



Trabajo de Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

DESARROLLO DE UN SISTEMA DE TRANSMISIÓN (EMISOR/RECEPTOR) CON MÓDULOS BLUETOOTH

Autor:

Pablo Antonio Quesada Aguilera

Tutor:

Félix Moreno González

Profesor Titular de Universidad

Departamento de Ingeniería Electrónica

(UPM)

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES

UNIVERSIDAD POLITECNICA DE MADRID

Madrid, Septiembre de 2016

ÍNDICE

Índice	3
Resumen	5
1 Introducción.....	10
1.1 Definición del problema.....	10
1.2 Objetivos	10
1.3 Antecedentes	11
2 Metodología.....	12
2.1 Descripción de la solución	12
2.2 Descripción detallada de cada dispositivo	13
2.2.1 Arduino Nano.....	13
2.2.2 Acelerómetro	15
2.2.3 Giróscopo	18
2.2.4 Módulo Bluetooth HC-06.....	20
2.2.5 Dispositivo Android	22
2.3 Sistemas de comunicación	23
2.3.1 Comunicación I ² C	23
2.3.2 Comunicación Bluetooth.....	25
2.4 Programación de dispositivos.....	28
2.4.1 Arduino Nano.....	29
2.4.2 Acelerómetro ADXL345.....	32
2.4.3 Giróscopo MPU-6050	35
2.4.4 Dispositivo Android	35
3 Resultados y discusión	40
3.1 Cumplimiento de objetivos	40
3.2 Posibles mejoras	41
3.3 Aspectos profesionales de futuro	42
4 Conclusiones	44
5 Bibliografía	45
6 Anexos	47
6.1 Presupuesto y planificación temporal	47
6.1.1 Presupuesto.....	47

6.1.2	Estructura de Descomposición del Proyecto	48
6.1.3	Diagrama de Gant.....	48
6.2	Índice de figuras	49
6.3	Índice de tablas	50
6.4	Abreviaturas, unidades y acrónimos	51
6.5	Glosario	53
6.6	Código fuente	54

RESUMEN

Este Trabajo de Fin de Grado se denomina “Desarrollo de un sistema de transmisión (emisor/receptor) con módulos Bluetooth”, donde se pone en práctica el envío de datos obtenidos en acelerómetros desde un Arduino a un dispositivo Android, mediante la conexión inalámbrica Bluetooth. Su objetivo es evaluar esta comunicación como posible mecanismo de transmisión de información a alta frecuencia y la determinación del proceso de ejecución más efectivo. Se trata de una etapa base o inicial a un proyecto de mayor envergadura, que llega a conclusiones más interesantes que la únicamente obtención de aceleraciones.

El uso de acelerómetros es debido a la propuesta de una futura instalación en palas de pádel como proyecto completo, significando este Trabajo de Fin de Grado como la realización de un sistema de control sobre este tipo de sensores, así como una efectiva comunicación Bluetooth entre este sistema y otro dispositivo en donde visualizar los valores en tiempo real.

Para la elección del método de funcionamiento de este sistema, ha dependido de los objetivos a llegar con la futura implementación en la pala, que, como es obvio, necesitará del menor peso posible que atenúe el efecto provocado con el aumento de carga, entre otras. Es por ello que, con la misión de disminuir al máximo este problema, se ha optado por la utilización de la comunicación I²C (*Inter-Integrated Circuit*) como método de transmisión entre los sensores y Arduino, ya que disminuye el uso de cables, manteniendo una velocidad de transmisión y control bastante buena. Además, de un Arduino Nano como elemento controlador de todos los sensores por su pequeño volumen y peso que ofrece respecto a otros tipos, donde es posible la aplicación de la comunicación I²C y Serie.

La velocidad también es una necesidad en cualquier procedimiento de control hoy en día, sobre todo, en aquellos que puedan provocar graves fallos y, en el peor de los casos, llegar al desastre. En este sistema a crear no ocurrirá esto, pero, sin embargo, se necesita alta velocidad tanto en la obtención de aceleraciones por parte del Arduino como en su transmisión al dispositivo para su representación y posterior estudio. Es necesario obtener rápidamente datos ya que en su puesta en práctica, en la de análisis de golpeo por la pala, se necesitará una constante captación de aceleraciones, puesto que existen altas variaciones de valores en un tiempo pequeño durante el proceso de golpear. Respecto al punto de alta velocidad en la transmisión Bluetooth, es obligatorio si se desea llegar al objetivo de representar los valores de las aceleraciones en tiempo real. Sin embargo, en este problema, también influirá la posterior evaluación de las cadenas de caracteres recibidas, que contienen toda la información, por parte del dispositivo Android y su trabajo con los datos almacenados, pudiendo llegar a ser un trabajo laborioso, influyendo gravemente en la velocidad de ejecución del programa.

La elección del tipo de acelerómetros ha dependido principalmente que obtenga datos fiables y que su transmisión pueda ser a través de la comunicación I²C. Otro aspecto influyente pero con menor importancia, el peso. Es por ello que se ha decidido por el acelerómetro ADXL345, que responde correctamente a las exigencias anteriores, obteniendo

aceleraciones en los tres ejes. El problema de éstos es que sólo admiten dos direcciones I²C, por lo que en una misma línea de comunicación sólo es posible la conexión de dos dispositivos de esta clase. Como solución, se decide por la utilización de un tercer sensor, el giróscopo MPU-6050. Éste es un dispositivo con mayores ventajas que el anterior, puesto que es capaz de alcanzar a conocer los giros, dándole un mayor campo de estudio al sistema de control.

Para la realización de esta parte de recogida de aceleraciones por parte del Arduino Nano, se ha creado un programa en el IDE (*Inter-Integrated Circuit*) de *Arduino* mediante un lenguaje de alto nivel denominado *Processing*, orientada a objetos, siendo similar a C++. En el método de programación, se ha buscado que se mantenga un control constante de aceleraciones lo más rápido posible, es decir, que su función principal sea la de toma de aceleraciones, junto con su envío por Bluetooth, sin grandes análisis de ellos, para que el periodo de obtención de valores sea lo mínimo posible.

Como se trata de un lenguaje orientada a objetos, aprovechando la ventaja que esto supone con la optimización de texto y la división en bloques, siendo más fácil su entendimiento y manejo, y buscando cierta flexibilidad en cuanto al número de sensores conectados, se ha optado por la creación de la clase *sensor* en el cual se instanciará cada acelerómetro o giróscopo que esté en conexión, guardando cada objeto como elemento de un vector. Para la creación de estos objetos, se realiza, como etapa inicial, un proceso de escáner en la línea I²C. Dicho proceso dará lugar también a un análisis del tipo de sensor a través de su dirección I²C, para clasificarlo, proceso necesario ya que cada uno debe de ser configurado de manera distinta.

La clase *sensor* contiene una serie de atributos como su número de identificación (para diferenciarlos de los demás), el tipo de sensor y otra serie elementos que definen su configuración (como el rango de escala). Además, incluye métodos para su posible reconfiguración, conocimiento de sus atributos y trabajos con ellos.

Posteriormente al proceso de escaneo, se da entrada al proceso principal que, como se dijo anteriormente, se trata de la obtención de aceleraciones mediante un bucle, leyendo cada sensor. Dicho proceso contará con un sencillo paso de los valores a su equivalente en fuerzas g, mediante su introducción en una fórmula que contiene constantes diferentes que dependen tanto del tipo de acelerómetro como del eje, obtenidas en una etapa de calibración. Como paso final, se enviarán todos los datos mediante un protocolo específico, a través de una cadena de caracteres, mediante el puerto Serie, en el cual está conectado el módulo Bluetooth, que se encargará de transmitirlo inalámbricamente.

Adicionalmente, como se quiere adquirir una cierta capacidad de control de todos los elementos y su posibilidad de cambiar sus configuraciones desde el dispositivo Android, se ha incluido en este bucle un proceso de lectura de órdenes enviadas por el dispositivo Android. Este proceso, para que no influya en la frecuencia de muestreo, sólo se iniciará en el caso de haber datos de entrada en el puerto Serie, es decir, llegado desde su módulo Bluetooth, y siguiendo un protocolo específico. En el caso de ser cumplidas ambas necesidades, se iniciará un trabajo de evaluación de la intención del mensaje y su actuación de acuerdo a ella.

La segunda parte de este proyecto es la de la obtención de información en un dispositivo Android a través de la comunicación Bluetooth y su representación principalmente. Se ha elegido dicho dispositivo por su facilidad de obtención y manejo por cualquier persona hoy en día, pudiendo ser eliminando un elemento en la lista de compra, puesto que no se necesita un dispositivo con altas prestaciones.

Para ello, se ha necesitado crear una aplicación que cumpla las dos necesidades comentadas en el apartado anterior. El entorno de desarrollo en este caso ha sido *Android Studio* en el que es posible programar en Java, C, C++, etc., que cuenta con la ventaja de simular las aplicaciones previamente a su instalación en un dispositivo.

Principalmente, respecto a la parte de su utilización, es una aplicación muy sencilla, con poco pasos para la misión de visualizar las aceleraciones. Sin embargo, en su interior, su comportamiento es más complicada por la ejecución de procesos en segundo planos que ralentiza el programa.

La comunicación Bluetooth en dispositivos Android se hace mediante la creación de hilos o *threads* que son los procesos en segundo plano que se acaban de comentar. Para lograr la conexión con otro dispositivo y su posterior comunicación mediante envío de datos, es necesaria la existencia de tres hilos diferentes, cada uno cumpliendo con la realización de un trabajo específico. Ellos son: el hilo *Servidor*, encargado de mantenerse en un estado de espera atendiendo solicitudes de posibles conexiones por parte de otros módulos Bluetooth, el hilo *Cliente*, crea las solicitudes de conexión que son llegadas al anterior hilo del otro módulo, y el hilo *Conexión*, siendo su trabajo el de enviar datos al otro módulo al cual está conectado y recibir lo entregado por éste.

El programa se divide en dos partes. La función de la primera es el encendido de su adaptador Bluetooth, visualización de todos los dispositivos anteriormente emparejados y su posible elección para hacer posible su conexión, dando lugar al inicio del hilo *Cliente*. Para ello, ha sido necesario el almacenamiento de todos los dispositivos emparejados en un *array*, en donde se guarda toda la información de cada uno de ellos, información necesaria para el enlace.

La segunda parte se inicia con la aceptación de la conexión y su paso al comienzo del trabajo del hilo *Conexión*, en donde se van a obtener las aceleraciones y el envío de solicitudes o cambios de configuración en los sensores. Este cambio de parte lleva consigo un cambio visual en la pantalla, con una nueva estructura.

En esta estructura, se representa mediante una lista de elementos, la información deseada de cada sensor, separados en niveles. En cada nivel, se representa la dirección del sensor para poder identificarlo de entre los sensores reales (los físicos), el valor de sus aceleraciones en los tres ejes en tiempo real y, además, la posible visualización de la historia de sus aceleraciones, mediante una gráfica dinámica aceleración-tiempo utilizada, que se actualiza en cada paso por la etapa de lectura.

Sin embargo, al igual que se buscó en el programa de Arduino la disminución de uso de memoria, sencillez en el entendimiento y manejo de datos, se inicia esta segunda parte con la obtención de información de los acelerómetros conectados. Con esto, se busca principalmente la obtención del número de sensores conectados para crear solo el número de objetos necesarios y así ahorrar en memoria y tiempo de ejecución.

Dichos objetos creados son de la clase creada *sensor.java*, que incluye atributos que determinan la configuración en la que se encuentran cada uno, necesarios a saber para mejorar el diseño de representación. Al igual que en el programa de Arduino, son también añadidos a un vector para hacer más sencillo su manejo.

En cuanto al método de ejecución predominante en esta segunda parte de la aplicación, que es al de obtención de información a través de Bluetooth, su análisis y actualización de los gráficos, se ha optado por un método temporizado con una frecuencia de reinicio alta, debido al colapso que suponía un procedimiento en bucle cerrado de lectura del buffer, al desearse la capacidad de interacción del usuario con la aplicación.

Esta interacción se basa principalmente en la visualización u ocultación de la gráfica y del envío de cambios de configuración al Arduino a través de la comunicación Bluetooth, provocando una interrupción en el bucle de lectura.

Todo este trabajo anteriormente explicado, como se dijo, es una base del proyecto de instalación en una pala de pádel en el que se quiere evaluar la eficacia de la utilización de la comunicación Bluetooth como método de transferencia inalámbrica de datos a alta velocidad. Es por ello que en esta parte no se ha querido efectuar grandes mejoras como la posibilidad de realizar más cambios desde el dispositivo Android, utilizar funciones de mejora en los sensores para obtener aceleraciones más precisas como filtros, mejorar el diseño y estudios adicionales con los datos obtenidos, etc., así como la utilización de otros sensores con mejores prestaciones. Todo esto corresponde en etapas futuras, siendo muy importantes en el logro de ellas para añadir valor al producto final.

Palabras clave

Acelerómetro, giróscopo, control, comunicación, Bluetooth, I²C, Arduino, Android.

Códigos UNESCO

Matemáticas:	Ciencia de los ordenadores:	Leguajes Algorítmicos (120302)
		Sistema de Control del Entorno (120314)
		Lenguajes de Programación (120323)
	Investigación Operativa	Análisis de Actividades (120701)
		Sistemas de Control (120702)
Física:	Electrónica:	Circuitos (220301)
		Elementos de Circuitos (220302)

Ciencias Tecnológicas: Electrónica: Diseño de Circuitos (330703)
 Receptores de Radio (330711)
 Transmisores de Radio (330712)
Telecomunicaciones: Radiocomunicaciones (332505)

1 INTRODUCCIÓN

En este capítulo se da una pequeña explicación sobre el problema que se plantea resolver en este trabajo, junto con los objetivos a llegar. Además, se hace una breve observación sobre aspectos de la actualidad relacionados con el problema, que ha servido como origen para hacer interesante su resolución.

