión estableciendo la comunicación. Para realizar esta labor enviara unos comandos de sincronización tal y como veremos en la siguiente figura:

Figura C.1: T

ras ejecutar este proceso se envían los comandos encapsulados en los segmentos TCP.

Comando setSpeed

///////////////// Sería mejor explicar el funcionamiento de los comandos etc y después hacer una tabla con todos los tipos de comandos y por último explicar alguna cosa especifica de cada uno ////////////////// Este comando indica a la velocidad que deben girar los motores. Para ello el usuario deberá haber configurado mediante el potenciómetro la velocidad y pulsar el botón correspondiente para el envío de la velocidad al motor determinado. El comando se enviará en el siguiente formato: Uart/mot1/255/ Cuando este comando llega al correspondiente sistema embebido se descompone en pequeñas cadenas de caracteres que están separadas por barras '/'. La primera cadena UART siempre es así en caso de que queramos comunicarnos con los motores. La segunda 'mot1' indicará de que motor queremos cambiar la velocidad y podrá variar entre mot1 y mot2. Por último la tercera subcadena 255 será la velocidad a la que queremos que funcione el motor, que en nuestro caso podrá variar entre 255, 128 o 0. //Para setear estas velocidades también deberemos indicar el modo en el que deben trabajar los motores en este caso el comando seria: uart/Mode/0/ Sub comando Parada emergencia Sub comando Conocer temperatura Sub comando obtener velocidad

C.3. diseño arquitectónico

Que es el diseño arquitectónico de un software

diseño arquitectónico del SE

Una de las grandes cualidades de los sistemas empotrados es que se pueden relacionar con el entorno mediante sensores. Esto hace que el software se pueda reajustar según los datos obtenidos por estos sensores. En sistemas empotrados simples, es decir, aquellos que no son en tiempo real el software su organización es más simple. En este caso como ya sabemos la gestión de las tareas se realiza en tiempo real por lo que la organización de estas es más compleja. Se deben asignar prioridades y es necesario utilizar un software que nos ayude en esta gestión, en este caso hemos utilizado FreeRtos En un SE sin restricciones podríamos utilizar cualquier otro sistema de gestión de tareas, cíclico, round robin, etc. En el primer ejemplo que he nombrado, cíclico, las tareas se ejecutan en bucle una detrás de la otra y una no puede empezar hasta que las anteriores hayan terminado. Cuando se ejecuta la última tarea se vuelve al principio. Además en este tipo de implementación podemos añadir un round robin (quantum) /es lo mismo?/ para que cada tarea tenga una duración máxima de tiempo de forma que el programa no se quede colgado, por otro lado también tiene el problema de que puede ser que algunas tareas no les dé tiempo a terminar de ejecutarse. Una buena práctica para este tipo de sistemas es la utilización de interrupciones, así evitamos que el programa tenga que realizar escaneos del entorno constantemente. Las interrupciones se pueden programar para activarse al pulsar un botón o al detectar algún cambio en el entorno mediante los sensores. En ese momento se activa la interrupción y se pone en marcha la tarea correspondiente. Vistas este tipo de arquitecturas como ya comentamos anteriormente se va a utilizar un sistema operativo en tiempo real para la gestión de las tareas. Esta decisión se basa en el bajo tiempo de respuesta que proporciona el SO. Mediante las prioridades elegimos además que tarea debe ejecutarse antes que otras aportando flexibilidad a la ejecución del programa. /fotos/

Mediante la programación de estas tareas se da respuesta a todos los casos de uso planteados en el apéndice anterior. //ampliar

Diseño del uso de la placa

???????

Apéndice D

Documentación técnica de programación

D.1. Introducción

Este apéndice muestra las herramientas necesarias para entender y poder reutilizar el software presentado. Por supuesto no es necesario utilizar las mismas herramientas. Para el desarrollo del trabajo se ha utilizado MCUXpresso. A continuación se muestra como instalar y configurar este IDE. Además también veremos otras herramientas que nos ayudaran a comprobar el correcto funcionamiento del software.

D.2. Estructura de directorios

La estructura de directorios del software es la siguiente:

Figura D.1: Estructura de directorios

- / Directorio raíz, contiene el resto de los directorios. Se incluyen ficheros como LICENSE que contiene los términos y condiciones del licenciamiento del software y el fichero MEX que contiene los datos de las configuraciones de los pines, relojes y periféricos. Este último fichero lo genera el propio IDE.

