

Simulación de disparo interdrónico en el entorno Game of Drones

Iván Yera Prados

Abstract—Recently, the drone industry has experienced a rapid growth and its business value has increased profits in many sectors such as military, audiovisual or package transport. Even in leisure, drone racing has become a business and has served as a precursor to the use of quadcopters in new games. In this context comes Game of Drones, based on the well-known Capture the Flag, where two teams compete for the control of bases in aerial battles demolishing the adversaries. The aim of this project is giving a technological solution to shooting simulation between different drones, which should be suitable for playing the game both indoor and outdoor environments.

Index Terms—Drones, IrDA, STM32F0, FreeRTOS

Resumen—En los últimos años, la industria de los drones ha experimentado un rápido crecimiento y su valor de negocio ha hecho aumentar los beneficios en muchos sectores como el militar, el audiovisual o el transporte de paquetes. Incluso en el ocio, las carreras de drones se han convertido en negocio y han servido de precursor para la utilización de cuadricópteros en nuevos juegos. En este contexto surge Game of Drones, basado en los conocidos juegos de toma de bandera, donde dos equipos compiten por el control de bases en batallas aéreas derribando a los adversarios. En este proyecto se pretende dar una solución tecnológica a la simulación de disparos entre los diferentes drones, que debe ser apta para la práctica del juego tanto en entornos interiores como en exteriores.

Palabras clave—Drones, IrDA, STM32F0, FreeRTOS



1 INTRODUCCIÓN

En el siglo pasado, se concibió por primera vez el término dron para designar a un vehículo aéreo motorizado que no lleva a bordo a un operador humano. La anterior definición resulta poco precisa si se tiene presente la gran cantidad de aeronaves que existen, pero es útil para denominar de manera genérica a los populares cuadricópteros que se comercializan actualmente.

Los cuadricópteros son helicópteros multi-rotor formados por cuatro brazos con un motor y una hélice en su extremo. A diferencia de los helicópteros convencionales, la estabilidad en su flotación no se consigue mediante un rotor de cola sino haciendo girar dos hélices en sentido horario y las otras dos en sentido antihorario.

Gracias a los avances tecnológicos y a la reducción de costes, muchas industrias, como la militar o la audiovisual, han visto en los drones una oportunidad de innovación para solucionar problemas existentes.

De hecho, el bajo coste de la producción de cuadricópteros y su gran demanda por los consumidores han contribuido a su comercialización y la posterior profesionalización de usuarios que compiten en carreras de drones.

En las carreras de drones, los competidores diseñan y montan sus cuadricópteros, que siempre incorporan una cámara y un transmisor de vídeo que envía los datos al receptor incorporado en unas gafas FLV. Gracias a esa tecnología son capaces de controlar los drones en primera persona mientras superan circuitos con banderas y obstáculos.

Anualmente, se celebra un campeonato en Dubai, el World Drone Prix, donde los mejores pilotos se ponen a prueba con sus drones para conseguir el preciado premio de 250.000 dólares. Con estas cifras, se observa como este eSport coge fuerza año tras año y empieza incluso a televisarse.

Gracias a la popularización de este deporte, surge la idea de introducir una nueva modalidad en el mercado, Game of Drones.

-
- E-mail de contacto: Ivan.Yera@e-campus.uab.cat
 - Titulación: Grado en Ingeniería Electrónica de Telecomunicación y Grado en Ingeniería Informática (Mención en Ingeniería de Computadores)
 - Tutor del Proyecto: Màrius Montón.
 - Curso 2017/2018

2 OBJETIVO Y REQUISITOS

Game of Drones es una nueva modalidad de juego en la que dos equipos de drones compiten por el control de bases, capturándolas al mantenerse en su interior durante un periodo de tiempo establecido. Para impedir la intrusión de sus enemigos, los drones incorporarán un mecanismo de disparo que permita dejar fuera de juego a los drones del equipo contrario, que a efectos prácticos será un envío de información de un dron a otro en línea recta.

El objetivo principal de este proyecto es el diseño, implementación y gestión de una transmisión de datos que simule el lanzamiento de un proyectil y sea capaz de incluir información sobre el dron atacante. Además, será necesaria la correcta integración de esta funcionalidad en el sistema completo, como se puede observar en el diagrama secuencial de la Fig. 1 del Anexo

Una vez que el mando emita la orden de disparo, el mensaje debe ser redirigido al dron atacante, que efectuará la transmisión lineal de datos. El dron receptor comunicará al servidor central encargado de la gestión de juego que ha recibido un disparo y el servidor gestionará todas las consecuencias asociadas.

Durante las reuniones iniciales se establecen unos requisitos mínimos a nivel de proyecto con el resto del equipo, además de dos objetivos adicionales en caso de seguir correctamente la planificación establecida.

2.1 Tiempo de latencia

Para garantizar una buena experiencia de juego, es necesario un tiempo de latencia parecido al de los actuales *shooters* de la industria. El tiempo de latencia es el tiempo que tarda en transmitirse un paquete por el sistema y es un factor clave en este tipo de juegos.

El máximo *ping* aceptable por el Product Owner es de 30ms desde que un jugador ordena la acción a través del mando hasta que el dron dispara. Con tiempos de latencia superiores la jugabilidad se vería afectada, provocando que los disparos no acertasen el blanco, debido a la increíble velocidad de los cuadricópteros.

2.2 Dirección del disparo

Para ser capaz de derribar a otros drones, es necesario un ataque direccional para aprovechar la visión proporcionada por el sistema FLV utilizado por los pilotos para controlar los cuadricópteros.

Esta característica permitirá definir y diferenciar esta modalidad de juego de las convencionales carreras, ya que además de esquivar todos los obstáculos que encuentren, incluyendo a otros drones, deberán tener en cuenta en todo momento la posición del enemigo y el equipo que controla cada base.

2.3 Información del disparo

Con el fin de que el servidor central controle la puntuación del juego, es necesario que la información enviada en cada disparo permita al receptor identificar al atacante y el arma que ha utilizado. De esta manera, el sistema podrá determinar el equipo que suma puntos por la baja y penalizar la movilidad del dron que recibe el ataque.

2.4 Opcional: Detección de dron inmóvil

Resulta prácticamente imposible acertar a un dron que permanece inmóvil sobre una superficie, ya que el dron atacante tendría que caer en picado sobre él o volar a ras de suelo. Es por este motivo que será necesario el control de los drones en el interior de las bases, para impedir que sean capaces de tomar su control manteniéndose quietos en el fondo de esta.

2.5 Opcional: Integración de componentes en PCB

Por cuestiones de estabilidad y espacio conviene reducir las dimensiones del dispositivo que se colocará sobre el dron para gestionar el disparo. Es por eso que se decide integrar los componentes en una placa de circuito impreso. Adicionalmente, permitirá disponer los transceptores infrarrojos lo suficientemente alejados entre sí para poder diferenciar la zona de impacto al recibir un paquete por IrDA.

3 ESTADO DEL ARTE

Para abordar el objetivo del proyecto, es fundamental indagar entre diferentes tecnologías inalámbricas para encontrar la que mejor se adapte a los requisitos especificados.

La primera opción consiste en una antena bidireccional con un transceptor de radiofrecuencia. Para asegurar que el disparo seguirá una trayectoria prácticamente recta, se analizan los mapas de RSSI de los diferentes módulos del mercado. Esta opción es descartada, ya que el ruido generado por los transceptores de todos los drones sumado al ruido electromagnético de ciertos ambientes donde se prevé que se llevará a cabo el juego, como fábricas, provocaría la pérdida de paquetes. Al tener que reenviarlos para constatar el disparo no se respetaría el tiempo de latencia.