1.1 Definición del problema

Hoy en día se vive una etapa en la que el control tanto de un elemento como de un sistema es uno de los ejercicios más importantes dentro de un proyecto para el análisis de su comportamiento, disminución de peligros y en la búsqueda de su perfección. Tan importante es esto, como en la posibilidad de mostrar los resultados en dispositivos cercanos a cualquier tipo de persona, siendo o no usuario profesional en el campo de tecnologías.

Es por ello que el problema planteado y resuelto en este TFG es la búsqueda de control de aceleraciones que se crean en una pala de pádel y la visualización de datos obtenidos o conclusiones en un dispositivo.

Además, como es obvio y más aún en jugadores profesionales, la búsqueda del menor peso posible ha sido también un punto a tener en cuenta. Por lo que también se busca implementar unos elementos y sistemas de comunicación en el control, que minimicen los impactos en este ámbito, intentando lograr que sea el menor daño a la dinámica y peso, con el que sea posible un movimiento lo más natural posible, pero siempre cumpliendo los objetivos, siendo uno de los más importantes, la velocidad de transmisión.

Sin embargo, este trabajo consiste en la creación de una base que dé soporte a este proyecto de mayor envergadura, que contiene actividades y estudios más específicos en la consecución de un control de dirección y dinámica más precisos. Es decir, con esta base se quiere demostrar que es posible su control y con qué medios de comunicación es más conveniente para cumplir con los objetivos propuestos.

1.2 Objetivos

En resumen al apartado anterior, la resolución de este problema en este trabajo busca cumplir estas misiones:

1. Lectura en tiempo real de aceleraciones.
2. Optimización del peso añadido.
3. Rapidez de comunicación.
4. Fiabilidad de datos obtenidos.
5. Visualización de datos en un dispositivo.

6. Precio bajo pero fiable de dispositivos.

1.3 Antecedentes

El aumento de importancia que se le está dando al deporte actualmente ha propiciado una mayor inversión para la búsqueda de unos mejores resultados durante la práctica de ellos. Dicha mejora se está logrando mediante una regeneración de elementos o de la técnica e introducción de instrumentos.

Respecto a la regeneración de elementos, se refiere a un aumento de calidad de los componentes propios a cada modalidad, que faciliten la ejecución del ejercicio, teniendo más relación con estudios sobre su mecánica o material.

Sin embargo, respecto a la técnica, mantiene una relación con el control de la realización de la prueba, en el cual se pone en examen los datos obtenidos tras su ejecución. La obtención de esta información es gracias a la utilización de dispositivos, conectados al deportista o no, que miden alguna característica durante la realización de la prueba. Esta evaluación de resultados da lugar a perfeccionar el modo de realizar el ejercicio por parte del profesional, corrigiendo fallos y su posibilidad de lograr a lo más alto y su posterior consolidación, consiguiendo una rentabilidad por la inversión.

Por lo tanto, en cada especialidad deportiva como en cada aspecto que influya durante su realización, se está intentando buscar métodos de control, permitiendo la ejecución de la actividad lo más natural posible, sin afectar al profesional, con la búsqueda de mejores resultados.

Tanto se busca mejorar la técnica como en obtener información del estado en el cual se encuentra el deportista. En estos años se han desarrollado dispositivos, en general, inalámbricos, que han permitido conocer su salud y evitar problemas graves, como fallos orgánicos o lesiones. Ejemplos de instrumentos electrónicos de medición que más se han utilizado son aquellos que controlan tu salud durante la actividad física como el ritmo cardiaco, calorías o estado de nivel de oxígeno o nutrientes, o aquellos que registran datos técnicos de la actividad, como velocidades, recorrido o movimientos. La característica principal que todos comparten es la de tener unas reducidas dimensiones y facilidad de implantación.

Además, debido al aumento de número de practicantes no profesionales de deportes que se está observando en los últimos tiempos y encontrarse en una generación tecnológica, que está utilizando similares productos que los profesionales, se ve con gran futuro el trabajo en este campo.

2 METODOLOGÍA

En este capítulo se describe cómo se ha conseguido resolver el problema planteado. Para ello, se detalla el proceso seguido puesto en práctica, junto con el fundamento de su elección. Además, se desarrolla detalladamente las características y el funcionamiento de cada dispositivo utilizado, así como brevemente otros tipos que realizan la misma función, para apoyar el método seleccionado.

2.1 Descripción de la solución

Para dar solución a este problema y cumplir con los objetivos que se demandan, ya que se trata del inicio de un proyecto mayor en el cual se busca la demostración de que es posible su realización, como se ha explicado anteriormente, se ha buscado dispositivos económicos principalmente pero, por supuesto, fiables en cuanto a sus respuestas y comportamiento a este tipo de proyecto.

Para ello, el punto más importante es la búsqueda de un sistema de control que impacte de forma mínima en el peso de la raqueta y a la forma de pegada. Por lo que se propone la implantación de la comunicación I²C, ya que con ello se minimiza el número de cables de control a utilizar, manteniendo una velocidad de emisión de datos rápida y muy aceptable.

En segundo lugar, el dispositivo que servirá de control, el cual recibirá los datos necesarios de aceleraciones obtenidos por parte de los acelerómetros será un Arduino Nano. Se escoge principalmente por su pequeño peso y volumen, precio bajo, alta velocidad de control, posibilidad de implantación de la comunicación I²C y facilidad de programación.

Además, como dispositivo donde mostrar los resultados obtenidos de los acelerómetros se ha optado por un dispositivo Android, con la utilización de la comunicación Bluetooth para la llegada de datos. Se quiere utilizar éste, por su actual cercanía a cualquier persona, facilitando el uso de esta aplicación y por la posibilidad de eliminación de un elemento en la compra para este sistema.

Por otro lado, la utilización de la comunicación Bluetooth, es obvia. Se busca la eliminación de obstáculos que en este caso ocasionaría el uso de cables para el envío de datos desde el elemento de control hasta el dispositivo móvil mencionado anteriormente, es decir, una transmisión inalámbrica. Además, se trata de una transmisión rápida y, sobre todo, admitida por los dos elementos que se desean interconectar. Un inconveniente respecto a esta elección es que se trata de una conexión punto a punto, en otras palabras, sólo es posible la existencia de dos elementos que interactúen. Otro aspecto en contra es la distancia entre dispositivos de transmisión, siendo un sistema de corto alcance.

Para dar respuesta a estos dos aspectos, se podría optar por la comunicación Wifi, que incluye los puntos a favor del anterior modo. Éste puede conectarse y comunicarse con

muchos puntos con mayor rapidez, es de largo alcance y mayor velocidad de transmisión. En contra, su consumo de energía es bastante mayor que la opción de Bluetooth, punto muy importante actualmente en el tema de ahorro energético.

Sin embargo, debido a estas superiores prestaciones de la comunicación Wifi, se opta por el Bluetooth, puesto que la utilización de este sistema inicialmente, como se ha declarado anteriormente como proyecto base, no se requerirá grandes distancias, si no de lecturas en el propio lugar de la aplicación del ejercicio, además del tema consumo energético comentado.

Como elemento de obtención de aceleraciones, se ha determinado la utilización de los acelerómetros ADXL345, dispositivo capaz de obtener aceleraciones de hasta 16g, con alta precisión y con posibilidad de comunicación I²C. En cambio, un inconveniente de éstos con el método de I²C es que sólo se puede configurar en dos direcciones distintas, por lo que únicamente es posible incorporar dos elementos de este tipo al sistema.

Como solución, y además para posibles futuras mejoras, se ha decidido incluir el giróscopo MPU-6050, pues incluye iguales posibilidades que el acelerómetro ADXL345, más lecturas de giros.

2.2 Descripción detallada de cada dispositivo

2.2.1 Arduino Nano

Arduino es una compañía del campo de la electrónica, dedicada al desarrollo hardware y software. Su principal característica es que es una plataforma de electrónica “open-source”, es decir, de código abierto, fácil de usar, que cuenta con su propia IDE (Integrated Development Environment) con la que crear nuestros propios programas.

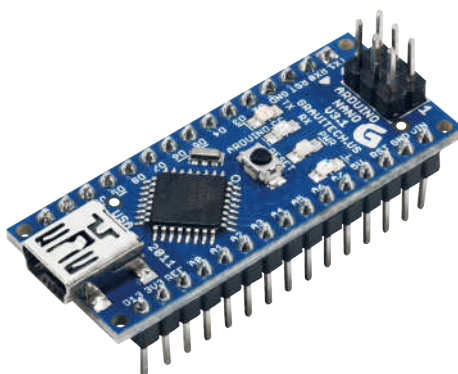


Figura 1 Arduino Nano

El hardware se compone de una placa de circuito impreso con un microcontrolador, normalmente Atmel AVR, y puertos de entrada/salida, los cuales se pueden conectar a otros módulos que aumenten las funcionalidades de la placa. La compañía ofrece gratuitamente los esquemáticos de diseño de hardware de sus placas, permitiendo su fabricación por cada persona sin necesidad de comprarla.

El Arduino Nano es una de las placas de desarrollo con menor tamaño que la marca Arduino trabaja, con unas dimensiones de 18,5x43,2 mm y un peso de 5gr, datos buenos para la implantación en la raqueta.

Esta placa incorpora el microchip ATmega328P, por lo que la hace mantener similares prestaciones que el Arduino UNO en cuanto a velocidad y tipos de señales, diferenciándose principalmente por el número de pines posibles a conectar. Respecto a su método de alimentación, es vía conector Mini-B USB.

Tabla 1 Características principales del Arduino Nano

Características principales electrónicas	
Microcontrolador	ATmega328P
Tensión de trabajo	5 V
Tensión de entrada recomendada	7-12 V
Límites de tensión de entrada	6-20 V
Número de pines digitales	14(6 de ellos con salida PWM)
Número de pines analógicos	8
Intensidad de corriente	40mA
Memoria Flash	32 Kb
Memoria SRAM	2 Kb
Memoria EEPROM	1Kb
Frecuencia	16 MHz
Comunicación admitidas	I ² C y Serie.

Entradas y Salidas

Como se acaba de comentar existen un total de 22 pines analógicos y digitales. Pero además, existen otros que tienen otras funcionalidades adicionales (la mayor parte de ellos) que los caracterizan:

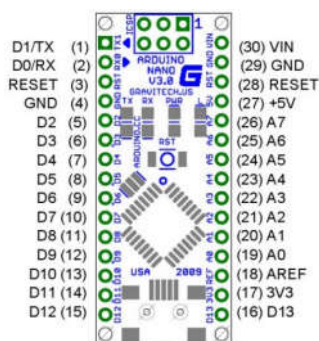


Figura 2 Diagrama de pines del Arduino Nano

- Pines RX, TX, D2-D13: Pines digitales.
- Pines A0- A7: Pines analógicos de 10 bits de resolución.
- Pin RX y TX: Son los pines de recepción y de transmisión respectivamente utilizados en la comunicación Serie.
- Pin REF: Proporciona una salida analógica de referencia a la que está trabajando el sistema.

- Pin RST: Pin de reseteo del programa.
- Pin GND: Pin de tierra.
- Pin 3V3 genera una tensión de 3,3 voltios.
- Pin 5V: genera una tensión de 5 voltios.
- Pines D2 y D3: Pin para interrupciones externas.
- Pines D3,D5,D6,D9,D10,D11: Salidas PWM de 8 bits
- Pines D10, D11, D12, D13: Para comunicación SPI.
- Pines A4 y A5: En la comunicación I²C, SDA (*Serial Data*) y SCL (*Serial Clock*), respectivamente.
- Pines ICSP: Se utilizan para configurar el microcontrolador.

2.2.2 Acelerómetro

Definición

Un acelerómetro es un dispositivo que se encarga de medir aceleraciones, definida como la tasa de cambio de velocidad de un elemento. Esta tasa se puede medir tanto en m/s² como en fuerzas g y su función principalmente se basa en medir orientaciones, movimientos o vibraciones.

Respecto a la forma de funcionamiento, son dispositivos electromecánicos muy variados, por lo que algunos ejemplos de los más utilizados son los mecánicos, capacitivos, piezoeléctricos, piezorresistivos o térmicos. Como breve explicación, se puede decir que los mecánicos emplean una masa inerte y un resorte elástico, obteniendo las aceleraciones con la medición de la deformación longitudinal del resorte.

En cuanto a los acelerómetros capacitivos, contienen internamente placas capacitivas, unas fijas y otras móviles conectados a unos resortes que se mueven debido a las fuerzas de aceleración en el elemento. Este movimiento ocasiona una diferencia de capacitancia entre ellos, dándonos una equivalencia de su aceleración.

El funcionamiento de los acelerómetros piezoeléctricos y piezorresistivos son semejantes, en el cual la deformación de un material ocasiona cambios en las características eléctricas que, en el caso de los piezorresistivos se determina la aceleración mediante el cálculo de su resistencia, y en el de los piezoeléctricos, a partir de su tensión eléctrica.

Por último, sobre los acelerómetros térmicos, hay que decir que constan de un sustrato de silicio hueco donde está colocada una resistencia con dos termopares en los extremos. La resistencia ocasionará una burbuja caliente entre estos dos termopares que, dependiendo de la inclinación o aceleración ocasionará su cambio de estado y un cambio de tensión, función de la temperatura.