- /CMSIS/** Cortex Microcontroller Software Interface Standard (CMSIS). Reúne las fuentes que proporcionan interfaces al procesador y sus periféricos.
- /amazon-freertos/** Fuentes relacionadas con FreeRTOS, sistema operativo en tiempo real usado en el proyecto.
- /board/** Fuentes autogenerados por el IDE que permiten habilitar y configurar el hardware de la placa de desarrollo.
- /doc/** Documentación del proyecto.
- /drivers/** Controladores necesarios para trabajar con el hardware.
- /lwip/** Fuentes relativos a lwIP, la implementación de la pila de protocolos TCP/IP.
- /source/** Código fuente del proyecto. El fichero main.c es el que contiene el código del funcionamiento de las placas. Además se encuentran los ficheros que agrupan herramientas para `////`
- /startup/** Código de arranque generado por el IDE.
- /utilities/** Código generado por el IDE con utilidades auxiliares usadas para depuración o registro de eventos.

D.3. Manual del programador

Descarga e instalación de MCUXpresso

En primer lugar será necesaria descargar el IDE desde la página oficial de Nxp. Este software estará en la pestaña de desarrolladores. Para poder descargarlo será necesaria tener una cuenta de NXP, la cual es gratuita y te puedes registrar fácilmente desde el sitio web. Dicho esto, iniciamos sesión en la página vamos a la pestaña en la que se encuentra el software y pinchamos en descargar. Una vez hecho esto, elegimos el sistema operativo donde se ejecutará nuestro IDE. El instalador sigue los pasos habituales en este tipo de instalaciones, aceptar la licencia, elegir la ubicación donde se guardarán los archivos y controladores de programa, etc. Podemos dejar todas estas opciones por defecto o variarlas a nuestro gusto. Al descargar el IDE viene con él la herramienta Config Tools que nos ayudara hacer la configuración a bajo nivel de la placa. Una vez tenemos instalado MCUXpresso, lo abrimos. Lo primero que nos pide es elegir una ruta donde guardar nuestros proyectos.

Es recomendable que sea una ruta fácilmente accesible pues nos será de gran ayuda poder llegar rápidamente y poder importar de manera más sencilla los proyectos que necesitemos.

Figura D.2:

Descarga e instalación del SDK

Tras esta instalación del IDE, es importante la realización de este segundo paso en el que descargaremos el SDK. Podemos descargarlo también desde la web de NXP, en esta pestaña deberemos elegir el SDK correspondiente a la placa y la versión, que deberá ser superior a la 2.0. La descarga del SDK nos permite construir proyectos específicos para nuestra placa, permite además añadir los componentes necesarios según las funcionalidades del SE. Permite descargar e instalar drivers, el tipo de sistema operativo o configurar el middleware. En nuestro caso deberemos seleccionar como mínimo el sistema operativo FreeRTOS y como drivers lwIP y ADC para la medición del sensor de temperatura y la lectura de los potenciómetros. Para poder elegir estos add-ons deberemos clicar en esta parte:

Figura D.3:

y Se abrirá un interfaz como esta:

Donde podremos elegir los componentes que queramos.

Configuración de pines, relojes y periféricos

La configuración a bajo nivel de la placa es una de las partes más importantes y a la vez más complejas de entender al principio, por ello voy a explicar brevemente cómo funcionan estas interfaces del ide

pines En cuanto a los pines, estas placas disponen de 100 configuraciones para los pines permitiendo así la integración de varios periféricos aumentando su funcionalidad. En la siguiente figura podemos observar la interfaz para configurarlo:

Figura D.4:

En la imagen se pueden ver tanto los pines disponibles, como los pines que ya hemos configurado y una imagen del microcontrolador con todos los pines. Como habréis podido observar cada uno de los pines de la placa tiene varias configuraciones que podemos utilizar según nuestras necesidades. Para configurar un pin buscaremos uno que cumpla con el uso que queremos, lo clicamos y pinchamos en el tipo de periférico que vamos a utilizar

relojes En el caso de este software se ha utilizado siempre el reloj de general de la placa pero podemos configurar más relojes dependiendo del objetivo del periférico que vaya a usarlo. Veamos una imagen de la interfaz.

