

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Sensórica integrada en red de área personal para la práctica de CrossFit



Grado en Ingeniería
en Tecnologías de Telecomunicación

Trabajo Fin de Grado

Alejandro Jiménez Cornago

Francisco Javier Falcone Lanas

Pamplona, 25 de Enero de 2019

upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Agradecimientos

A mi familia que han estado apoyándome en todo este proceso y que sin sus ánimos jamás lo hubiera conseguido.

A mis compañeros de clase y amigos con los que he convivido durante este periodo y de los que he aprendido mucho.

A Francisco por animarme a idear un trabajo de fin de grado que estuviera relacionado con alguna de mis aficiones. A Peio por acompañarme el día de las mediciones.

A Pablo por ofrecerse voluntario para las mediciones y cedernos su tiempo e instalaciones de CrossFit Runa.

Abstract

In this project Zigbee technology is used through XBee modules to obtain data referring to movements of Crossfit sports practice, in particular weightlifting movements have been studied. The technology allows the inclusion of as many nodes located on the athlete as needed (up to 255).

The inclusion of this technology in this sport can improve user experience of athletes and spectators if they are included in official competitions.

Reducing its scale to the maximum, nodes could be obtained that could be placed in sports garments and that would not be annoying for the practice of this sport modality.

Tabla de contenido

Agradecimientos	1
Abstract	2
1 Introducción	5
2 Resumen y objetivos del proyecto	6
3 Palabras clave	7
4 Estado del arte	8
4.1 Antecedentes	8
4.1.1 Mallas inteligentes	8
4.1.2 Reconocimiento de tipo de ejercicio.....	9
4.1.3 Uso de encoder lineal en halterofilia	10
4.1.4 Estimación de Carga máxima en función de la velocidad de ejecución.....	11
4.2 Modelos comerciales	12
4.3 Conclusiones tras analizar la oferta y demanda.	14
5 Fundamentos teóricos	15
5.1 WPAN	15
5.2 Zigbee	15
5.2.1 Capa física (802.15.4)	16
5.2.2 Capa MAC (802.15.4).....	16
5.2.3 Capa de red (Zigbee)	17
5.2.4 Capa de aplicación (Zigbee).....	19
5.3 UART.....	19
6 Desarrollo hardware	21
6.1 Microcontrolador Arduino Lilypad USB plus.....	21
6.1.1 Lilypad USB Plus vs. Arduino Nano.....	21
6.2 Microcontrolador Arduino Uno.....	22
6.3 Conexión inalámbrica. Modulo Xbee	22
6.4 Sensor. Acelerómetro.....	25
6.4.1 Calibración acelerómetro.....	26
7 Desarrollo software	29
7.1 Arduino IDE	29
7.2 XCTU	31
7.3 Matlab	35
8 Primera conexión Zigbee punto a punto en modo transparente.	37

9	Conexión punto a punto en modo API	39
10	Prototipo funcional inalámbrico	43
10.1	Sistema de alimentación	43
10.2	Colocación en usuario	45
11	Primeras medidas de movimientos en usuario	47
11.1	Ejemplo de aplicación. Validación de repetición en competición de powerlifting.	48
12	Red multipunto	52
13	Adecuación del sistema para pruebas de campo.	54
13.1.1	GUI en Matlab	54
13.1.2	Sustitución de cables.....	55
14	CrossFit Runa	56
14.1	Resultados de mediciones.....	58
15	Conclusión	63
16	Líneas Futuras	64
17	Hojas de características	66
18	Referencia de imágenes	67
19	Bibliografía y referencias	69
20	Código Fuente	71
20.1	Microprocesador Lilypad USB plus (Router.ino)	71
20.2	Microprocesador Arduino Uno (CoordinatorXbee.ino)	71
20.3	GUI en Matlab (InterfazGrafica.mlapp).....	72
20.4	Función para leer los valores de la trama (xyzDataExtract.m).....	76

1 Introducción

El método de entrenamiento de Crossfit nace en los años 70 como el entrenamiento ideal para policías y militares norteamericanos, el cual, les permite estar en forma para poder afrontar diversas situaciones en condiciones adversas.

Crossfit incluye movimientos multiarticulares muy variados repetidos a alta intensidad en el menor tiempo y máxima potencia posible. Se combinan diferentes disciplinas como pueden ser halterofilia, movimientos gimnásticos, remo, bicicleta, etc. Y cada vez son añadidos más ejercicios, ya que no es un sistema cerrado.