Descripción del acelerómetro ADXL345

Se trata de un acelerómetro desarrollado por Analog Devices pensado para la implantación en aplicaciones que mantienen un movimiento tanto constante como interrumpido por la gran variedad de funciones que incluye.

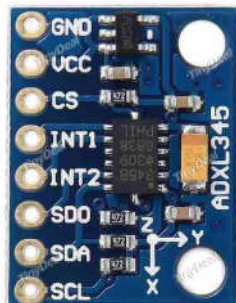


Figura 3 Acelerómetro ADXL345

ADXL345 es un pequeño acelerómetro de 3 ejes de bajo consumo energético pero con alta resolución (13 bits), con capacidad de medir hasta $\pm 16g$. Sus salidas son en formato de 16 bits en formato de complemento a 2, siendo accesible mediante comunicación SPI o I²C.

Sus medidas pueden ser tanto estáticas, para obtención de inclinaciones, como dinámicas, para movimientos y choques. Debido a su resolución de 4mg/LSB es posible detectar cambios de inclinación de menos de 1°.

Entre las funciones que mantiene, se puede destacar detección de ausencia de actividad o inactividad, así como de excesos de aceleraciones respecto a un nivel establecido por el programador, detección de pequeños impulsos individuales o dobles, o cuando se produce la caída libre.

Como se ha comentado anteriormente, se trata de un dispositivo que puede consumir poca energía. Para conseguir este objetivo, el ADXL345 modula su consumo proporcionalmente a la frecuencia de muestreo. Además, es posible su programación en modo de ahorro energético, pero sólo en un intervalo de frecuencia determinado, entre 12.5 Hz y 400Hz. Sin embargo, como desventaja, el ruido es mayor. En las tablas siguientes¹ se puede ver cómo disminuye la intensidad de corriente con el paso al modo de ahorro.

¹¹ Tablas obtenidas del documento Datasheet del Acelerómetro ADXL345.

Tabla 2 Consumo energético en el modo activo

Output Data Rate (Hz)	Bandwidth (Hz)	Rate Code	I _{DD} (μA)
3200	1600	1111	140
1600	800	1110	90
800	400	1101	140
400	200	1100	140
200	100	1011	140
100	50	1010	140
50	25	1001	90
25	12.5	1000	60
12.5	6.25	0111	50
6.25	3.13	0110	45
3.13	1.56	0101	40
1.56	0.78	0100	34
0.78	0.39	0011	23
0.39	0.20	0010	23
0.20	0.10	0001	23
0.10	0.05	0000	23

Tabla 3 Consumo energético en el modo ahorro

Output Data Rate (Hz)	Bandwidth (Hz)	Rate Code	I _{DD} (μA)
400	200	1100	90
200	100	1011	60
100	50	1010	50
50	25	1001	45
25	12.5	1000	40
12.5	6.25	0111	34

También, una mayor disminución de consumo es posible mediante la habilitación de la funcionalidad *Auto Sleep Mode*. Con esto, es posible entrar automáticamente en un estado de trabajo menor con una frecuencia de muestreo muy bajo, tras un cierto tiempo determinado en el cual se mantenga estático, no registrándose cambios, con intensidades cercanas a 23 μA. Aún, para una menor intensidad de corriente durante las operaciones, menor a 0,1 μA, existe el estado *Standby Mode*, que se diferencia del anterior en que no registra ninguna medida, pero sin perder los datos almacenados en la memoria.

En relación a su composición, el acelerómetro ADXL345 contiene una estructura de superficie micromecanizada de polisilicona construida sobre una oblea de material semiconductor. La estructura se suspende sobre la oblea con un resorte de polisilicio que proporciona resistencia contra las fuerzas debidas a la aceleración aplicada.

Se trata de un acelerómetro capacitivo. Existen unas placas fijas y otras unidas a la estructura de polisilicona. La aceleración debida al movimiento del acelerómetro provoca la deformación de la masa en el cual está fijo con la placa capacitiva móvil, dando lugar a una diferencia capacitiva, proporcional a la magnitud de la aceleración.

Las aceleración obtenidas en los tres ejes son enviadas en un formato de 16 bits, dividida en byte alto y byte bajo. Por lo que la forma de obtención de datos es mediante la lectura de los registros DATA1, DATA0, DATAY1, DATAY0, DATAZ1 y DATAZ0.

2.2.3 Giróscopo

Definición

Los giróscopos o giroscopios son unos dispositivos que sirven para medir, mantener o cambiar la orientación en el espacio de algún elemento. Antiguamente, los primeros estaban compuestos principalmente por un cuerpo con simetría de rotación que gira alrededor del eje de dicha simetría. En el caso de existencia de una fuerza que busca cambiar la orientación del eje de rotación, cambia de orientación girando respecto de un tercer eje, perpendicular tanto al eje inicial de rotación como de aquel que se le ha empujado a girar.



Figura 4 Giróscopo

Este peculiar comportamiento proviene gracias a lo que se conoce como inercia giroscópica y la precesión, propiedades a todo material en rotación.

La inercia giroscópica también se le denomina rigidez en el espacio. Es consecuencia de la primera ley del movimiento de Newton que expresa que un cuerpo tiende a continuar en su estado de reposo o movimiento uniforme si no está sometido a fuerzas externas. Aplicado en el giróscopo, se ve cumplida con el continuo movimiento de rotación del disco respecto del mismo eje tras una fuerza perturbadora.

Respecto a la precesión, éste expresa que cuando una fuerza es aplicada a un giroscopio que cambia la dirección del eje de rotación, este eje se desplaza en una dirección que forma un ángulo recto con la dirección de la fuerza. Esto es causado por el momento angular del cuerpo en rotación y la fuerza, simultáneamente.

Sin embargo, la tecnología ha evolucionado y ahora han sido sustituidos por otros. Existen muchos tipos pero los más utilizados son los giróscopos piezoeléctricos, los capacitivos y los MEMS (Micro Electric-Mechanical System).

El giróscopo piezoeléctrico se basa en un cristal que oscila cuando una fuerza actúa sobre él, que hace que los elementos piezoeléctricos que lo soportan se deformen y por lo tanto, su resistencia eléctrica cambie, variación que es medida para obtener el valor de la fuerza actuada y la aceleración, ambas relacionadas con la fuerza de Coriolis.

En segundo lugar, el giróscopo capacitivo está formado por una parte móvil que realiza un movimiento de vibración con respecto al eje de rotación del giróscopo, y otra fija respecto a ésta, por lo que forma un condensador. En el caso de que esté girando, el elemento vibratorio se moverá, por lo que la distancia entre el elemento fijo y móvil variará, provocando la variación de capacidad del sistema.

Respecto a los giróscopos MEMS, se basa en el estudio en las variaciones en las vibraciones de materiales pero en este caso, a nivel molecular. Actualmente, son pequeños sensores de bajo costo que están integrados en gran cantidad de dispositivos como móviles por estos dos motivos, además de sus altas prestaciones de control.

Descripción del giróscopo MPU-6050

Este sensor proviene de la empresa InvenSense, el cual contiene un acelerómetro y giróscopo MEMS, con la posibilidad de añadirle un magnetómetro. Es un dispositivo de 9 grados de libertad (3 para el acelerómetro, 3 para el giroscopio y 3 para el magnetómetro) que posee un ADC (*Analog-Digital Converter*) con una resolución 16 bits, con lo que el rango dinámico está dividido en 65536 fracciones, que hace de él un sensor muy preciso y ampliamente utilizado. Cuenta con la capacidad de comunicación SPI e I²C.

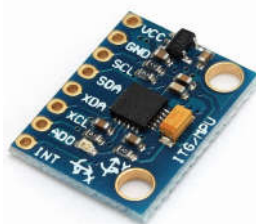


Figura 5 Giróscopo MPU-6050

Es un sensor muy apropiado para la parte de control en robótica, medida de vibraciones, sistemas de medición inercial o IMU (*Inertial Measurement Unit*), detector de caídas, sensor de distancia y velocidad, entre otras. Adicionalmente, incluye un sensor de temperatura digital, parte importante ya que el material que mide las aceleraciones y giros es muy susceptible a la temperatura, provocando errores en las medidas; junto con algoritmos internos para la calibración del dispositivo.

Las características técnicas de la parte que trabaja como giróscopo se pueden resumir en estos puntos:

- Salida digital de ángulos respecto a los ejes X, Y y Z con un rango de escala programable de ± 250 , ± 500 , ± 1000 y $\pm 2000^\circ$ /segundo.
- ADC de 16 bits para el muestreo.
- Rendimiento de ruido a baja frecuencia mejorado.
- Filtro de paso bajo programable.
- Intensidad de corriente en modo Activo de 5mA.
- Intensidad de corriente en modo *Standby* de 5mA.
- Factor de escala de sensibilidad calibrado de fábrica.

La sensibilidad, dependiendo del rango de escala al que esté programado, disminuye altamente como se muestra en la siguiente tabla, sin embargo, siguen siendo altos por cada grado, manteniendo su cualidad de precisión.

Tabla 4 Sensibilidad en la medida de la velocidad de giro del giróscopo MPU-6050

Rango de escala completa (°/s)	Sensibilidad (LSBs/dps)
±250	131
±500	65,5
±1000	32,8
±2000	16,4

En cuanto a las características técnicas del acelerómetro, son:

- Salida digital de aceleraciones respecto a los ejes X, Y y Z con un rango de escala programable de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$.
- Convertidor ADC de 16 bits para el muestreo.
- Intensidad de corriente en modo de trabajo de $500\mu A$.
- Intensidad de corriente en modo de baja potencia: $10\mu A$ a 1.25Hz, $20\mu A$ a 5Hz, $60\mu A$ a 20Hz, $110\mu A$ a 40Hz.
- Detección de orientación.
- Detección de impulsos.
- Interrupción por caída libre.
- Interrupción por registros superiores al límite.

Al igual que pasaba en el giróscopo, la sensibilidad del acelerómetro también se verá afectada por el rango de escala programado, aunque sigue obteniendo medidas precisas:

Tabla 5 Sensibilidad en la medida de la aceleración del giróscopo MPU-6050

Rango de escala completa	Sensibilidad
±2g	16384
±4g	8192
±8g	4096
±16g	2048

2.2.4 Módulo Bluetooth HC-06

El modo de funcionamiento de la comunicación Bluetooth se explica más adelante en el apartado 2.3.2.

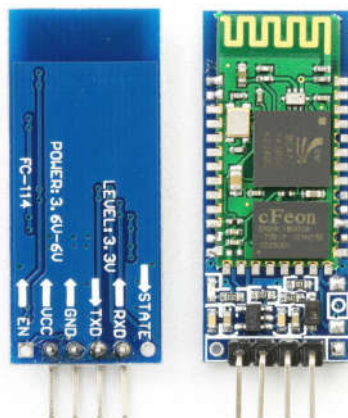


Figura 6 Módulo Bluetooth HC-06

Como se comentó en el apartado de descripción de la solución, para el envío de datos desde la placa Arduino, se eligió el módulo Bluetooth HC-06. Este módulo, muy económico y fácil de usar, distribuido por la empresa *Wavesen*, es muy popular para aplicaciones en Arduino y microcontroladores PIC. Además, es elegido ampliamente por su pequeño tamaño y sus buenas características de transmisión y recepción, junto al bajo consumo tanto en el estado de funcionamiento como en el de espera. Sin embargo, un inconveniente es su escasa información en la web, siendo muy diferente a otras versiones como HC-03/04/05 en cuanto a conexiones y configuración.

En él, existen 4 pines con los que conectará a la placa Arduino:

- VCC: Entrada de alimentación.
- GND: Conexión a tierra.
- RX: Pin receptor. Entrada de datos del puerto Serie.
- TX: Pin transmisor. Salida de datos del puerto Serie.

En cuanto a las características técnicas, se puede resumir en estos puntos:

- Protocolo Bluetooth: Bluetooth Specification v2.0+EDR
- Voltaje de alimentación: 3.6-6V.
- Corriente de operación: 30 mA a 40 mA.
- Corriente en estado *Sleep Mode* : <1mA.
- Perfiles: Puerto Serie Bluetooth.
- Frecuencia: 2.4 GHz, banda ISM.
- Modulación: GFSK (Gaussian Frequency Shift Keying).
- Ancho de banda ajustable: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
- Potencia de emisión: ≤ 4 dBm, Clase 2.
- Sensibilidad: ≤ -80 dBm a 0.1% BER (Bit Error Rate).
- Velocidad: Asíncrona: 2 Mbps (max.)/160 kbps, Síncrona: 1 Mbps/1 Mbps.
- Seguridad: Autenticación y encriptación (Contraseña por defecto: 1234).

- Dimensiones totales: 4.4 cm x 1.6 cm x 0.7 cm.
- Temperatura de operación: -25 °C a +75 °C.
- Aviso de encendido y emparejamiento mediante LED (*Light-Emitting Diode*).

Aspectos adicionales de este dispositivo, es que sólo funciona como esclavo, es decir, él no puede iniciar la petición de emparejamiento con otro módulo, sino que debe de ser éste el que reciba invitaciones de conexión por parte de otros y se encargue de aceptarlas o no. Además, incluye una memoria por lo que es capaz de recordar a su compañero de conexión para futuras comunicaciones y no volver a pedirle la contraseña. Aunque dicha memoria es posible resetearla mediante la activación del pin 26 (o KEY).

Por último, al igual que sus otras versiones, es posible su configuración y conversación para conocer información suya como su nombre, versión, estado, etc. Para ello, se realiza mediante con los comandos AT que aparecen en el *Datasheet*² del dispositivo, a través del puerto Serie.