Una segunda opción basada en un módulo ultrasonidos o de radar en cada dron permitiría detectar distancias entre drones y resultaría suficientemente direccional. El problema de esta tecnología es que el envío de datos resulta complicado al no disponer de protocolo de comunicación, por lo que sin envío de información resultaría complejo que el sistema central conociese los detalles de los participantes en el disparo.

Otra propuesta sería utilizar un sistema de posicionamiento para espacios reducidos como el Real Time Locating System (RTLS) para mapear todos los drones y bases a tiempo real. Esta opción resulta compleja y costosa, ya que requiere la instalación de balizas a poca distancia para garantizar la precisión. Cada una de estas balizas supone un aumento extra de las comunicaciones y, por lo tanto, interferencias en los mensajes, además de añadir coste y complejidad al sistema

La solución que mejor se adapta a los requisitos es, sin duda, la del módulo infrarrojos. Esta tecnología permite el envío direccional de datos y la frecuencia de la radiación infrarroja no interfiere en el resto de comunicaciones del sistema, al tener una longitud de onda mayor que la luz visible pero menor que las microondas. En concreto, se utilizarán infrarrojos cercanos con una longitud de onda de 890nm. Además, todos los protocolos permiten el envío de datos y pueden ser personalizados vía software.

En una primera instancia, se descubren los protocolos asociados a comandos. En concreto, los protocolo RC-5/RC-6 de Philips y el SIRC de Sony parecen los más extendidos, ya que han sido creados para enviar comandos a través de los mandos a distancia de sus televisores. El principal inconveniente radica en el tamaño de los paquetes, al ser insuficiente para el total de datos previstos para el envío, y el tiempo de latencia de cada uno de los paquetes es de 20ms, por lo que se hace imposible cumplir el requisito de tiempos.

Cuando parece que habrá que crear un protocolo propio para enviar información mediante rayos infrarrojos, se descubre el protocolo IrDA.

Se conoce como IrDA a un estándar físico de transmisión y recepción de datos mediante rayos infrarrojos. Su robustez ante señales de frecuencia de radio y su comunicación punto a punto hacen que esta tecnología sea idónea para este proyecto. Asimismo, los dispositivos que utilizan este protocolo son capaces de transmitir datos a unas decenas de metros de distancia. La velocidad de transmisión es propia de puerto serie, de hasta 4Mbps, representando los datos en un formato RZI (Retorno a Cero Invertido). Es por eso que para conseguir transmitir datos mediante este protocolo serán necesarios microcontroladores con encoders capaces de transformar la señal NRZ (Sin Retorno a Cero) al formato propio de IrDA (Fig. 1).

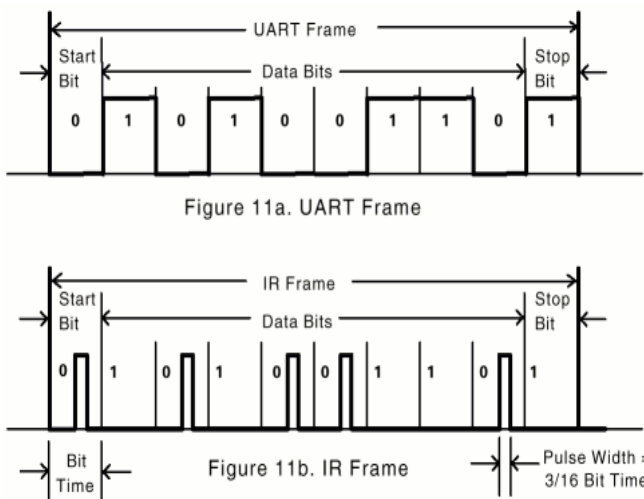


Fig.1: Comparación entre protocolo UART e IrDA

4 METODOLOGÍA

La metodología que se seguirá a lo largo de la realización del proyecto será SCRUM debido a algunas necesidades de este, como requisitos poco definidos o la intención de obtener resultados pronto.

Para la gestión ágil de este trabajo, se realizarán sprints de 2 semanas con reuniones periódicas de planificación del sprint, donde se estimarán los tiempos de desarrollo de tareas y los problemas obtenidos durante el anterior sprint. En estas reuniones será imprescindible la comunicación, no solo con el Product Owner y el Scrum Master, que guiarán las siguientes tareas individuales e introducirán nuevos requisitos, sino con el resto de integrantes del grupo. El papel de Product Owner recaerá sobre David Matanzas, CEO de Eurek, que propondrá requisitos y proporcionará soporte al proyecto en todo momento. Por otro lado, el papel de Scrum Master será llevado a cabo por Màrius, tutor del proyecto. Esto facilitará la integración final de todas las partes.

Se utilizará el software online Trello como tablero de tareas y el programa Microsoft Project para la estimación de tiempos de las tareas generales del proyecto.

Al acabar la ejecución de una tarea, y con el fin de garantizar el cumplimiento de los objetivos generales del trabajo, se reevaluarán los requisitos establecidos para adaptar los tiempos generales de las tareas y la posible introducción de mejoras/correcciones en el trabajo realizado hasta la fecha.

Por otra parte, se seguirá el estándar MISRA-C para la codificación y validación, utilizando Unity para el Unit Test y se utilizará Github como repositorio para el control de versiones.

5 PLANIFICACIÓN

En cualquier proyecto de ingeniería es fundamental distribuir las actividades a lo largo del tiempo, además de los recursos asociados, para garantizar la terminación en plazo del proyecto.

Una vez definidas las tareas, se debe estimar su duración y las relaciones entre ellas, para poder distribuirlas eficientemente. Este proceso se recoge en el diagrama de Gantt del Anexo 2, pero se puede observar la planificación de las tareas generales en la Fig. 2.

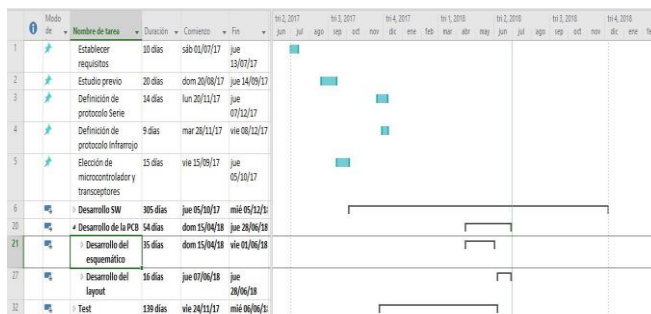


Fig 2: Planificación de tareas generales mediante Microsoft Project

6 DESARROLLO

6.1 Microcontrolador y creación del proyecto

Al indagar en el mercado sobre los microcontroladores con interfaces IrDA destacan los pertenecientes a las compañías ST, Texas Instruments o NXP. Debido a las facilidades proporcionadas por las placas núcleo de ST, se selecciona el STM32F051R8 para el proyecto y en especial la STM32F0DISCOVERY para el prototipo. Este microcontrolador dispone de dos UART, una de ellas apta para transmitir datos siguiendo el protocolo IrDA, y dos I2C que resultarán útiles para incorporar periféricos.

Utilizando la herramienta STMCubeMX, se seleccionan los periféricos que serán utilizados y se generan las librerías correspondientes, como se aprecia en la Fig. 3.