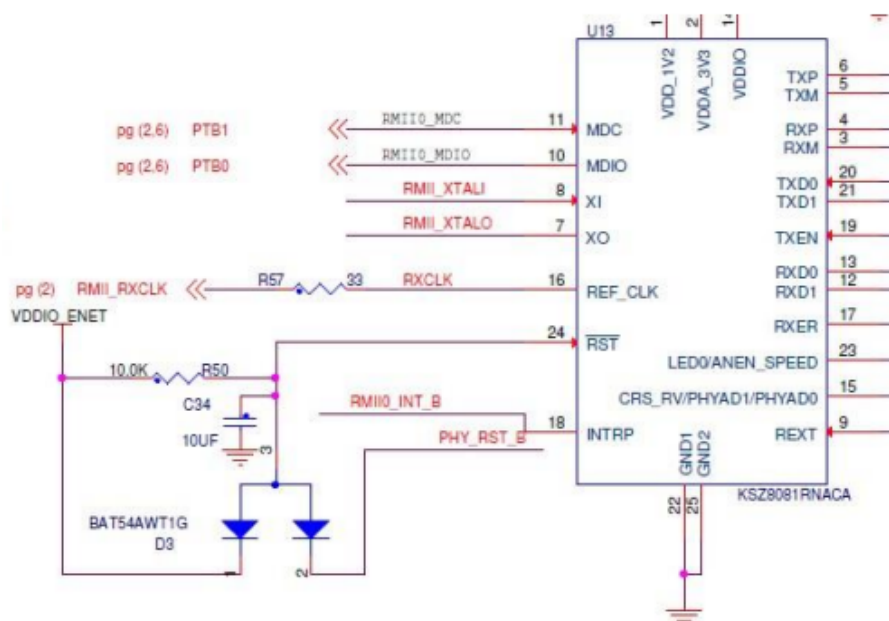


Figura D.5:

periféricos el microcontrolador permite conectar varios periféricos a la placa al mismo tiempo. En la siguiente figura veremos cuales son:

Figura D.6:

Como se puede apreciar hay algunos repetidos como por ejemplo Uart puesto que esta placa permite tener hasta 4 comunicaciones Uart configuradas.

D.4. **Compilación, instalación y ejecución del proyecto**

En esta sección vamos a ver como compilar y ejecutar un proyecto. Para ello lo primero que vamos a hacer es coger un proyecto de ejemplo del SDK. Vamos a la pestaña de la imagen y seleccionamos ‘////////’

Figura D.7: U

na vez tenemos el proyecto vamos a la carpeta sources y vemos que fichero tiene los datos fuente del proyecto. Para poder compilar clicaremos en la opción build para ver si el proyecto compila adecuadamente o tiene errores. Posteriormente deberemos colocar por usb la placa FRDMK64F al ordenador. Antes de seguir con el debugueo es importante que la placa este en modo OpenSDA, para ello al conectar la placa al ordenador deberemos estar pulsando el botón reset mientras lo hacemos. Tras esto se debería de abrir una carpeta llamada bootloader donde deberemos copiar el fichero openSDA. Desconectamos y conectamos la placa y ya podremos ejecutar la opción debug desde el IDE esto comenzara la ejecución de nuestro ejemplo. Una vez comience la ejecución se abrirá también una consola y podremos utilizar el SE sin problemas.

rpc Con los entornos de desarrollo y sus respectivos proyectos preparados es posible compilar los códigos fuente. Dependiendo del software a ejecutar es necesario tomar diferentes caminos para su puesta en marcha. Compilación, escritura y ejecución del sistema empotrado Existen varias vías para compilar el código fuente. Una de ellas es hacer clic derecho sobre el proyecto y en el menú contextual, pulsar sobre “Build Project”. Otra forma es pulsar sobre su icono correspondiente en la barra de herramientas. Para realizar la escritura, o flash, de los binarios en el sistema empotrado y que pueda ejecutarlos hay que lanzar desde el IDE la operación de “Debug”. Igual que la compilación,

se puede hacer desde el menú contextual o desde la barra de herramientas. Cabe decir que la operación de depuración ejecuta automáticamente la de compilación, haciendo innecesario tener que ordenarla manualmente. La primera vez que se lanza un debug el IDE solicita la identificación de la placa de desarrollo. Para identificar la placa de desarrollo hay que especificar el uso de “SEGGER J-Link probes”, compatibles con OpenSDA, el adaptador serie y de depuración integrado en la placa