2.2.5 Dispositivo Android

El sistema operativo Android, hoy en día, es uno de los sistemas más utilizados en los dispositivos móviles, tablets y relojes inteligentes con una ocupación del 85% del total elegido principalmente por ser un “sistema operativo abierto”, con una mayor posibilidad de control y personalización. Es por ello que se decidió utilizar Android y no iOS, aun teniendo éste una mayor eficiencia en importantes procesos de ejecución y mejores gráficos.

Respecto a este elemento, no hay que incidir mucho puesto que en la mayoría de dispositivos móviles que actualmente se utilizan es posible su aplicación. La función necesaria que debe de tener es la comunicación Bluetooth, función que todos lo incluyen.

En cuanto a la mínima versión Android es indiferente, puesto que no se necesitan altas prestaciones para la ejecución de la aplicación, sólo la conexión con otros dispositivos mediante Bluetooth y la creación de unos gráficos dinámicos muy simples.

Sin embargo, debido a este proceso continuo de toma de datos y actualización de gráficas, hace que sea un trabajo que consuma una cantidad media de recursos, recomendándose tener un terminal ni muy antiguo ni con baja RAM (*Random Access Memory*) que pueda provocar bloqueos durante la ejecución del proceso.

² Datasheet disponible en http://www.wavesen.com/mysys/db_picture/news3/2015121885021101.pdf.

2.3 Sistemas de comunicación

2.3.1 Comunicación I²C

I²C viene de la abreviatura de *Inter-Integrated circuits* (circuitos integrados e interconectados). Se trata de un bus de datos bidireccional que fue diseñado por Philips Semiconductors durante los años 80, basado en dos hilos. Se utiliza para simplificar la interconexión entre dispositivos, memoria u otros con el microprocesador de una manera inteligente.

Su principal característica es que su comunicación es a través de dos líneas, la de datos (SDA) y la de reloj (SCL). Es por ello que se trata de un protocolo síncrono. Los dispositivos que están interconectados pueden jugar dos tipos de papeles: ser maestros o, por otro lado, esclavos, diferenciándose en que son los másteres los que deciden qué esclavo habilitar para la puesta en marcha de la comunicación y los que marquen la señal de reloj. En caso de que existieran varios másteres, debería existir una comunicación entre ellos, por el cual decidieran quien mantiene el mando en cada momento.

Sin embargo, existen desventajas con respecto a otros tipos de protocolos, siendo una de ellas la velocidad de transmisión, que no es muy alta. Es comparable a la de las UART (*Universal Asynchronous Receiver-Transmitter*), de unos 100Kbytes por segundo, programado en el modo de funcionamiento estándar. Aunque el bus serie I²C ha sido extendido a velocidades de hasta 3,4Mbits/s configurado en el modo *High-speed*. Otro aspecto negativo es que no admite grandes longitudes de separación, pocos metros. En tercer lugar, el número de esclavos a utilizar viene limitado por el número de bits de direccionamiento que exista (normalmente 8 bits) y siempre que su capacidad máxima sumada de todos los dispositivos no supere los 400pF.

Las líneas SDA y SCL son del tipo drenador abierto, igual que las de colector abierto pero asociadas a un transistor de efecto de campo. Se deben polarizar en estado alto (conectándose a la alimentación por medio de resistencias “pull-up”), lo que define una estructura de bus que permite conectar en paralelo múltiples entradas y salidas. Es decir, con la imposición de este tipo en estas líneas se consigue la posibilidad de conectar varias fuentes de datos a un mismo hilo. Si ningún dispositivo accede al bus o transmite un cero, el bus se mantendrá a nivel alto o, en cambio, a bajo si alguno envía un cero.

Los inconvenientes de este tipo de conexión en colector abierto o drenador abierto es que estas resistencias juegan un papel fundamental en las capacidades de la línea, influyendo en el tiempo de cambio de nivel, intensidad de corriente, etc. Es por ello que se debe elegir con razonamiento estos valores, que pueden ir desde 1,8k Ω a 47k Ω . Aunque un valor menor de resistencia incrementa el consumo, también disminuye la sensibilidad al ruido y mejora el tiempo de subida y bajada de flancos. Es por ello, que se suele utilizar una resistencia entre 1,8k Ω y 10k Ω .

Modo de funcionamiento

El reloj determina la velocidad de transmisión de los datos, por lo que los modos de transferencia de datos se dividen en:

- Modo Estándar (*s-Mode*): 100kBits/seg.
- Modo Rápido (*F-Mode*): 400kBits/seg.
- Modo Alta Velocidad (*Hs-Mode*): Más de 3,4MBits/seg.

Una característica de este sistema es que se transmite la información por la línea SDA mediante datos de 8 bits, seguidas a continuación por un noveno bit de reconocimiento ACK (*Acknowledgement*) como respuesta por parte del esclavo, que avisa de que la transferencia ha sido posible (nivel bajo si se ha efectuado correctamente, nivel alto en caso contrario) y se continúa o, en cambio, hay algún error por lo que se libera el bus I²C.

Para iniciar la comunicación, se tiene que cumplir la condición de bus libre, es decir, tanto las señales SDA como SCL deben estar en estado lógico alto, pudiendo ocupar el bus cualquier maestro y estableciendo la condición de inicio o *start*. Para ello, pone SDA a nivel bajo manteniendo SCL alto, avisando a todos los esclavos de una posible solicitud de conexión y manteniéndolos en un estado de espera.

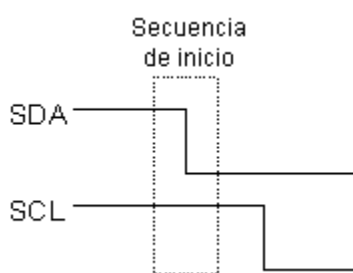


Figura 7 Secuencia de inicio en I²C

A continuación, manda el primer byte, en el que los siete primeros bits contiene la dirección del dispositivo con el que se quiere comunicar, más un octavo bit que le informa del tipo de operación a hacer, pudiendo ser de lectura o escritura, con un 0 o 1 respectivamente. Seguidamente, el esclavo con dicha dirección genera un pulso de reconocimiento (en estado bajo) con el que avisa de que reconoce la solicitud y es posible iniciar la comunicación. Respecto a los demás esclavos, ignorarán todos los cambios en el bus hasta nueva condición de inicio.

En caso de que el maestro le indique una escritura, éste le enviará datos mientras continúe recibiendo señales de reconocimiento. Finaliza la transmisión en el caso de errores de transmisión anunciado por ACK o cuando se hayan enviado todos los datos.

Por el contrario, si indica que el maestro necesita una lectura, actuará de la misma forma, siendo esta vez mandando el bit ACK por el maestro.

Finalmente, el maestro puede seguir direccionando y comunicándose con otros esclavos, o, en cambio, dejar libre el bus con la condición de parada en el cual pone SCA y SCL a nivel alto.

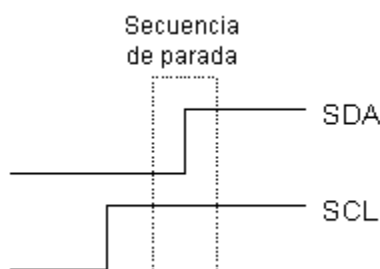


Figura 8 Secuencia de parada en I²C

2.3.2 Comunicación Bluetooth

Bluetooth es una tecnología de red de área personal inalámbrica o WPAN (*Wireless Personal Area Network*), es decir, una tecnología de corto alcance, que posibilita la transmisión de voz y datos entre equipos mediante un enlace por radiofrecuencia. Fue desarrollado por Ericsson en 1994 aunque su nombre es una referencia a un rey danés del siglo X llamado Harald Blatand (en inglés, Harold Bluetooth) en el que intentaba unir diferentes tecnologías como las de los ordenadores, móviles y resto de periféricos, que dio lugar a la formación del grupo empresarial Bluetooth Special Interest Group (Bluetooth SIG).

La comunicación Bluetooth es una forma de intercambiar información entre dos dispositivos sin necesidad de una conexión física. Sus características principales son, cómo se acaba de decir, que sólo es posible la existencia de dos puntos de comunicación inalámbricamente, y que es un sistema de corto alcance, pero con alta velocidad de transmisión.

Sin embargo, existen ventajas de usar esta tecnología pues, a diferencia de otras como la tecnología *IrDa* (*Infrared Data Association*) que utiliza una conexión infrarroja, no necesita una línea de visualización directa para comunicarse, pudiendo haber obstáculos entre ellos. Además, es un protocolo de comunicaciones de bajo consumo. Con respecto a la comunicación Wifi, no necesita una configuración previa, aunque este sistema ofrece conexiones más rápidas y sólidas, mayores distancias y mejores mecanismos de seguridad.

Bluetooth opera en el intervalo de frecuencias de 2,4 - 2,48 GHz. Los dispositivos que usan este tipo de comunicación se pueden clasificar respecto a su potencia de transmisión en “Clase 1”, “Clase 2” y “Clase 3”:

Tabla 6 Clases de dispositivos Bluetooth respecto a su potencia de transmisión

Clase	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Alcance aproximado
Clase 1	100 mW	20 dBm	100 metros
Clase 2	2.5 mW	4 dBm	5-10 metros
Clase 3	1 mW	0 dBm	1 metro

Una característica de esta clasificación es que si un sistema formado por un dispositivo de clase 2 y uno de clase 1, la cobertura efectiva del elemento de clase 2 se extenderá al conectarse con un transceptor de clase mayor, puesto que gracias a la mayor sensibilidad y potencia del de la clase mayor ofrecerá la posibilidad de recibir la señal a pesar de ser débil y que llegue la señal con más potencia al dispositivo menor.

Existe también la clasificación de dispositivos en relación a su capacidad de canal:

Tabla 7 Clasificación de dispositivos Bluetooth respecto a su ancho de banda

Versión	Ancho de banda
Versión 1.2	1 Mbit/s
Versión 2.0 +EDR	3 Mbit/s
Versión 3.0 + HS	24 Mbit/s
Versión 4.0	32 Mbit/s

Modo de funcionamiento general

Este protocolo se basa en el modo de operación maestro/esclavo, en el cual es el maestro el que elige con quién conectarse. El maestro es capaz de detectar hasta 7 dispositivos Bluetooth a su alrededor (denominados *piconets*), que poseen una dirección lógica de 3 bits.

El establecimiento de la conexión se establece siguiendo los siguientes pasos, proceso complicado para garantizar cierta seguridad:

1. Modo pasivo: Todos los dispositivos se mantienen en escucha ante una posible solicitud de conexión.
2. Solicitud: El maestro envía una solicitud a todo aquel dispositivo dentro de su rango de acceso (puntos de acceso), respondiendo éstos con su dirección.
3. Paginación: Elección por parte del maestro del dispositivo esclavo con el que conectarse, sincronizándose sus relojes y frecuencias.
4. Descubrimiento del servicio del punto de acceso: Se establece un enlace con el punto de acceso en el cual el maestro ingresa en la fase de descubrimiento del servicio mediante el protocolo SDP (*Service Discovery Protocol*).
5. Creación de un canal con el punto de acceso: Dependiendo del servicio, se establecerán un canal adicional.

6. Emparejamiento mediante el PIN (*Personal Identification Number*): Para dar cierta seguridad a la comunicación, sólo con los dispositivos autorizados se podrá establecer. Esta autorización es mediante un código PIN, enviado de forma cifrada al maestro
7. Utilización de la red: En caso de ser correcto el código PIN, la comunicación se establecerá libremente.

Modo de funcionamiento en dispositivos Android

La comunicación Bluetooth en dispositivos Android, aunque actúa del mismo modo que como se ha comentado anteriormente, existe la necesidad de explicar su propio método de funcionamiento, necesario de conocer para su programación en una aplicación.

La principal característica de este recurso es que trabaja internamente mediante *hilos* o *threads*, procesos en segundo plano. Éstos son una unidad de procesamiento que responde a los problemas de concurrencia, es decir, hace posible la ejecución de dos o más tareas que se necesitan llevar a cabo simultáneamente. En el caso de este tipo de dispositivos, cada uno cuenta con tres hilos distintos, dependiendo de su trabajo durante la comunicación:

- Hilo Servidor: Recibe solicitudes de conexión.
- Hilo Cliente: Envía solicitudes de conexión.
- Hilo Conexión: Intercambia datos.

Dependiendo del uso de un hilo u otro, hará necesario la utilización de diferentes clases como se muestra en la siguiente figura (para ésta explicación utilizaremos al dispositivo A como *cliente* y al dispositivo B como *servidor*):

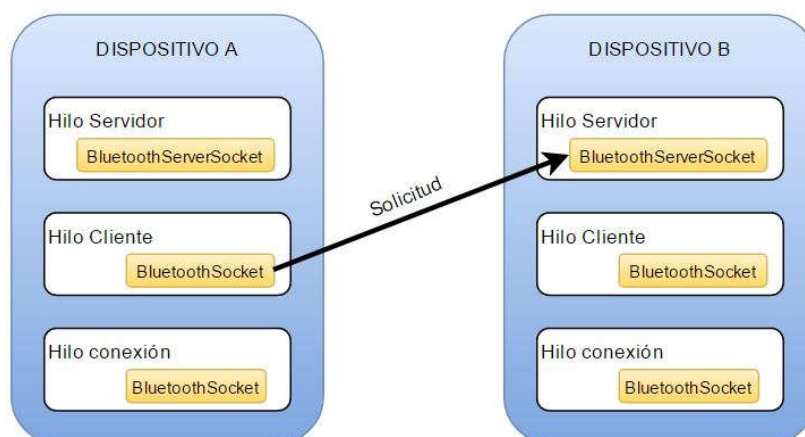


Figura 9 Solicitud de conexión en dispositivos Android

Como aclaración, un *socket* es un mecanismo utilizado en informática para hacer posible la comunicación entre dos elementos o programas distintos, existiendo *Socket* o *ServerSocket* si es cliente o servidor respectivamente.