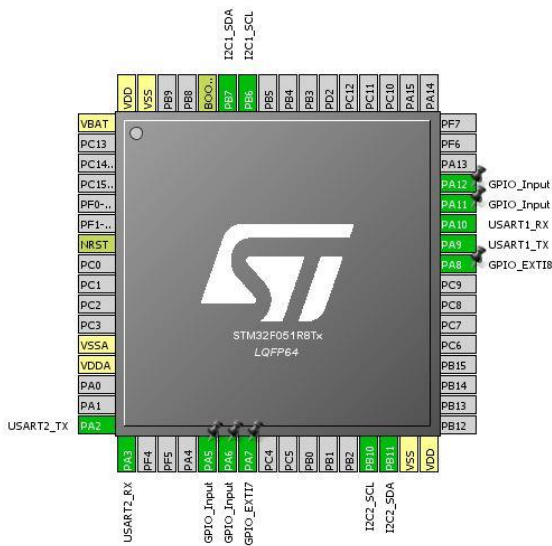


Fig. 3: STM32F051R8 con los periféricos utilizados en el proyecto

Una vez creado el proyecto en C, se estructura el código de forma modular para poder reaprovechar cada módulo en futuros proyectos, poder migrarlo a un microcontrolador de ST más potente en caso de que fuera necesario y poder probar la funcionalidad de cada módulo por separado.

6.2 GPIO

En todos los microcontroladores existen pines genéricos que pueden ser controlados por el usuario. Estos pines se conocen como GPIO, General Purpose Input/Output.

En el proyecto se utilizarán estos pines, observables en la Fig. 3, para comunicar los periféricos como el I2C y la UART y para gestionar los botones y los LEDs.

A nivel de implementación, se introducen todos estos pines en un mapping donde también se anotan sus características: el tipo, que mayoritariamente será de entrada o salida, la velocidad, el pull y el valor del pin, que puede ser de un '1' o un '0' lógicos.

6.3 FreeRTOS

Un sistema operativo no deja de ser un programa informático que gobierna las tareas de un ordenador a las que proporciona servicios. Además, los SO convencionales, también conocidos como sistemas operativos de propósito general, suelen proporcionar una interfaz gráfica de usuario (GUI) o al menos una consola para que el usuario final sea capaz de interactuar con el SO.

En los sistemas embebidos, las características de los microcontroladores y su necesidad de respuesta a eventos en un tiempo definido han provocado una nueva concepción de los SO. Los convencionales, de propósito general, han sido sustituidos por los sistemas operativos de tiempo real (RTOS) que son predictivos (por tanto, deterministas) permitiendo asignar prioridades a las tareas.

Entre ellos destaca FreeRTOS, pequeño, portable y de código abierto. Está distribuido bajo una licencia GPL no contaminante, lo que permite que el código del usuario pueda ser cerrado.

La herramienta STMCubeMX incluye librerías de FreeRTOS facilitando su uso. Una vez generado el proyecto, se configura el scheduler para ejecutar todas las tareas creadas, en función de la prioridad asignada.

Para verificar su correcto funcionamiento, se elaboran dos tareas para blinkear LEDs, aprovechando el mapeo de los GPIO anteriormente comentado. A cada una se le asigna una prioridad diferente, por lo que se observa que uno de los LEDs parpadea con mayor frecuencia.

A continuación, se generan tareas para la recepción y transmisión por puerto serie y por infrarrojos, parte fundamental de la aplicación del proyecto. Asimismo, se crean y gestionan dos semáforos para controlar la transmisión de datos en función de los mensajes recibidos.

6.4 Comunicación por puerto serie

En el caso de la comunicación por puerto serie, se utiliza una UART del microcontrolador. La UART, en inglés Universal Asynchronous Receiver-Transmitter, es un bus de comunicaciones asíncrono donde se transmite un dato por unidad de tiempo, secuencialmente, y sin ninguna señal de reloj externa.

Para este proyecto se configurarán los pines TX y RX de la UART2 (pines PA2 y PA3) como full-duplex, que permite la transmisión y recepción simultánea de información, aprovechando unas colas donde se almacenan los datos hasta que pueden ser procesados. También se parametrizarán el baud rate, o número de unidades de señal por unidad de tiempo, que en este caso será de 115,2Kbps, la longitud de palabra que será de 8 bits y el bit de paridad, que no será utilizado.

La recepción mediante UART hará uso de interrupciones, de manera que, al recibir un byte, la ISR (rutina de interrupciones) lo encolará en un buffer circular hasta encontrar el carácter de final de paquete. En ese momento, se analizará el buffer para parsear el mensaje y procesarlo. Gracias al uso de interrupciones se incrementa la eficiencia del microcontrolador, reduciendo el tiempo perdido y permitiendo el uso óptimo de diferentes periféricos.

Con el objetivo de comprobar su correcto funcionamiento, se utiliza un controlador puente USB a UART (CP2101) conectado a un ordenador. Mediante un cliente de puerto serie, se analizan la transmisión y la recepción de mensajes.

Una vez verificado, se implementa el protocolo acordado por todas las partes del proyecto para la transmisión de datos al sistema, representado en el diagrama secuencial del Anexo 1. Para poderlo implementar, durante el parseado del buffer de recepción se determina el tipo de mensaje a través del primer campo, conocido como tag. El tag determinará el número de campos y el tipo de tratamiento que ese mensaje tendrá que recibir. Los mensajes que serán tratados por el dron serán los siguientes:

1. CFIRE: Representa la orden de disparo que el mando emite. Incluye dos parámetros: el mando asociado y el arma con la que se ha disparado.
2. LED: Este mensaje contiene los LEDs que deberá encender el dron que recibe el disparo, en función de la zona donde se produce el impacto.
3. FIRE: Este mensaje es enviado por el dron que recibe el impacto y contiene el dron que ha disparado, el arma utilizada y el lado del dron afectado.

6.5 Comunicación por infrarrojos

La comunicación infrarroja se implementará aprovechando la interfaz IrDA de una de las UART del microcontrolador. Se trata de una transmisión half-duplex de datos que, al ser enviada a un transceptor de rayos infrarrojos, será modulada para su emisión por aire.

Esta comunicación será gestionada mediante interrupciones, de la misma manera que con el puerto serie, para poder liberar recursos del procesador.

Es necesario parametrizar la IrDA de manera análoga a la comunicación por puerto serie, y redirigir la rutina de interrupción de la UART a una creada específicamente para la interfaz infrarroja, que recibirá los datos y los almacenará en un buffer.

Para comprobar la funcionalidad del módulo infrarrojos, se conectan los pines TX y RX de la UART1 (PA9 y PA10, respectivamente) a sus homólogos en una placa de evaluación Altera. Esta placa contiene un transceptor de rayos infrarrojos Agilent HSDL-3201 por el que transmite y recibe información.

Para el caso de la comunicación infrarroja también se diseña un protocolo, basado en el utilizado para el puerto serie, con el objetivo de diferenciar la simulación de disparo de la toma de las bases. El protocolo acordado para la comunicación IrDA se puede apreciar en la Fig. 4 y Fig. 5, y los mensajes que serán tratados son:

1. SHOT: Representa el disparo entre drones. Incluye tres parámetros: el mando atacante, zona afectada por el impacto y arma con la que se ha disparado
2. BASE: Para la captura de la base se envía esta orden, incluyendo el mando que controla el dron.