Pruebas del sistema: Packet Sender

Otra herramienta que puede ayudar enormemente a los desarrolladores es packet sender. Para descargarla tendremos que ir a su página web oficial y descargar la herramienta para el sistema operativo donde vayamos a utilizarla. Packet Sender permite enviar paquetes por el protocolo tcp, a una ip y un puerto determinados. Además se pueden guardar nuestros propios comandos de envío para reutilizarlos de forma más sencilla. De esta forma conseguimos poder comprobar si las placas reciben los comandos adecuadamente y cómo se comportan al recibirlo. Es como tener otra placa en red pero los comandos se envían de forma más sencilla desde nuestro propio ordenador

Figura D.8: Interfaz de la herramienta Packet Sender

Apéndice E

Documentación de usuario

E.1. Introducción

Esta sección recoge las instrucciones y conocimientos técnicos que un usuario debe conocer para poder utilizar el laboratorio presentado. Veremos como instalar el software en nuestra placa y como controlar la placa maestro. Se explicaran que funciones realizan cada boton y cuales son los usos que tiene el laboratorio.

E.2. Requisitos de usuarios

Para poder utilizar el programa lo primero es disponer de todas las piezas que componen el laboratorio. Deberemos comprobar que todos los componentes están correctamente conectados entre si. Para ello deberemos:

1. Disponer de tres cables de red y conectar cada uno de ellos a una placa y a un switch.
2. Comprobar que los motores están correctamente conectados a la placa controladora y que las placas FRDM k64F también están conectadas a la placa controladora de los motores. Es importante cerciorarnos de que todos los pines están debidamente conectados a su posición correspondiente
3. Tendremos que asegurarnos que todos los componentes están conectados a una toma de corriente y reciben energía, por lo general si están recibiendo corriente tendrán algún led encendido.

4. También deberemos comprobar que el switch cumple con su función mirando si todos los puertos donde se conectan los cables de red están recibiendo y enviando datos
5. Se deberá conectar la placa de expansion a la placa maestro, en este caso es muy sencillo porque solo existe una forma de conectarla para que todos los pines queden conectados.
6. Por ultimo deberemos conectar los pines de la pantalla lcd a una de las placas maestro para poder ver de manera adecuada la información de los comandos recibidos.

Tras tener configurado el entorno el siguiente paso sera cargar el programa correspondiente a cada microcontrolador. Recordemos que disponemos de tres placas y tres softwares con algunas diferencias entre ellos. En el apéndice anterior ya vimos como podíamos realizar este procedimiento. Una vez se tienen los códigos cargados en las placas deberemos esperar a que los tres microcontroladores adquieran su respectiva dirección ip y podremos comenzar a enviar comandos desde la placa maestro como veremos en el siguiente apartado.

E.3. Manual del usuario

En el apéndice /// pudimos ver en los requisitos funcionales todos los usos de la placa y sus correspondientes pasos. En este apartado veremos de una forma mas gráfica las utilidades de la misma.

En primer lugar vamos a visualizar y nombrar los botones de la placa maestro.

Para configurar la velocidad del motor A deberemos mover el potenciómetro de la placa según la velocidad (sentido del giro) que le queramos. Si lo giramos hacia la derecha completamente el motor recibirá el valor 255 y girará hacia la derecha, si por el contrario lo giramos hacia la izquierda recibirá el valor de 0 y girara en el sentido opuesto. En caso de que dejemos el potenciómetro en un rango medio de su giro, el motor recibirá el dígito 128 y dejara el motor parado.

Lo mismo pasa con el potenciómetro 2 y el botón 2. En ambos casos, al pulsar el boton 1 o dos, recibiremos un feedback en la placa esclavo mediante un mensaje en la pantalla que nos mostrara si se recibió el comando o no.

En caso de que queramos parar ambos motores utilizaremos el botón 3 a modo de parada de emergencia. El botón 4 envía a la placa maestro la temperatura ambiente. Esta temperatura se muestra en °C por la pantalla del esclavo.

Por ultimo los botones 5 y 6 muestran la velocidad de los motores A y B respectivamente, esta información se mostrara por la pantalla de la placa esclavo. Es importante conocer que en muchos casos las primeras lecturas tanto de los potenciometros como del sensor de temperatura no le ha dado tiempo a leer la nueva posición y se debe repetir la acción para mandar el comando.

/////////////////FOTOS