El *ServerSocket* del servidor se mantendrá en un estado de escucha de solicitudes de conexión, mediante el método *accept()*, por parte de *Sockets* externos, siendo esta petición

con una demanda de contraseña o no, mediante el método *createInsecureRfcommSocketToServiceRecord(MY_UUID)* o *createRfcommSocketToServiceRecord(MY_UUID)*, dependiendo de la seguridad deseada. Dicha conexión o intento de conexión siempre será necesario identificarla, es decir, diferenciarlas de otras posibles uniones inalámbricas. Éste código identificador es el UUID (*Universally Unique Identifier*), código de 32 bytes.

Aceptada la conexión por parte del servidor, éste se lo notifica a su cliente, ambos creando un *Socket* en el hilo de conexión para el intercambio de datos juntos con sus objetos de la clase *InputStream* y *OutputStream* necesarios para la entrada y envío de datos.

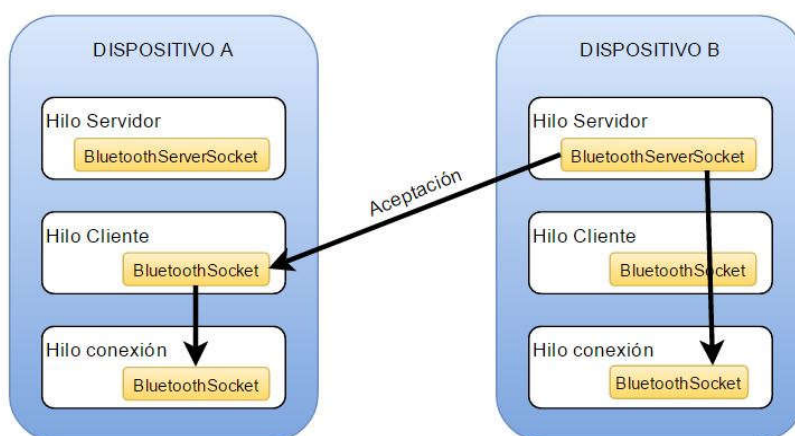


Figura 10 Inicio de conexión Bluetooth en dispositivos Android

Como se dijo al inicio del apartado, estos hilos trabajan en segundo plano, invisible a ser captado por el usuario de la aplicación, por lo que no es posible representar algo en la pantalla, que sería el hilo principal. Para responder a este problema, se hace uso de los *Handlers*. La instanciación de esta clase sirve de puente entre dos *threads*, pudiendo intercambiar datos.



Figura 11 Método de unión entre dos hilos

2.4 Programación de dispositivos

La interconexión de todos los elementos viene representada en el siguiente esquema, excluyendo el dispositivo Android, el cual va por libre sin una conexión física con los demás:

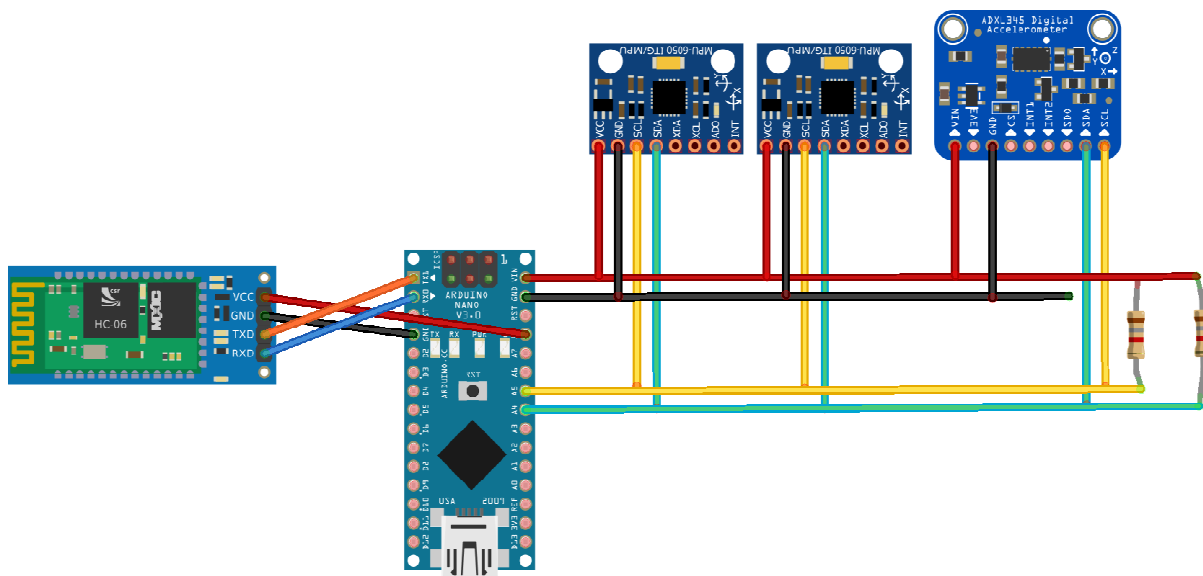


Figura 12 Conexión entre dispositivos

2.4.1 Arduino Nano

Configuración

Como resumen, se puede decir que el dispositivo Arduino Nano ha sido programado en un lenguaje orientado a objetos, en donde se ha creado la clase *sensor* para la instanciación de los diversos elementos de medición que estén conectados.

Además, se hace uso del puerto Serie configurándolo a una velocidad de 9600bps para la comunicación Bluetooth y la librería *Wire.h* para hacer posible la comunicación I²C.

Modo de funcionamiento

Respecto al método de trabajo, ya que su trabajo principal en el cual ocupa su mayor parte de tiempo es la de obtención de datos y envío, se ha elegido por un procedimiento continuo, es decir, en la entrada a un bucle de toma de datos a partir de los acelerómetros y envío de estos por el puerto Serie.

Sin embargo, existe la posibilidad del envío de datos o solicitudes (información, cambios en la configuración de acelerómetros, etc) desde el dispositivo Android a éste que hay que tenerlo en cuenta. Para ello, debido a que no es un proceso que se hace constantemente, es decir, no se produce en cada ciclo del bucle comentado en el párrafo anterior, se ha optado por un procedimiento de consulta periódica. Dicha consulta se efectuará posteriormente a la lectura de datos mediante el cuestionamiento de la existencia o no de datos de entrada en el puerto Serie.

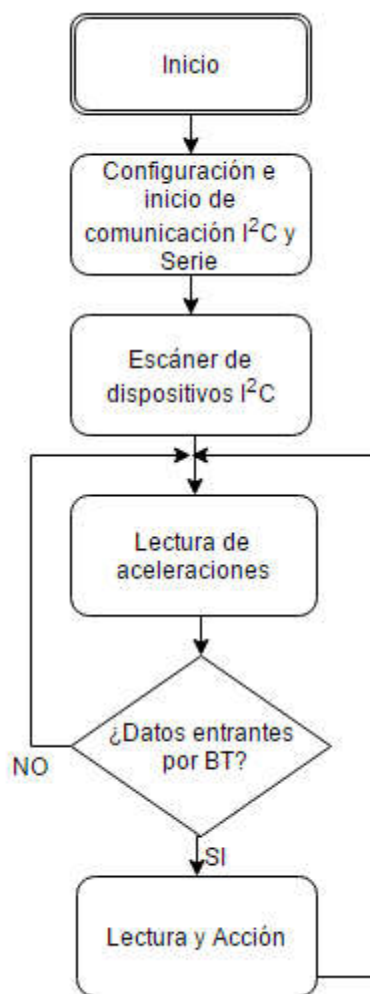


Figura 13 Diagrama de funcionamiento del programa de Arduino

Como se puede observar en el anterior diagrama de estados del proceso, el inicio de ejecución del programa cuenta con la configuración e iniciación del puerto Serie e I²C, seguido con una operación de escaneo o reconocimiento de dispositivos conectados mediante I²C como pasos previos a la entrada del bucle principal. El objetivo de esta etapa es la posibilidad de tener una mayor flexibilidad en cuanto al número y tipos de dispositivos conectados, y por lo tanto, poder usarse más como un programa general que por uno específico que dependiera del número y tipos de sensores, donde habría que hacer un programa para cada aplicación, teniendo en cuenta sus aparatos en la línea I²C.

En cambio, el problema de esta búsqueda de programa de aplicación general, si se busca la mayor eficacia, es la necesidad de búsqueda y programación de todos aquellos dispositivos aptos para la conexión, además de que cumplan la función requerida (en este caso, la toma de datos de aceleraciones). Como conclusión, sería un trabajo laborioso, además de limitado, puesto que hay un número máximo de direcciones, siendo muchas de ellas compartidas por distintos tipos de dispositivos, por lo que habría que hacer un estudio de compatibilidad de todos ellos, aumentando el volumen de dicho trabajo.

El proceso de escaneo de dispositivos I²C consiste en un intento de conexión en todo el rango de direcciones posibles y, en el caso de conexión, añadirlo al vector de acelerómetros creados y su configuración, teniendo en cuenta el tipo de acelerómetro que sea. La clasificación de los tipos de acelerómetros se realiza mediante el conocimiento de su dirección I²C. En este proyecto, los únicos tipos de acelerómetros creados serán los denominados *acelerómetro* y *giróscopo*, refiriéndose al acelerómetro ADXL345 y giróscopo MPU-6050 respectivamente.

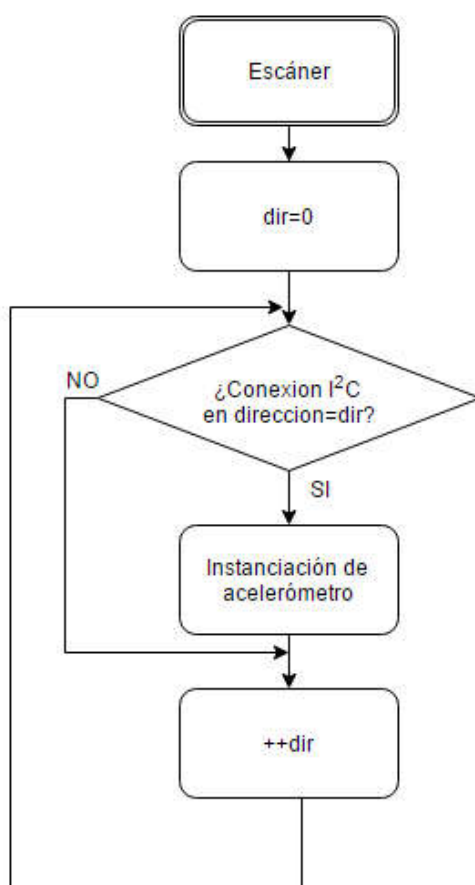


Figura 14 Diagrama de proceso de escaneo de dispositivos I²C

Posteriormente a este escaneo, dará lugar al inicio del proceso repetitivo principal, al de toma de datos. La ejecución de este trabajo consiste en dos pasos muy simples, pasos realizados de mediante un proceso reiterativo de evaluación de cada acelerómetro.

En primer lugar, se reciben los valores de las aceleraciones obtenidas por el acelerómetro y, en segundo, la creación de un *string* o cadena de caracteres que contenga la información de aceleraciones, la cual se enviará al módulo Bluetooth. Para hacer posible el reconocimiento de toda la cadena por parte del elemento receptor (en este caso, el dispositivo Android), se utiliza una serie de caracteres significativos, los cuales van a indicar la separación tanto entre las distintas cadenas de cada acelerómetro como los distintos valores o datos contenidos dentro de ellas. Su representación es el siguiente:

$$\{_id \$_x \$_y \$_z\}$$

Siendo:

- $_id$: número de identificación del dispositivo de medición.
- $_x$: valor de la aceleración en el eje X.
- $_y$: valor de la aceleración en el eje Y.
- $_z$: valor de la aceleración en el eje Z.

Como instrucción final en esta etapa, será la del envío de la cadena al módulo Bluetooth.

Como función adicional y tratada anteriormente, se cuenta también con un proceso de lectura en la comunicación Bluetooth por parte del Arduino, en caso de que el dispositivo externo requiera alguna información o busque cambiar la configuración de algún elemento conectado a la placa.

Para ello, como se dijo, se realizaba una consulta periódica al final del proceso de toma y envío de datos de aceleraciones, consulta en la que se comprueba si existe alguna entrada por el puerto Serie (por la línea de comunicación Bluetooth).

En caso afirmativo, se hará una evaluación de esta entrada puesto que, al igual que se hace para el envío de cadenas, la entrada también mantendrá un protocolo para su correcta lectura y actuar en función de ella. La parte efectiva de la cadena entrante estará contenida entre los caracteres ‘<’ y ‘>’, siendo posible distintos tipos de estructuras dependiendo del propósito de la solicitud, propósitos clasificados mediante palabras claves entre caracteres ‘#’:

- a) <#numdev#>: se solicita el envío del número de dispositivos conectados y su rango de escala al que están configurados cada uno de ellos. Su respuesta es mediante:

$$<num_{disp}:_id_1:direccion_1\$tipo_1\$rango_1\$id_2:direccion_2\$tipo_2\$rango_2\$...\>$$

- b) <#rango#_id\$rango_nuevo> : se solicita la reconfiguración del sensor $_id$ para que obtenga datos con un rango diferente.

En caso negativo, se omitirá el proceso de lectura.

2.4.2 Acelerómetro ADXL345

Como se dijo en el apartado anterior, se ha necesitado crear una clase *sensor* para un mejor manejo de ellos.