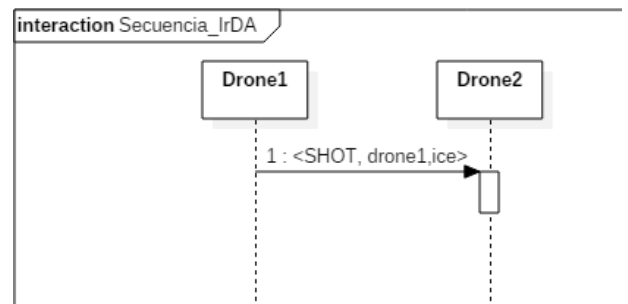


Fig. 4: Diagrama de secuencia asociado al disparo infrarrojo

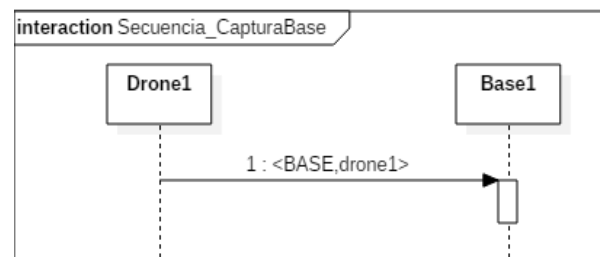


Fig. 5: Diagrama de secuencia asociado a la captura de base mediante infrarrojos

Una vez implementado el protocolo, se selecciona un transceptor que se adapte a las características del proyecto, en este caso el TFDU4101-TR3, que es capaz de transmitir a 115,2 kbps y emitir a 1 metro de distancia con un ángulo de transmisión y recepción de 15°. Se suelda en una placa de topos junto a la red de resistencias y condensadores especificada en su datasheet, como se aprecia en la Fig. 6.

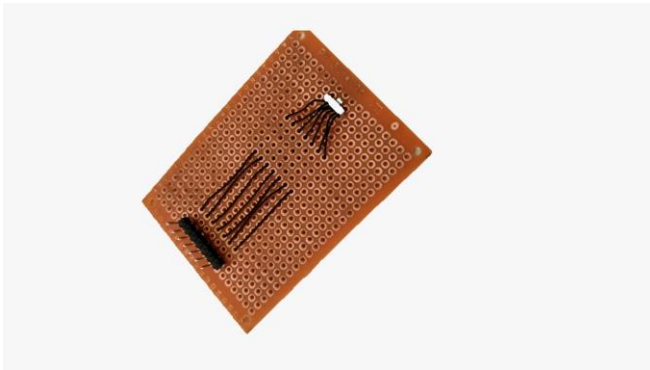


Fig.6: Transceptor TFDU4101-TR3 soldado en una placa de topos

6.6 Modo Bridge

Las funciones principales del proyecto son transmitir por infrarrojos, una vez procesados, los datos que se reciben a través de la UART para ejecutar el disparo, y por puerto serie los que se reciben por IrDA para comunicar la recepción de un disparo. Esto es conocido como modo Bridge.

En el momento en que se reciben datos por puerto serie, una rutina de interrupción (ISR) encola los datos en un buffer circular accesible para las tareas definidas en la aplicación. La tarea de recepción por UART espera a que el buffer previamente mencionado esté lleno para empezar a analizarlo. Para ello, hace uso de un parser que detecta el final de paquete definido en el protocolo y elimina el contenido restante. Una vez confirmado que el buffer solo contiene la información propia del protocolo, se analiza en busca del número de campos separados por comas. El primero de ellos se denomina Tag y determina el tratamiento del mensaje.

En caso de recibir el mensaje LED se encenderán los LEDs asociados a la zona donde se recibe el disparo. En caso de recibir el mensaje con el tag CFIRE se volcarán los datos sobre otro buffer y se liberará un semáforo para poder enviar este nuevo paquete utilizando el protocolo IrDA. Las tareas de transmisión de cada bus de comunicación están bloqueadas por semáforos que solo son liberados cuando se recibe el mensaje pertinente por el bus opuesto.

Por otra parte, al recibir un paquete por IrDA, la rutina de interrupción vuelca los datos sobre un buffer accesible para la tarea de recepción, a nivel de aplicación. Si la tarea de recepción detecta dicho bufer lleno, se utiliza el mismo parser anteriormente mencionado para analizar el mensaje buscando el carácter de final de paquete y los diferentes campos de este. Se utiliza el tag, también presente en los mensajes transmitidos por IrDA, para definir la posterior acción.

En caso de recibir el mensaje con tag SHOT se genera un buffer de transmisión con los datos incluidos en el paquete recibido, añadiendo la zona de impacto, detectada por el transceptor por el que se recibe, y el identificador del dron que ha recibido el mensaje. Además, se libera el semáforo asociado a la transmisión por puerto serie, por el que se enviará el nuevo paquete.

6.7 Acelerómetro

Para cumplir con el requisito 2.4 y detectar que un dron permanece quieto en el interior de la base, se utiliza un acelerómetro, en concreto el LIS3DH de ST, observable en la Fig. 7.

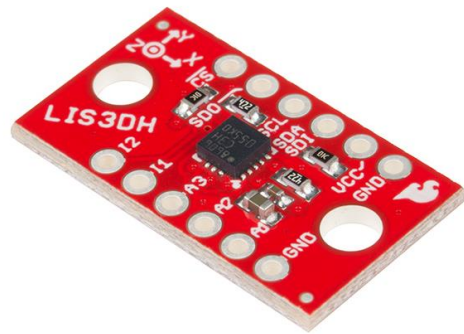


Fig. 7: Acelerómetro LIS3DH

Este acelerómetro dispone de una interfaz I2C por la que poder interactuar y extraer las aceleraciones en los 3 ejes. Es destacable que el LIS3DH compensa la fuerza de la gravedad, así que, si no se produce ningún movimiento, las 3 aceleraciones deberán ser nulas.

Cuando un cuadricóptero intenta conquistar una base introduciéndose en su interior, se produce una comunicación por IrDA entre el dron y la base. Al intentar capturar la base, se leen periódicamente los registros que contienen los valores de las aceleraciones del LIS3DH. Se genera un paquete donde se incluye un flag que indica si el cuadricóptero está quieto dentro de la base, analizando las aceleraciones en los 3 ejes. En caso de que todas marquen 0 durante unas determinadas iteraciones, se levantará el flag y el servidor anulará la toma de la base por parte de ese dron.

Para poder implementar esta funcionalidad, se configura un I2C del microcontrolador en modo master para poder escribir y leer en los registros accesibles del periférico en cuestión. En este caso, se escriben los registros necesarios para la configuración del acelerómetro como los ejes analizados, el threshold en mG, un filtro pasa altos para eliminar el offset y recibir valores reales y el modo low-power. También se configuran las interrupciones mediante GPIO (EXT17) para permitir detectar la caída libre del dispositivo. Al detectar la interrupción de GPIO, se analizan las aceleraciones de los 3 ejes para confirmar dicho estado.

6.8 Driver I2C to IrDA

El microcontrolador escogido para la realización del proyecto solo dispone de una UART apta para transmitir utilizando el protocolo IrDA. Esto permite comprobar la funcionalidad de la aplicación y el cumplimiento de los principales objetivos, pero limita a uno el número de transceptores conectados. Con el ánimo de cumplir el requisito 2.5, se utiliza el driver I2C implementado para el acelerómetro para controlar un dispositivo denominado SC16IS752 de la compañía NXP, que actúa como interfaz SPI/I2C a dos canales UART con compatibilidad con el protocolo IrDA. Este periférico permite mantener 4 transceptores conectados y funcionando simultáneamente, aprovechando dos pines del controlador, que actúan como un pin de CS (chip select) propio de SPI.