Éste contiene los siguientes atributos:

- *Id*: Identifica mediante un número al *sensor*.
- *direccion*: Representa su dirección I²C.

- *tipo_sensor*: Clasifica al elemento como acelerómetro o giróscopo.
- *g_range*: Guarda el rango de escala de aceleraciones.
- *divisor*: Valor numérico utilizado para representar los valores en fuerzas G dependiendo del rango de escala al que está programado.

Además, respecto a los métodos:

- *sensor()*: Método para la instanciación de la clase.
- *inicializacion_sensor (byte address, int n_device)*: Tras la instanciación, es la función que se encarga de configurar el dispositivo dependiendo del tipo que sea, a partir del estudio de *address*. En principio, siempre lo configura para tener un rango de $\pm 4g$.
- *standby_mode ()*: Método que modifica su estado a un modo en el cual no obtiene aceleraciones. Es necesario este estado para los cambios en registros.
- *active_mode()*: Método que modifica su estado a un modo en el cual sí obtiene aceleraciones.
- *programarRegistro(byte registro, byte valor)*: Método de cambio de valor en un registro.
- *lectura_registro(byte registro)*: Método de obtención del valor de un registro. Es usado a la hora de modificar sólo unos determinados bits de un registro, no alterando los demás.
- *get_range()*: Método de obtención del valor del rango de escala al que está configurado.
- *rango_g_cambio(char rango)*: Método de modificación del rango de escala.
- *get_id()*: Método de obtención del identificador del objeto.
- *lectura_datos (float *xyz_datos)*: Método encargado de obtener los valores de las aceleraciones y la obtención de su equivalencias en fuerzas g.

La ventaja de este acelerómetro es que es muy fácil su programación, obteniendo unos datos fiables con tan solo la configuración de dos registros únicamente: el registro POWER_CTL (0x2D) para la configuración de su modo (activo, standby, etc), y el registro DATA_FORMAT (0x31) para el rango de escala, siendo 0x00 para $\pm 2g$, 0x01 para $\pm 4g$, 0x02 para $\pm 8g$ y 0x03 para $\pm 16g$. Contiene la posibilidad de realizar más funciones que hace obtener unos datos más fiables, uso de filtros, entre otros, pero, evidentemente, dificultaría su programación.

La lectura de aceleraciones sigue el proceso siguiente:

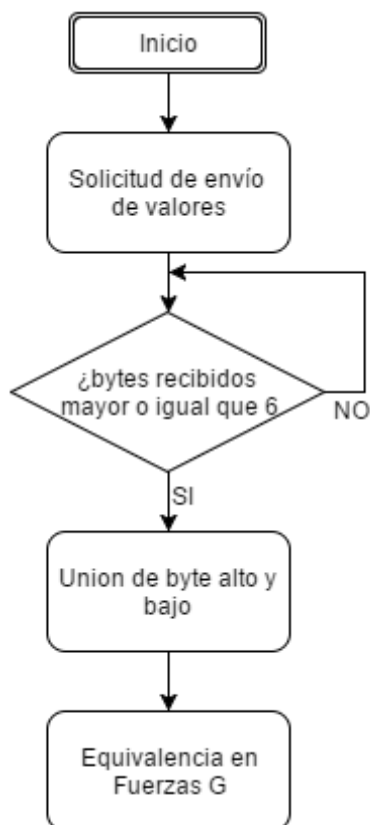


Figura 15 Método de obtención de aceleraciones

Como se muestra, para recibir los valores en los tres ejes, es necesario, como en cualquier paso para iniciar su comunicación I²C, la escritura de su dirección. Posteriormente, hecha efectiva la conexión, la escritura del registro X0 (0x32) como solicitud de recibirlos. Ante esto, el Arduino se mantiene en bloqueo hasta recibir los seis bytes de datos, continuando con la unión de los bytes alto y bajo para formar el valor real en cada eje.

Como paso siguiente, se realiza la equivalencia a fuerzas G de los valores obtenidos mediante la sustitución en la siguiente ecuación (como ejemplo, se utilizará la equivalencia de la aceleración en el eje X):

$$Xg = \frac{x - (x_0 / \text{divisor})}{(x_+ - x_0) / \text{divisor}}$$

Siendo:

- x: el dato obtenido.
- x₊: representa el dato obtenido en posición vertical positiva al eje X, es decir, manteniendo a 1G el eje X.
- x₀: representa 0G. Se obtiene como el valor medio entre los datos obtenidos a 1G y -1G.

Todas las constantes han sido obtenidas durante la etapa de calibración del acelerómetro.

2.4.3 Giróscopo MPU-6050

La utilización de estos dispositivos se realiza de la misma forma que los acelerómetros ADXL345, con la instanciación de la clase *sensor*, por lo que mantendrán iguales atributos y métodos, organizando el modo de realización de dichas funciones mediante un análisis de su atributo *tipo_sensor*. Es obvio que, al ser distintos tipos de sensores, su configuración también lo será.

Los registros necesarios a configurar son el *PWR_MGMT_1*(0x6B) que maneja el estado del sensor (activo o pausado) y el reloj principalmente. Para la obtención de medidas, hace necesario colocarlo en modo activo y, en cuanto al reloj, se ha escogido el oscilador interno de 8MHz.

El siguiente registro es el *SMPLRT_DIV*(0x19) que determina la relación entre la velocidad de salida del giróscopo y la velocidad de muestreo. Este registro se ha configurado a 0x19 para obtener una velocidad de transmisión similar al de ADXL345.

El tercero a modificar es el *CONFIG*(0x1A), encargados de la determinación del Filtro de Paso Bajo, que en este caso se opta por un ancho de banda de 44 Hz y sincronizado con el eje X.

Por último, es el registro que configura el rango de escala, definiendo los máximos y mínimos de aceleración que puede llegar a medir. Se hace mediante el registro *ACCEL_CONFIG*(0x1C), que en este caso puede ser: 0x00 para $\pm 2g$, 0x08 para $\pm 4g$, 0x10 para $\pm 8g$ y 0x18 para $\pm 16g$.

Una diferencia de estos dispositivos respecto a los acelerómetros anteriormente descritos es que no necesitan estar en un modo *standby* o pausados para cambiar el valor de los registros.

2.4.4 Dispositivo Android

El trabajo con los valores de las aceleraciones y representación de resultados se hace desde el dispositivo Android, así como la posibilidad de cambios de configuración en los sensores conectados al Arduino. Para la realización de esto, se ha diseñado una sencilla aplicación cuyos objetivos a llegar son:

1. Elección del dispositivo Bluetooth a conectar y conexión.
2. Visionado de los valores de aceleraciones a tiempo real.
3. Representación de la historia de aceleraciones en un diagrama.
4. Cambios de configuración en los dispositivos de medida desde el dispositivo Android.
5. Finalización de la conexión Bluetooth.

Modo de funcionamiento de la aplicación

Principalmente, la aplicación se puede dividir en dos partes:

Parte 1) Cumple el objetivo 1.

Parte 2) Se encarga de lograr los puntos 2, 3, 4 y 5.

El funcionamiento de la aplicación puede decirse que mantiene fundamentalmente un camino continuo, es decir, sin ramificaciones, por lo que hay que seguir unos pasos obligatorios para la consecución del propósito esencial, el de obtener aceleraciones. Es por ello que, como proceso de explicación más detallada, se va a seguir este camino, paso a paso.

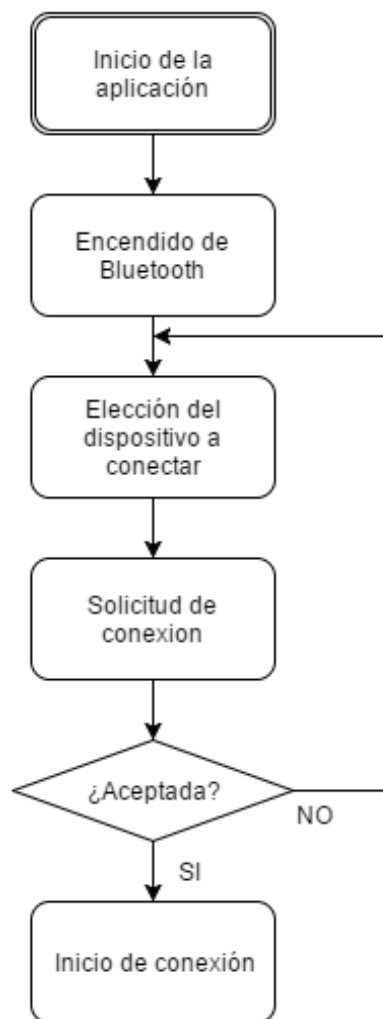


Figura 16 Funcionamiento de la primera parte de la aplicación Android

Como etapa inicial, se puede decir que es la correspondiente a la elección del módulo Bluetooth con el que se desea conectar y proceso de solicitud de conexión. Además, incluye la posibilidad de habilitación o inhabilitación del adaptador Bluetooth. En caso de no contener este tipo de adaptador, se avisará mediante una ventana emergente y el posterior cierre de la aplicación.

Para la elección del elemento Bluetooth con el que poder conectarse, se hace uso de un `ArrayList<BluetoothDevice>`, donde está almacenada toda la información de los dispositivos actualmente emparejados, y de un `ListView` o lista de elementos donde es posible ver y elegir el dispositivo deseado de entre los que contiene el array anterior.

Elegido dicho dispositivo, se inicia la creación y procesos de trabajo de los *threads* explicados anteriormente en el método de creación de una conexión Bluetooth en Android. Para ello, la creación de cada hilo en este dispositivo se ha hecho mediante la construcción de una clase derivada de la clase *Thread*. Sin embargo, como el módulo Bluetooth conectado al Arduino sólo puede admitir solicitudes y no enviar (sólo actúa como servidor), los únicos hilos creados en la aplicación son el de *Cliente* y *Conexión*.

La clase que da origen al hilo *Cliente* se ha denominado *ConectarThread.java*, la cual pide la solicitud de conexión, y en caso de ser aceptada por el elemento *Servidor*, dará paso a la creación del hilo *Conexión* con la instanciación de la clase *ThreadConectado.java*, con el paso a la segunda parte de la aplicación.

Como resumen de esta última clase mencionada, se puede decir que contiene dos atributos de la clase *InputStream* y *OutputStream*, que serán los encargados de recibir y enviar la información deseada a través del hilo mediante la iniciación de los métodos *run()* y *write()* de la clase *ThreadConectado* y llevadas al hilo principal mediante el *Handler*.

El paso a esta parte lleva consigo una nueva pantalla, así como un cambio en la forma de ejecución de los procesos, que se mantendrá en un bucle con la obtención de aceleraciones hasta el cierre de la aplicación. Sin embargo, es necesaria en primer lugar la obtención de una serie de información de los sensores conectados al Arduino, con el objetivo de mejorar la eficiencia y ser más flexible por ejemplo, ante cambios del número de sensores con el objetivo de disminuir tanto el uso de memoria como de tiempo de ejecución en procesos. Por lo que su camino seguido es el siguiente:

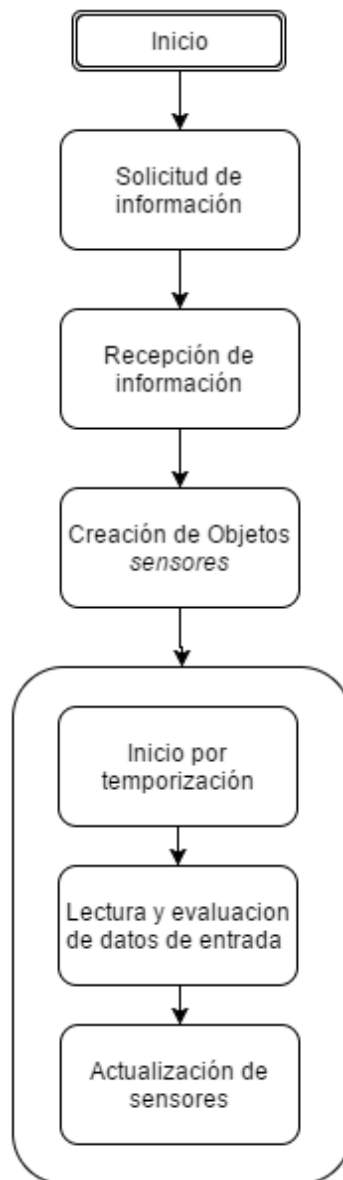


Figura 17 Funcionamiento de la segunda parte de la aplicación Android

Es por ello que, el primer paso dado en esta etapa es la solicitud de información mediante el envío del string `<#numdev#>` como se explicó en el apartado del modo de funcionamiento del programa Arduino. Respondiendo éste con el número de acelerómetros e información de cada uno de ellos sobre su id, dirección I²C, tipo de dispositivo y rango de escala al que está configurado. Este proceso será bloqueante, es decir, permanecerá en éste hasta que el Arduino le haya respondido con dicha información completa. En caso de error, se reinicia este proceso.

Siendo efectiva la llegada, el trabajo siguiente es el de evaluación de la cadena recibida para la creación de los objetos de la clase *sensor.java* con los cuales trabajar con ellos y guardar la información de cada uno de ellos, así como el historial de sus aceleraciones.

Posterior a esto, se da entrada al proceso de obtención de aceleraciones, que serán mostradas en un *ListView* que, para personalizarlo, se ha creado una clase derivada de

BaseAdapter, *list_view_acelerometros*, que está vinculado al anterior *Listview*, haciendo posible su visionado y actualización mediante su método *getView()*, sin necesidad de repintar toda la lista de elementos, proceso que consumiría mucho tiempo. Dicha entrada será temporizada cada cierto intervalo de tiempo en lugar de continua, provocando un proceso en bucle, ya que produce un gran bloqueo y que no se puedan efectuar una serie de tareas de interacción con el usuario.

En el *Listview*, además de los valores de las aceleraciones en tiempo real, es posible visionado del historial en un gráfico dinámico XY de ellos mediante un click corto en él, así como su desaparición, mediante la programación del método *onItemClickListener()* de la clase *ListView*. Este gráfico es desarrollado con el recurso *XYPlot* que ofrece la librería *JFreeChart*. En él, se refleja los últimos veinte valores de las aceleraciones, en los tres ejes, cada uno diferenciado con un color distinto, con límites superior e inferior determinados por el rango de escala al que está configurado el sensor.

Además, se comentó la posibilidad de realizar cambios en la configuración de los elementos conectados al Arduino desde el dispositivo Android. En esta aplicación sólo interesa cambiar los rangos de aceleración. Dicho cambio se realiza mediante un click largo esta vez, con la programación del método *setOnItemLongClickListener()* del *ListView*, el cual nos mostrará la serie de cambios disponibles a hacer mediante un menú *Popup*. Elegido una opción de éstos, dará lugar al inicio del proceso de enviar datos por el hilo *Conexión* del string con formato *<#rango#_id\$rango_nuevo>* siendo *_id* el identificador del sensor a cambiar y *rango_nuevo*, el rango deseado.

Como último objetivo importante de esta aplicación era el del cierre de ella, siendo posible con la pulsación del botón inferior “APAGAR”, que además, rompe la comunicación Bluetooth mediante la utilización del método *cancel()* de la clase *ThreadConectado*.

3 RESULTADOS Y DISCUSIÓN

Esta parte sirve de comentario sobre la eficiencia del método de resolución desarrollado en este proyecto al problema que se planteó inicialmente. Como fundamento de esta explicación, se hace una evaluación sobre la consecución de los objetivos que se propusieron en una etapa previa. Además, se hace una vista al futuro, en el que se proponen ideas que podrían mejorar el proceso, incrementando su eficiencia durante la ejecución y obteniendo unos resultados con mayor calidad o, incluso, óptimos.

Por último, es importante que todo proyecto no esté cerrado a una sola idea, es decir, a la resolución del problema planteado. Es necesario seguir caminos distintos por los que seguir regenerando nuestro trabajo. Por lo que, como punto final, se hará una descripción de distintas propuestas que pueden ser desarrolladas a partir de esta base propuesta, tanto como sistema conjunto como con la utilización únicamente de varios elementos que contiene.

3.1 Cumplimiento de objetivos

Como recordatorio de los objetivos propuestos a llegar en el desarrollo de este proyecto en el apartado 1.2, eran:

1. Lectura en tiempo real de aceleraciones.
2. Optimización del peso añadido.
3. Rapidez de comunicación.
4. Fiabilidad de datos obtenidos.
5. Visualización de datos en un dispositivo.
6. Precio bajo pero fiable de dispositivos.

El objetivo número 1 ha sido logrado puesto que era uno de las funciones principales que se buscaba para este trabajo. En este aspecto, ha sido posible la obtención de los valores de las aceleraciones a alta velocidad por parte del Arduino, así como su conversión a fuerzas g para un mejor entendimiento de estos valores. Sin embargo, aunque se obtienen unos datos aparentemente muy fiables, éstos pueden llegar a ser aún más con la configuración de filtros y otros aspectos que, como se comentó en la descripción de cada sensor, contienen cada uno.

Respecto al peso adicional en la raqueta por la instalación de todos elementos de control, se puede decir que el aumento es bajo, siempre teniendo en cuenta el número de sensores a colocar. En este proyecto sólo se han utilizado tres, siendo suficientes, con lo que la carga agregada es de unos 9gr por el Arduino Nano y módulo Bluetooth, más el de 3gr por cada sensor. El peso de cables y el de elementos de unión se puede despreciar si se eligen unos adecuados en este aspecto. Sin embargo, en la dinámica influye todo estos pesos así como su posicionamiento. Por lo que, en la etapa del estudio de instalación de ellos, es necesario conocer bien los parámetros de carga a llegar y distribución, puesto que afectarán a la inercia de la pala y, por lo tanto, a su movimiento.

Otro aspecto imprescindible en los sistemas de control es la velocidad. La velocidad en la comunicación I²C ya fue comentada anteriormente, declarando que era muy alta, por lo que válida. Además, en este proyecto también existe la velocidad de transmisión inalámbrica por Bluetooth, sin embargo, este tipo de comunicación es efectiva igualmente, muy rápida, siendo el problema que más le afecta, la posterior evaluación de los datos llegados y trabajo con ellos, por parte de los dispositivos interconectados, problema no propio de éste tipo de comunicación, si no del dispositivo.

En cuanto a la visualización de resultados, era también uno de los puntos esenciales en este trabajo, que en este caso se ha hecho en un dispositivo Android por su cercanía tanto de obtención como de utilización a toda persona. Mediante una sencilla aplicación instalada en el dispositivo y siguiendo varios pasos de conexión al aparato de medida, es posible conocer la variación de aceleraciones en ese mismo instante y su historia de valores en formato numérico y en un diagrama aceleración-tiempo respectivamente.

Finalmente, siempre hay que tener una consideración al gasto económico necesario para la puesta en marcha de un trabajo. En este proyecto, será el del tipo y número de dispositivos que se vayan a utilizar. Como se puede observar en el apartado de Presupuesto, no se necesita gran inversión en cuanto a hardware para hacer posible su implantación, inversión que variará principalmente en el número de sensores a utilizar, ya que, el precio de otras versiones de Arduino o módulo Bluetooth que pueden ser usadas, no varía en gran medida.

3.2 Posibles mejoras

Como se dijo al principio, este trabajo significa la base de un proyecto mayor en el que se quiere comprobar la posibilidad de obtención continua de aceleraciones y su transmisión mediante un sistema inalámbrico a algún dispositivo para su posterior visualización. Es por ello que, necesitará mejoras para el aumento de su eficiencia y, además, el aumento de funciones que lo haga un sistema más completo.

Un aspecto imprescindible de los sistemas de control en la obtención de datos es que éstos sean lo más semejantes a los valores reales. Sin embargo, existen perturbaciones que afectan a su obtención, significando un aumento de desigualdad entre ellos, es decir, existen ruidos o interferencias que afectan a los dispositivos electrónicos que provocan un error en sus valores con respecto a las medidas reales.

Para solucionar o atenuar el problema anterior se usan filtros. En el caso del acelerómetro ADXL345 es posible mediante el filtro de paso bajo, configurable en el dispositivo. En cambio, en los giróscopos, es posible el uso de un filtro mejor con la utilización tanto de aceleraciones como de giros. El más eficaz es el Filtro de Kalman usado en aviones y cohetes. Sin embargo, su desventaja es que tiene un coste de procesamiento elevado. Es por ello que lo sustituye el Filtro Complementario, que resuelve el aspecto anterior, además de ser fácil de entender y obteniendo precisos datos, siendo ésta su fórmula:

$$\text{Ángulo} = 0.98 * (\text{Ángulo} + \text{ÁnguloGiro} * t) + 0.02 * \text{ÁnguloAcel}$$

Siendo *ÁnguloAcel* el ángulo obtenido en la fórmula de la tangente de las aceleraciones.

La posibilidad de controlar todos los elementos del sistema desde el dispositivo Android siempre se ve como un punto a favor, por la mayor flexibilidad que eso supone y no tener que reprogramar el Arduino para cambiar ciertas configuraciones deseadas. Aunque respondiendo a esto, se ha conseguido implementar la posibilidad de cambiar el rango de escala de los acelerómetros, existen aún muchas otras características que podrían ser manejadas, como por ejemplo, el filtro anteriormente comentado o la frecuencia de muestreo, elemento muy influyente en el consumo de energía del sensor.

Tan importante es la capacidad de configurar todo aspecto involucrado en el sistema como el de obtener resultados que vayan más allá que el de saber sus aceleraciones. Una aplicación útil de la obtención de aceleraciones es llegar a conocer direcciones, por ejemplo. Lo que se quiere decir con todo esto es que una mejora puede ser la realización de estudios más profundos a partir de los datos obtenidos en cuanto a aspectos estáticos, dinámicos, energías, etc., con la llegada a conclusiones. Obviamente, todo este proceso desde el dispositivo Android debido a que trabaja a más velocidad que un Arduino. En caso contrario, siendo el Arduino el encargado de llegar a conclusiones, sería necesario transmitir por Bluetooth todas ellas al dispositivo Android, disminuyendo frecuencia de muestreo y aumentando la dificultad de comunicación entre ellos.

Finalmente, la apariencia de la aplicación es una cuestión interesante a estudiar tanto para el manejo de ella como para dar valor al producto. En cuanto al primer punto, se refiere a la estructura, que sea sencilla de usar. El segundo, va más encaminado al marketing, haciendo un diseño que sea visualmente aceptado y atractivo para el cliente.

3.3 Aspectos profesionales de futuro

Actualmente, todo evoluciona muy rápidamente, llegando a ser productos más eficientes y competitivos respecto a otros que cumplen la misma misión. Es por ello que todo proyecto a realizar, necesita una sección encargada al estudio de necesidades tanto actuales como de futuro, que ayuden a responder a estas variaciones, manteniéndose en un nivel alto de competitividad en el campo de actuación en el que trabaje su empresa. Es decir, la constante actualización es una prioridad.

El futuro de los elementos y comunicaciones puestos en marcha en este proyecto es amplio, siendo en estos días muy utilizado, en especial, los acelerómetros. Este tipo de proyecto en el que los elementos principales son los acelerómetros, la comunicación I²C y la comunicación Bluetooth, es posible su puesta en marcha en todas aquellas actividades en las que se necesita un constante control y envío de resultados a algún dispositivo de almacenamiento o *display*, para su posterior análisis.

Muchos profesionales en el deporte buscan la necesidad de este tipo de control. Es por ello que se han llegado a instalar en balones para el estudio de su trayectoria, movimientos o impactos. Incluso en algunos simuladores de deportes, como de golf. Un caso práctico interesante en el uso de este sistema creado podría ser las actividades de jabalina. Esta modalidad de lanzamiento es especialmente interesante ya que es vital conseguir que tenga unas determinadas velocidades y orientaciones durante su vuelo, si se desea obtener los óptimos resultados. Esto ayudaría tanto a los deportistas como a los fabricantes, que les ayudaría a fabricar unas jabalinas con mejor dinámica.

Sin embargo, su implantación no es única en los deportes. Tiene futuro en la medicina, como para el control de movimientos en articulaciones artificiales o estudio en personas, que actualmente está implantado en dispositivos que mantienen un conocimiento del comportamiento del corazón. O incluso en animales, que al igual, buscan un análisis de sus movimientos con el objetivo de creación de máquinas lo más semejantes a ellos.

Incluso en aparatos de uso diario, como en dispositivos de música o cascos con el uso de Bluetooth para el envío de la música junto a acelerómetros que captan ciertos movimientos para efectuar ciertos trabajos como el de cambio de pista.

Como utilización separada de los acelerómetros, su campo es muy amplio. El caso que actualmente tiene más importancia es en el uso en drones, que gracias a ellos es posible que se mantengan en una posición estable. La evolución de los drones ha sido espectacular en los últimos años, manteniendo la misma expectativa en los próximos, donde los acelerómetros serán decisivos. Es por ello que habrá que mejorar en su método de obtención de medidas y en la forma de transmitir las a su elemento de control, que hagan obtener datos con un error mínimo.

Además, se está poniendo en marcha en sistemas de alarmas por robo, en el que la aparición de fuerzas origina avisos, intentándose implementar en coches y viviendas. Un caso simple de alarma en coches ha sido implementación Vodafone Automotive, un dispositivo conectado al coche que está formado principalmente por un acelerómetro junto con una antena para la comunicación de alarmas, que avisará ante casos de movimientos por robos o golpes.

En cuanto a la comunicación Bluetooth, también es muy interesante, aunque existen otras muchas comunicaciones inalámbricas, pues contiene ventajas respecto a otras. Una de ellas es su bajo consumo energético. La cuestión energética es uno de los aspectos que más se están teniendo en cuenta, tanto para la mejora económica como medioambiental.

La respuesta a la cuestión en el hogar, con relación a la comunicación Bluetooth, puede ser las *viviendas inteligentes* o Domótica. En este tipo de casa se busca un control total del hogar, buscando los objetivos de ofrecer seguridad, confort, ahorro energético y comunicación. El papel que juega Bluetooth es la posibilidad de controlar y conectar todo dispositivo, junto con un ahorro en cables y su fácil instalación al ser un método inalámbrico, mejorando el aspecto económico, medioambiental y confort.

4 CONCLUSIONES

Finalizado el trabajo, con el logro de todos los objetivos, se ha llegado a la conclusión de que la comunicación Bluetooth es un método efectivo para la transferencia de datos en este tipo de sistema, en el que se necesitaba un control permanente de aceleraciones y su envío mediante este mecanismo a otro punto. Es un tipo de conexión en el que se puede llegar a unas velocidades muy altas de transmisión, bajo consumo de energía y, sobretodo, la mayoría de los dispositivos de control lo admiten o, en cambio, es fácil la conexión de un módulo Bluetooth para añadirle esta función.

Respecto a la comunicación I²C, consigue ejercer su trabajo con buenos resultados en este tipo de control. Además, es posible su utilización en otros tipos de comunicación entre distintos módulos en los que se busque el mínimo de peso adicional, con la única desventaja de no poder implementarse entre elementos ubicados a larga distancia.