De manera análoga al acelerómetro, se configuran registros para la inicialización del dispositivo, incluyendo el baud rate, el bit de stop y el bit de paridad. También se habilita el protocolo IrDA y la recepción mediante interrupciones. Al recibir un mensaje, se levanta el pin de interrupción del dispositivo que será leído como un GPIO del microcontrolador. Al detectar la interrupción, la ISR del GPIO en cuestión, EXT18, analizará los registros pertinentes para descubrir el transceptor que ha recibido el paquete IrDA y leerá el buffer de recepción para poder tratarlo como un paquete del protocolo, utilizando el parser previamente comentado.

6.9 Implementación de la PCB

Una vez comprobado el funcionamiento del código sobre la placa núcleo, se decide elaborar una PCB con los periféricos previamente enumerados. Una PCB, en inglés Printed Circuit Board, consiste en una composición de capas de diferentes materiales laminadas conjuntamente con calor para formar un único bloque. El material base, o sustrato, suele fabricarse de fibra de vidrio, mayoritariamente FR-4. Esta capa es la que da forma y rigidez a la placa. Sobre la fibra se añade una fina película de cobre donde se imprimirán las pistas del circuito y, encima de esta, una máscara de soldadura para proteger las zonas de cobre donde no se quiere aplicar pasta de soldadura.

Actualmente todos los parámetros de una PCB son personalizables, desde el número de capas o el material de estas hasta las dimensiones de las vías. En nuestro caso se optará por una placa sencilla de dos capas para reducir los costes, ya que deberían ser suficientes para conseguir colocar y enrutar todos los componentes implicados en el proyecto, y un ancho de pista de 0,2mm.

6.9.1 Diseño del esquemático

Se utiliza la herramienta Kicad, y en concreto el editor de esquemas electrónicos, para incorporar todos los componentes en el esquemático. Se descargan los símbolos de la mayoría de componentes y se crean los que no se encuentran disponibles a través del editor de librerías. En ambos casos, se revisa el número de pines y el tipo de estos, confirmando que se ajustan a las especificaciones de los datasheets.

Para alimentar la PCB se opta por un conector USB del que obtendremos la señal de alimentación de 5V, que será gestionada por un regulador de voltaje que la transformará en los 3V necesarios para alimentar el microcontrolador y los periféricos. Esta señal de alimentación circula por una ferrita y un diodo TVS para filtrar interferencias electromagnéticas y transitorios de voltaje, respectivamente. Además, se incluye un circuito de reset, conectado al pin NRST del microcontrolador, que controla las bajadas de tensión y reinicia el microprocesador en caso de detectarlas. En el Anexo 3 se puede observar todo el circuito de alimentación.

En cuanto a los periféricos del microcontrolador, se implementa un pull-up en los buses I2C y se añade un header de 4 pines para poder programar la placa externamente.

El microcontrolador y todos los periféricos incluyen capacidades de desacoplamiento para reducir el ruido, además de redes de resistencias y condensadores en casos más concretos para ajustar el voltaje que recibe cada pin del componente en cuestión. El circuito asociado a la transmisión infrarroja se puede observar en el Anexo 4 y el asociado al microcontrolador en el Anexo 5.

Una vez colocados y cableados todos los componentes, como se observa en el Anexo 6, se realiza el control de reglas eléctricas para garantizar la coherencia en la conexión entre pines y, mediante la herramienta CvPCB, se asocian componentes y huellas, generando así la netlist con la que se procede a diseñar la PCB.

6.9.2 Diseño del layout

Una vez generada la netlist, se utiliza la herramienta PCBnew de Kicad para posicionar los diferentes componentes sobre la placa, cuya forma y medidas también se definen durante este proceso.

Tras colocar las huellas de los componentes sobre la placa, se realiza el routing, es decir, la unión de los pads de diferentes componentes mediante pistas. Aprovechando las dos capas de la PCB se realizan vías, donde la pista pasa a la capa inferior, para evitar cruzar otras pistas y así disponer de mayor flexibilidad a la hora de trazarlas.

Es importante que se utilicen ángulos de 45° para las pistas, ya que las puntas producidas los ángulos de 90° provocan ruido en señales de alta frecuencia y se obtiene un resultado más ordenado y estético, por lo que resulta más sencillo detectar errores en el layout.

También se añade un plano de masa para conectar todas las tierras de los componentes (pin GND) y evitar el trazo de las respectivas pistas. Por lo que respecta a la tensión de alimentación, se trazan todas las pistas desde la salida del regulador de voltaje hasta el pad de cada componente para garantizar el voltaje necesario.

Una vez acabado el diseño, observable el Anexo 7, se genera el gerber, seleccionando las 7 capas necesarias para la futura fabricación de la PCB entre las que se incluyen las dos capas utilizadas para el trazado de pistas, dos capas de serigrafiado y dos de máscara protectora de soldadura, una para cada capa de cobre que contiene las conexiones entre componentes. Asimismo, se genera el archivo drill, que denota las posiciones de las vías sobre la PCB. Este fichero será utilizado para el taladrado y mecanizado de estos puntos durante la producción.

6 INTEGRACIÓN Y RESULTADOS

Para evitar resultados inesperados al final del proyecto a nivel de software, se comprueba cada uno de los módulos utilizando Unit Test. Este test funcional verifica que, ante unas entradas concretas, las funciones responden con las salidas esperadas. Para ellos se utiliza Unity y se generan los respectivos makefiles para poder compilar a través de consola cada test creado.

A nivel de proyecto individual, se comprueba la funcionalidad principal, consistente en el envío de información a través de rayos infrarrojos. Para ello se genera un código de test donde se envía constantemente una cadena de 10 bytes y se *blinkea* un LED cada vez que se recibe. Se sitúan frente a frente los transceptores de las dos placas y se varían progresivamente la distancia y el ángulo de transmisión, con el objetivo de generar un mapa de señal observando la cantidad de veces que parapadea el LED durante 5 segundos. Se repite la prueba 3 veces y se realiza una media de los resultados, que se plasman sobre la Fig. 8.

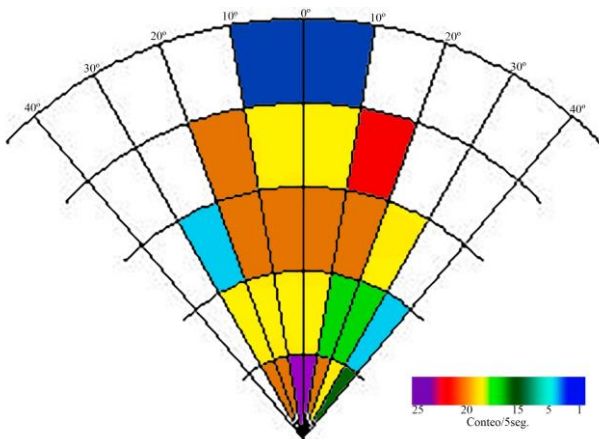


Fig. 8: Mapa de señal de comunicación IR

Tras analizar los resultados, se comprueba la hipótesis realizada, consiguiendo un disparo direccional hasta a 1 metro de distancia, con capacidad de transmitir la información suficiente para la correcta gestión del modo de juego Game of Drones.

Por otra parte, se incorpora el módulo de disparo en el sistema completo, conectando al puerto serie del F0 una placa nodemcu. Esta placa enviará los mensajes recibidos por puerto serie al servidor a través de Wi-Fi.

Se realiza un test de integración, verificando todo el flujo de paquetes apreciables en el diagrama de secuencia del Anexo 1. De esta manera, en el momento en que el mando dispara, el microcontrolador incorporado en el dron 1 recibe la orden de disparar, acción que realiza a través del módulo infrarrojos. En ese momento, el dron 2 recibe un impacto a través de uno de sus transceptores y comunica a través de puerto serie que ha sido alcanzado por el dron 1. Ese mensaje llega al servidor, que comunica al dron 2 el LED a iluminar debido al disparo y al mando 1 la penalización asociada a la zona de impacto y el arma utilizada.