En tercer lugar, los acelerómetros son unos sensores que están en auge por su gran capacidad de control de movimiento en objetos, en el que en muchos aparatos ya incluyen al menos uno por su facilidad de control y las ventajas que aporta. Incluso optan por utilizar giróscopos por el aumento de opciones de medidas que se pueden obtener. Aunque en este proyecto no ha sido posible su valoración para el fin último que se desea para este proyecto, el de golpear con la pala de pádel, se ha podido observar de una manera más superficial que pueden obtenerse valores cercanos a los reales siempre que se realice una configuración buena, con el uso de filtros para los ruidos, frecuencia de muestreo específico a la actividad a controlar, además de una buena calibración.

Además, cabe reseñar que el control de un dispositivo como de una actividad es una operación fundamental para evitar futuras roturas o daños que provoquen graves problemas o para obtener los mejores resultados posibles. El control está basado en la toma de datos, su representación visual (que puede ser opcional) y la toma de decisiones. Esta última pudiendo ser de acuerdo al deseo de una persona, por lo que hace necesario la representación, o respecto a una serie de normas que hace actuar automáticamente. Por lo que muchas máquinas están implementando un cierto control por los mejores resultados que obtienen con su actuación.

Como comentario final, la consecución de los objetivos de este proyecto abre muchas puertas a líneas futuras, tanto en la aplicación inicial que se buscaba, como en otras muchas en el campo deportivo, de la medicina o entretenimiento, entre otras.

5 BIBLIOGRAFÍA

Comunicación I²C

Vicente García. 2012. Introducción al Bus I²C. Electrónica Práctica Aplicada.
<http://www.diarioelectronicohoy.com/blog/introduccion-al-i2c-bus>

Eduardo J. Carletti. 2007. Comunicación – Bus I²C. Robots Argentina.
http://robots-argentina.com.ar/Comunicacion_busI2C.htm

Antonio Moreno Fernández. 2004. Tema 5: El bus I²C.

Comunicación Bluetooth

Sergio Alberto González Vergara. 2008. Tesis de Grado en Ingeniería en Comunicaciones y Electrónica: Tecnología Bluetooth.

Arduino

Óscar Torrente Artero. 2013. ARDUINO. Curso práctico de formación.

José Manuel Ruiz Gutiérrez. 2012. Manejo y Aplicaciones del Bus I²C de Arduino.

Brian Evans. 2007. Arduino Programming Notebook.

Acelerómetro ADXL345

Analog Devices. Datasheet Digital Accelerometer ADXL345. Descargable en:
<http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf>

Giróscopo MPU 6050

InvenSense. Datasheet MPU-6050. Descargable en:
<https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>

Módulo Bluetooth HC-06

Wavesen. Datasheet Bluetooth Module. Model: HC-06. Descargable en:
http://www.wavesen.com/mysys/db_picture/news3/2015121885021101.pdf

Android

Joan Ribas Lequerica, 2016. Desarrollo de Aplicaciones Para Android.

Daniel García. Let's code something up! - Code once, run you fools!.
<https://danielggarcia.wordpress.com/>

Android. Android Developers. <https://developer.android.com/>

JFree. XYPlot (JFreeChart Class Library (version 1.0.19-fx))
<http://www.jfree.org/jfreechart/api/javadoc/org/jfree/chart/plot/XYPlot.html>

6 ANEXOS

6.1 Presupuesto y planificación temporal

6.1.1 Presupuesto

COSTE PERSONAL

Nombre	Categoría	Euros/hora	Horas Totales	Coste Total (€)
Pablo Antonio Quesada Aguilera	Ingeniero Junior	20	354	7080

COSTES DE ELEMENTOS HARDWARE

Producto	Tienda	Precio (€)	Unidades	Coste Total (€)
Arduino Nano v3.0 Atmega328P	Amazon	21,5	1	21,5
Módulo Bluetooth HC-06	pccomponentes.com	12,95	1	12,95
Acelerómetro ADXL345	electan.com	5,75	2	11,5
Giróscopo MPU-6050	pccomponentes.com	9,95	1	9,95
Cables	pccomponentes.com	2,9	1	2,9
Dispositivo Android	Tienda de móviles	100	1	100
			Total	158,8

COSTE TOTAL DEL PROYECTO (€) **7238,8**

6.1.2 Estructura de Descomposición del Proyecto

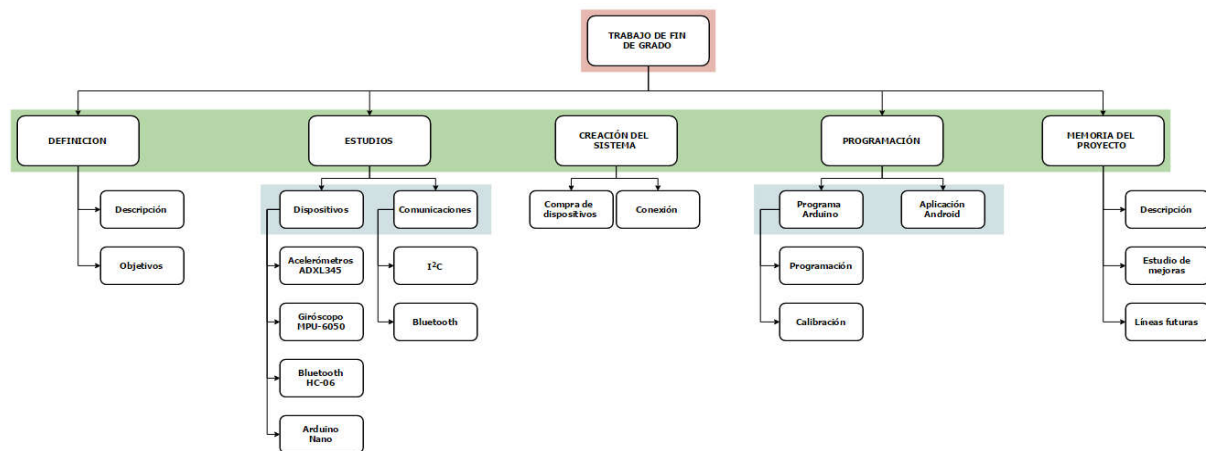


Figura 18 Diagrama EDT

6.1.3 Diagrama de Gant

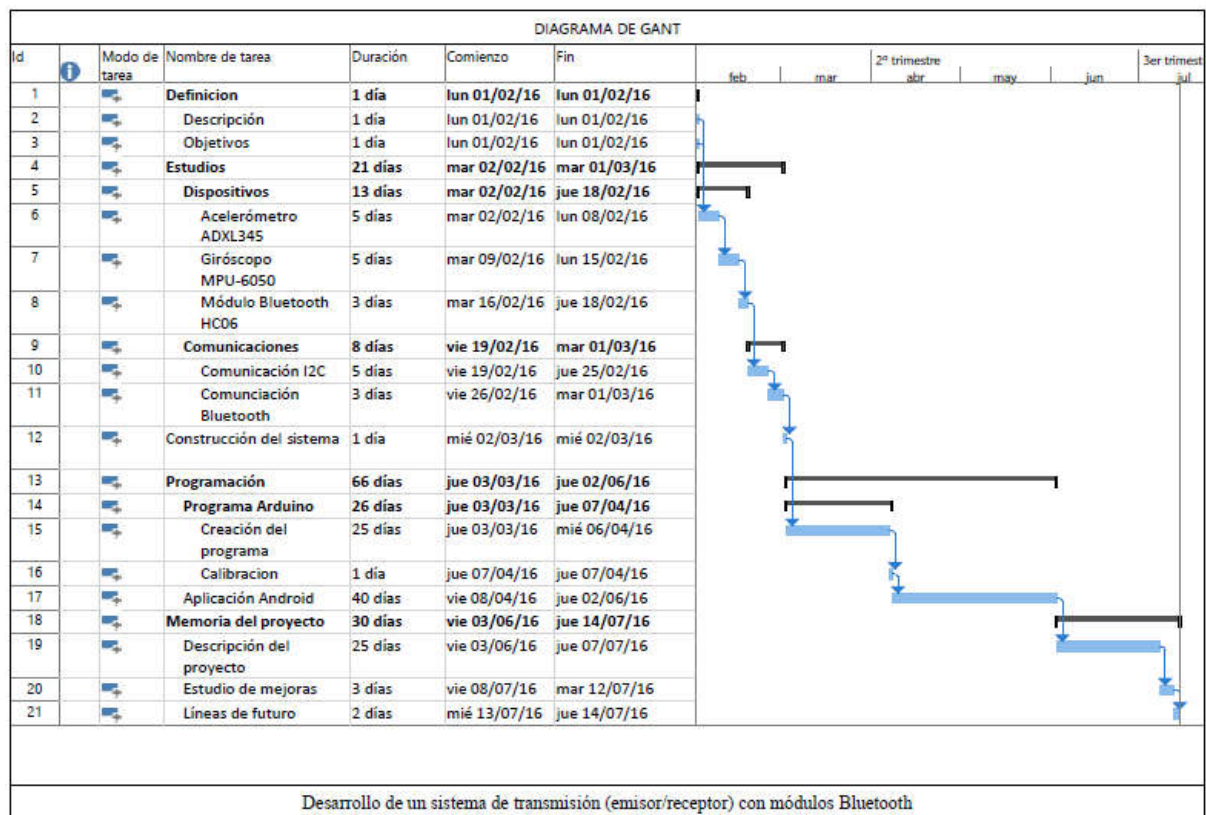


Figura 19 Diagrama de Gant

6.2 Índice de figuras

Figura 1 Arduino Nano	13
Figura 2 Diagrama de pines del Arduino Nano	14
Figura 3 Acelerómetro ADXL345	16
Figura 4 Giróscopo	18
Figura 5 Giróscopo MPU-6050	19
Figura 6 Módulo Bluetooth HC-06	21
Figura 7 Secuencia de inicio en I ² C	24
Figura 8 Secuencia de parada en I ² C	25
Figura 9 Solicitud de conexión en dispositivos Android	27
Figura 10 Inicio de conexión Bluetooth en dispositivos Android	28
Figura 11 Método de unión entre dos hilos	28
Figura 12 Conexión entre dispositivos	29
Figura 13 Diagrama de funcionamiento del programa de Arduino	30
Figura 14 Diagrama de proceso de escaneo de dispositivos I ² C	31
Figura 15 Método de obtención de aceleraciones	34
Figura 16 Funcionamiento de la primera parte de la aplicación Android	36
Figura 17 Funcionamiento de la segunda parte de la aplicación Android	38
Figura 18 Diagrama EDT	48
Figura 19 Diagrama de Gant	48

6.3 Índice de tablas

Tabla 1 Características principales del Arduino Nano	14
Tabla 2 Consumo energético en el modo activo	17
Tabla 3 Consumo energético en el modo ahorro	17
Tabla 4 Sensibilidad en la medida de la velocidad de giro del giróscopo MPU-6050	20
Tabla 5 Sensibilidad en la medida de la aceleración del giróscopo MPU-6050	20
Tabla 6 Clases de dispositivos Bluetooth respecto a su potencia de transmisión	26
Tabla 7 Clasificación de dispositivos Bluetooth respecto a su ancho de banda	26

6.4 Abreviaturas, unidades y acrónimos

Abreviaturas

etc. Etcétera

TFG Trabajo de Fin de Grado

Unidades

kBits	kiloBits
k Ω	kiloOhmios
pF	picoFaradios
mW	miliVatios
dBm	decibelio por miliVatio
m	metros
bps	bits por segundo
gr	gramos
°	grados
dps	grados por segundo
°C	grados Celsius o centígrados
mA	miliamperios
seg	segundos

Acrónimos

ACK	Acknowledgement
ADC	Analog-Digital Converter
BER	Bit Error Rate
I ² C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit

IrDa	Infrared Data Association
LED	Light-Emitting Diode
PIN	Personal Identification Number
RAM	Random Access Memory
SCL	Serial Clock
SDA	Serial Data
SDP	Service Discovery Protocol
UART	Universal Asynchronous Receiver-Transmitter
UUID	Universally Unique Identifier
WIFI	Wireless Fidelity
WPAN	Wireless Personal Area Network

6.5 Glosario

Arduino	Plataforma de hardware y software de código abierto utilizado en electrónica en proyectos multidisciplinarios.
Array	Vector de elementos.
Bit-rate	Velocidad de transferencia de datos o número de bits que se transmiten por unidad de tiempo.
Bluetooth	Tipo de mecanismo de comunicación inalámbrica.
Bus Serie	Mecanismo de envío de datos bit a bit a través de una línea.
Display	Pantalla o indicador numérico utilizado para visualizar una determinada información de un aparato electrónico.
Handler	En Android, elemento que permite la comunicación entre dos hilos distintos.
Java	Tipo de lenguaje de programación orientada a objetos.
ListView	En Android, lista de elementos.
Piconets	En comunicación Bluetooth, denominados a los distintos dispositivos a los que se puede establecer la conexión.
Processing	Lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java.
Protocolo	Conjunto de normas establecidas en la secuencia de mensajes en una comunicación.
Puerto Serie	Elemento de conexión entre dos sistemas cuya transmisión de información es bit a bit.
Sensor	Dispositivo electrónico capaz de medir una magnitud física y convertir la medida en una señal digital.
Socket	Punto final de una comunicación bidireccional entre dos programas que intercambian información.
String	Cadena de caracteres.
Thread	En Android, hilos o procesos en distinto plano de ejecución que el principal.

6.6 Código fuente

Tanto el código Arduino como la aplicación Android puede encontrarse en el depósito del Departamento de Electrónica de la Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Madrid.