Para acabar, se analizan los tiempos empleados en el flujo de mensajes, aprovechando un GPIO del microcontrolador. En el primer caso, se levanta el pin ('1' lógico) al recibir la orden de disparo por el puerto serie y se baja en el momento de emisión del disparo. Este pin se conecta a un analizador lógico que empieza a contar el tiempo utilizando un trigger de flanco de subida. Una vez capturados los datos, se obtiene que el tiempo dedicado es de 2,175ms, como se observa en la Fig. 9.

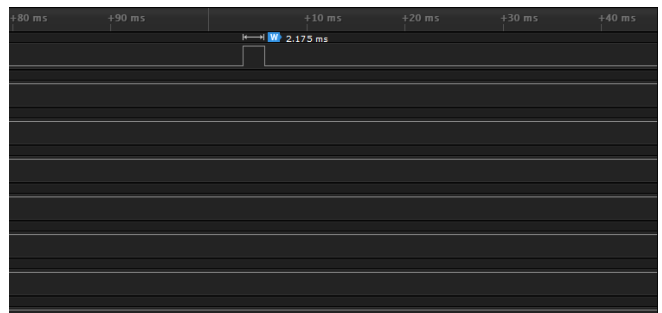


Fig. 9: Tiempo asociado al procesamiento de datos por parte del F0, desde que se recibe la orden de disparo hasta que se efectúa

En el segundo caso se calcula el tiempo empleado desde que el dron 1 recibe la orden de disparar hasta que el dron 2 envía al servidor que ha recibido un impacto. En ese caso, el tiempo empleado es de 9,60ms, como se puede apreciar en la Fig. 10.



Fig 10. Tiempo transcurrido desde que un dron recibe la orden de disparar hasta que el receptor lo comunica al servidor

Por otro lado, se consigue incorporar todos los elementos empleados en el proyecto en una PCB de 100*90mm, apta para ser colocada sobre un dron junto a una power bank para dar alimentación a través del USB, generando los archivos gerber y drills necesarios para su fabricación a través de alguna empresa dedicada.

7 CONCLUSIÓN

Tras finalizar el proyecto, podemos concluir que todos los objetivos han sido desarrollados con éxito, tanto los individuales como los del entorno de juego Game of Drones. Como futuros pasos, sería interesante fabricar la PCB e introducirla en un dron, para poder comprobar su comportamiento en el entorno real y poder probar el módulo SC15IS752 y el acelerómetro. Sería necesario ajustar parámetros y analizar paquetes perdidos, ya que debido al alto tráfico de paquetes y los entornos donde se practica el juego, la experiencia podría verse afectada.

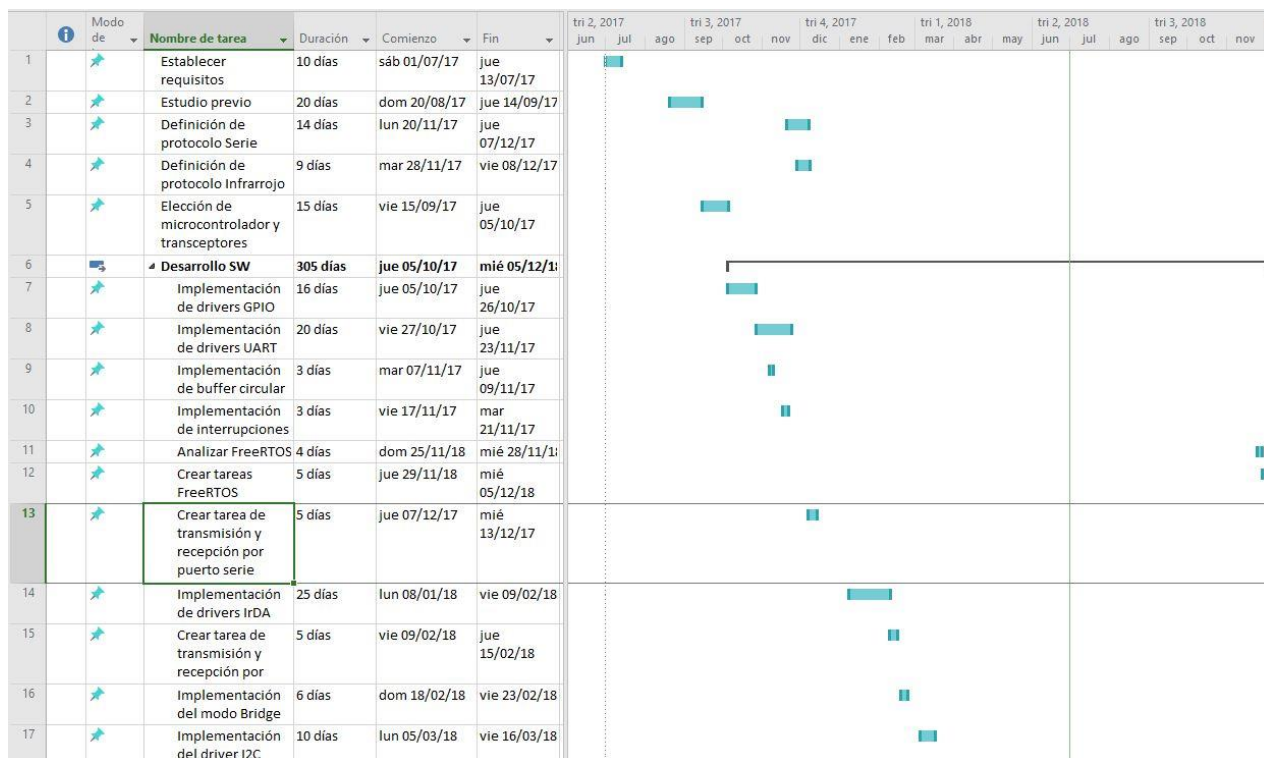
8 AGRADECIMIENTOS

A Patricia y mi familia por su apoyo incondicional. También a mis compañeros, amigos y tutor por toda la ayuda prestada durante la realización de este proyecto.

9 REFERENCIAS

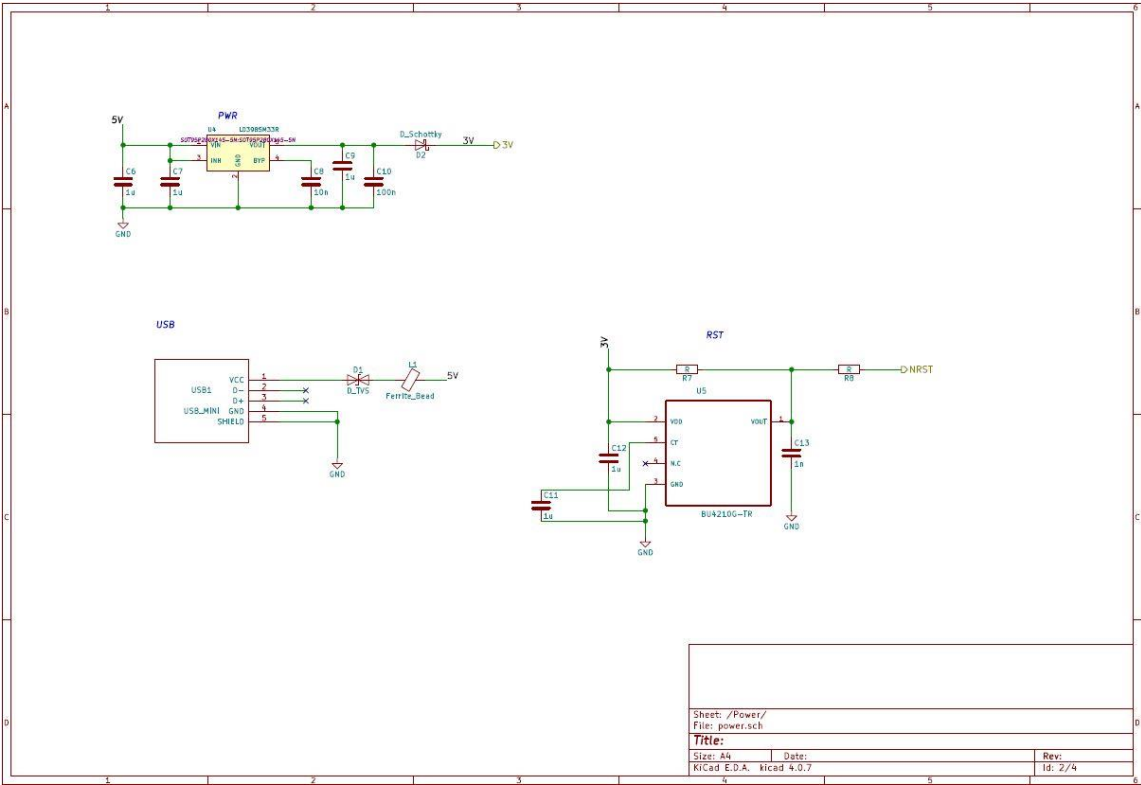
- [1] DDCStaff, "FPV ¿Y eso que es?"
- [2] Cristina Sánchez, "Grandes pilotos, rumbo incierto: los drones de carreras se atascan en España", ElDiario.es, Abril 2017.
- [3] Muy Drones, "¿Que es un cuadricóptero?"
- [4] Alberto Vilches, "Uso de drones en el sector industrial", Pilotando.
- [5] Departamento de comunicación, "Origen y desarrollo de los drones", Universidad de Valencia.
- [6] MikroElectrónica, "UART-Serial communication".
- [7] Altera, "Development and Education Board User Manual"
- [8] Advanced Circuits, "The history of the PCB"
- [9] Siobhannichofaigh, "What is a printed circuit board?", Design Spark.
- [10] Sparkfun, "PCB Basics".
- [11] GeekTheory, "Puertos y Buses 1: I2C y UART"
- [12] Paul Jackson, "Understanding PCB Manufacturing: Silk-Screening", OmniCircuit Boards.
- [13] SBProjects. "IR Remote Control Theory" (2016).
- [14] ElectronicaSi, "Montaje de transmisor y receptor de IR"
- [15] Página oficial de IrDA
- [16] P. Barker, A.C. Boucouvalas, V. Vitsas. "Performance modeling of the IrDA infrared wireless communications protocol". Artículo científico (2000).
- [17] ST. "Implementation of transmitters and receivers for infrared remote control protocols STM32CubeMX.".
- [18] ST. "Implementation of transmitters and receivers for infrared remote control protocols with MCUs of the STM32F0 and STM32F3 Series".
- [19] Panasonic. Datasheet de CND0313A.
- [20] "IR Receiver Modules for Remote Control Systems".
- [21] Vishay. Datasheet de TSOP2236.
- [22] Surface Emitter Technology. "High Speed Infrared Emitting Diodes, 850 nm"
- [23] Vishay. Datasheet de VSMY2853RG.
- [24] Infrared Transceiver Module (SIR, 115.2 kbit/s) for IrDA® Applications.
- [25] Vishay. Datasheet de TFDU4101.
- [26] ST. Existing projects for STM32F0.
- [27] ST. "STM32 configuration and initialization C code generation"
- [28] ST. Datasheet de LIS3DH.
- [29] Kicad. "Getting Started Tutorial".
- [30] NXP. Datasheet de SC16IS740.

Anexo 1

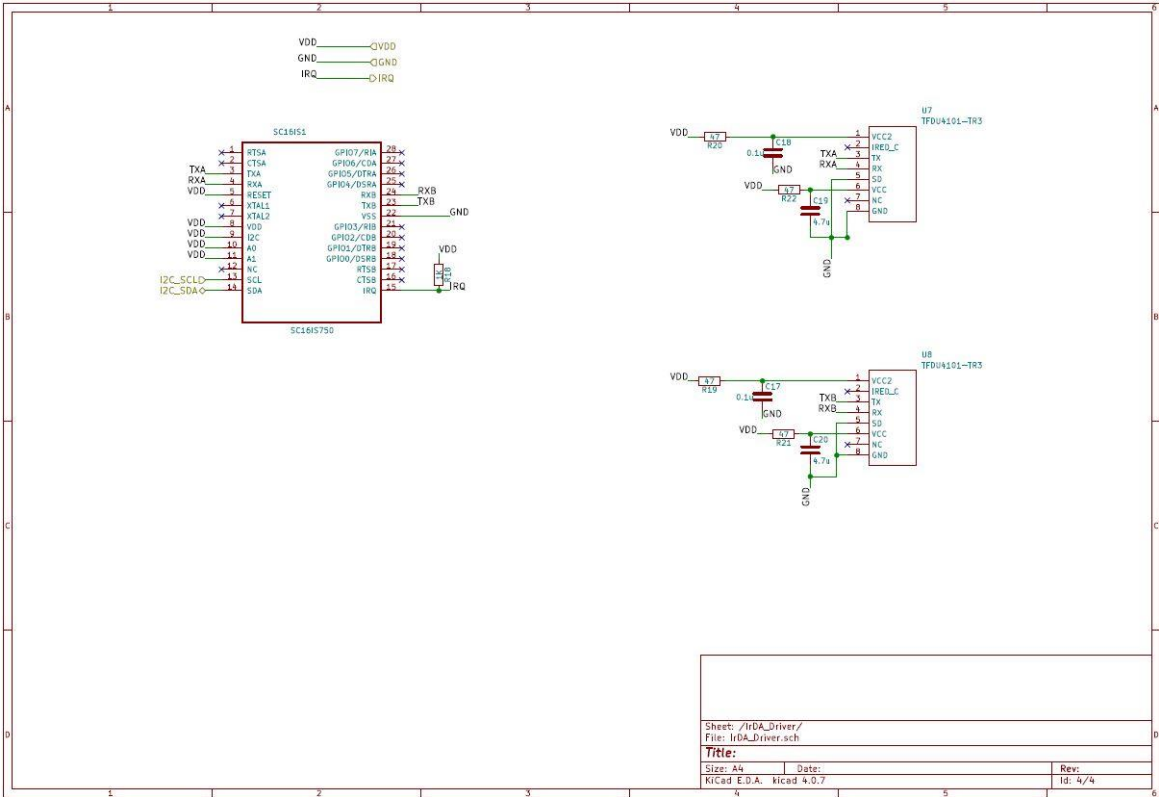


	<div><div><div><div></div><div></div><div></div></div><div></div></div></div>	Modo de	Nombre de tarea	Duración	Comienzo	Fin	tri 2, 2017 junjulago	tri 3, 2017 sepoctnov	tri 4, 2017 dicenefeb	tri 1, 2018 marabrmay	tri 2, 2018 jun
18			Implementación del driver del acelerómetro LIS3DH	5 días	lun 26/03/18	vie 30/03/18					
19			Implementación del driver del módulo SC16IS752	5 días	lun 02/04/18	vie 06/04/18					
20			Desarrollo de la PCB	54 días	dom 15/04/18	jue 28/06/18					
21			Desarrollo del esquemático	35 días	dom 15/04/18	vie 01/06/18					
22			Listar componentes para PCB	5 días	dom 15/04/18	jue 19/04/18					
23			Implementación del módulo de alimentación	10 días	vie 20/04/18	jue 03/05/18					
24			Implementación del microcontrolador	10 días	vie 04/05/18	jue 17/05/18					
25			Implementación del módulo	7 días	jue 17/05/18	vie 25/05/18					
26			Implementación del acelerómetro y	5 días	lun 28/05/18	vie 01/06/18					
27			Desarrollo del layout	16 días	jue 07/06/18	jue 28/06/18					
28			Generar netlist	3 días	jue 07/06/18	lun 11/06/18					
29			Distribuir huellas de	3 días	lun 11/06/18	mié 13/06/18					
30			Routear la PCB	10 días	jue 14/06/18	mié 27/06/18					
31			Generar gerbers y drills	2 días	mié 27/06/18	jue 28/06/18					
32			Test	139 días	vie 24/11/17	mié 06/06/18					
33			Test de transmisión por	5 días	vie 24/11/17	jue 30/11/17					
33			Test de transmisión por	5 días	vie 24/11/17	jue 30/11/17					
34			Test de transmisión por	5 días	sáb 10/02/18	jue 15/02/18					
35			Test de modo Brid	10 días	dom 25/02/18	jue 08/03/18					
36			Test de acelerómetro	5 días	dom 01/04/18	jue 05/04/18					
37			Test de electromagnético de PCB	3 días	lun 04/06/18	mié 06/06/18					

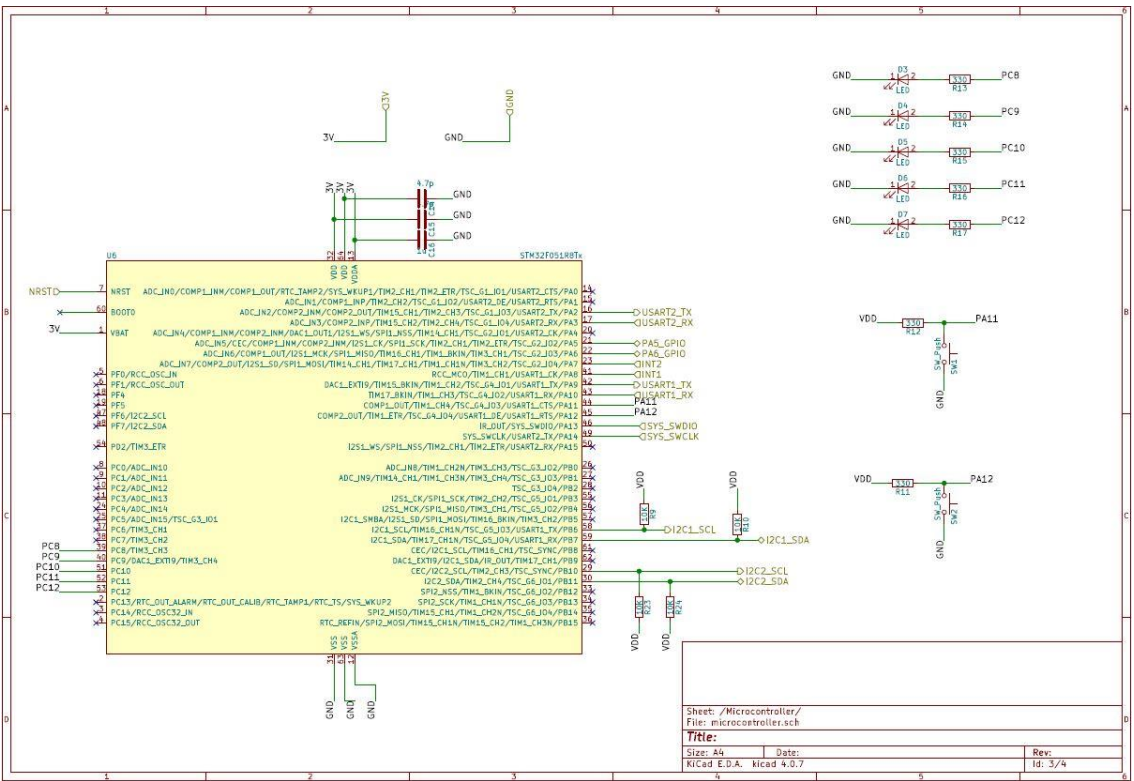
Anexo 3



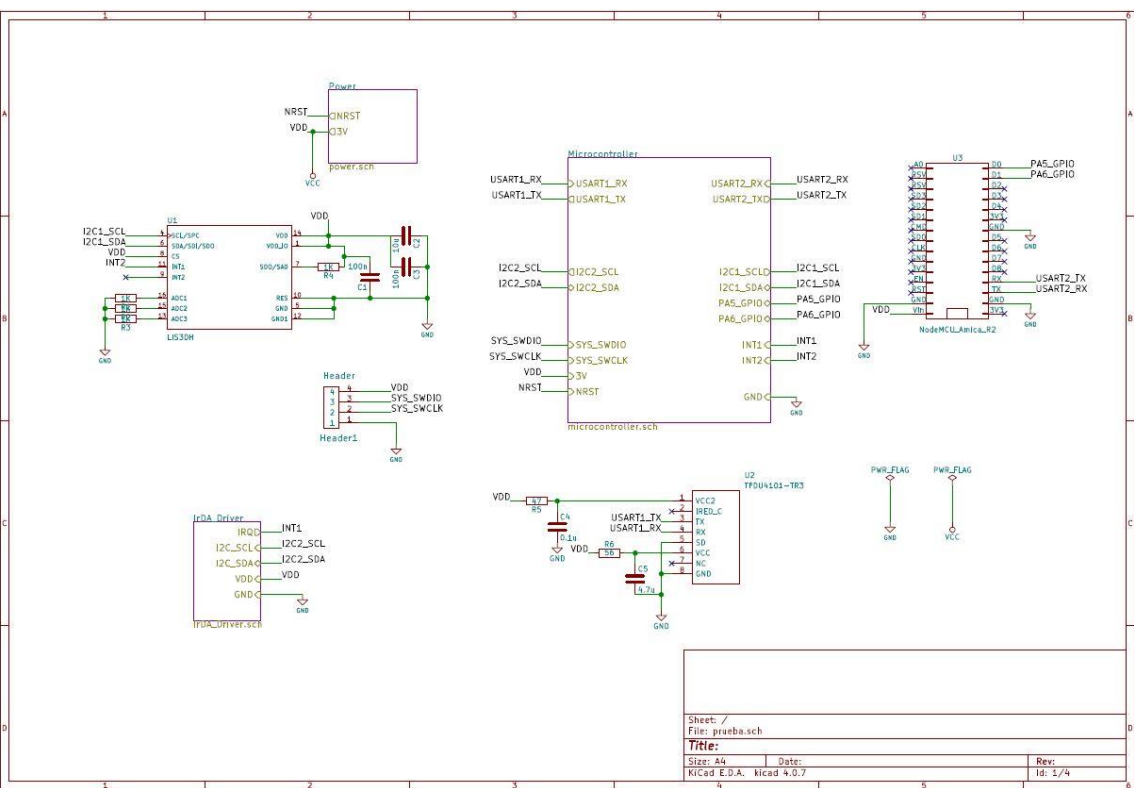
Anexo 4



Anexo 5



Anexo 6



Anexo 7