Figura 1-1 Atleta en CrossFit Games
2015 [1]

El escaparate mundial de este deporte ocurre cada verano. Se trata de la competición llamada Crossfit Games, en la que durante 5 días se pone a prueba a atletas élite de diferentes partes del mundo con pruebas sorpresa. El título que obtiene el atleta que mejor resultados obtiene al finalizar este periodo es el "Fittest Man/Woman on Earth" o el hombre/mujer más en forma de la tierra.

Y es tras esto cuando empezó a popularizarse y a ser acogido por millones de atletas amateur que día a día lo practican. Y es que desde CrossFit animan a practicarlo sea cual sea tu estado físico, ya que se puede adaptar la intensidad a cualquiera persona.

Este entrenamiento se suele realizar en forma de clases grupales lo cual consigue que el intenso ejercicio físico sea más ameno e incluya un componente de competitividad que tanto gusta a quienes lo practican. Cuando un atleta termina el entrenamiento del día, este apunta el tiempo que le ha llevado en una pizarra que hace sus veces de tabla clasificatoria que se va completando conforme los demás deportistas van terminando también. Los más rápidos pueden presumir de tener su nombre en la parte más alta de la clasificación.

Si bien hasta ahora parece todo muy divertido, hay ocasiones en las que, sumando la fatiga acumulada, el alto peso por repetición y la máxima velocidad que se le intenta imprimir, pueden producirse lesiones.

Desde Crossfit se intenta que esto no suceda, supervisando que todos los centros que utilizan su sistema posean personal con títulos de entrenador Crossfit. Pero a veces esto no es suficiente y todo depende de que los atletas sean consecuentes y no crucen niveles peligrosos.

2 Resumen y objetivos del proyecto

El proyecto que se detalla a continuación consiste en buscar una aplicación directa de ingeniería en el campo anteriormente expuesto. Se intentará usar hardware del mínimo tamaño para la adquisición de datos en movimientos seleccionados que permitan un posterior análisis para la mejora de los atletas.

Se pretende conseguir un prototipo funcional que, mediante algún tipo de sensor, consiga recoger datos que sean enviados de forma inalámbrica para que puedan ser procesados.

3 Palabras clave

- CrossFit
- WPAN
- XBee
- Zigbee
- Arduino
- Acelerómetro
- Lilypad

4 Estado del arte

En el marco contextual en el que se desarrolla el proyecto se aprecia un notable incremento de la popularidad de los “*wearables*”, dispositivos electrónicos situados sobre el cuerpo del usuario y que realizan diversas funciones como la de interconectar dispositivos, medir la actividad física, adquirir datos biométricos e incluso la simple tarea de decorar prendas.

Una de las aplicaciones que más éxito ha tenido hasta ahora han sido las llamadas “*Smart bands*” o pulseras inteligentes. Estas son pequeños dispositivos compuestos por diferentes sensores que brindan al usuario información de su día a día o de sus entrenamientos.

A diferencia de los relojes inteligentes, éstas suelen estar más encaminadas a la adquisición de datos y su tecnología de interconexión suele ser más limitada. Normalmente solo permite al usuario navegar por los diferentes registros de datos en la propia pulsera o en un teléfono móvil mediante una aplicación. Esto hace que su precio sea menor, pudiéndose encontrar las más económicas por un precio de 30 euros o similar. Esto probablemente ha sido uno de los factores que han hecho que estos dispositivos hayan sido un éxito a la hora de acercar más la tecnología a la sociedad.

4.1 Antecedentes

Debido a las exigencias del deporte en categorías elite se empezó a buscar la forma de medir y correlacionar variables en diferentes disciplinas con el fin de encaminar los entrenamientos a obtener mejores resultados.

4.1.1 Mallas inteligentes

La inclusión de los dispositivos en la ropa deportiva ha estado siempre muy contemplada. Este tipo de prenda permite mantener los sistemas de adquisición pegados al cuerpo del atleta. Según el nivel de compresión también puede evitar que los sensores se muevan desembocando en medidas erróneas.

En la universidad de Universidad King’s College de Londres (Inglaterra) se utilizaron estas ventajas para desarrollar un prototipo de malla que, mediante un microcontrolador Arduino y una serie de electrodos distribuidos por la superficie de la prenda, fuera capaz de medir la fatiga muscular que se va acumulando conforme avanza la sesión de entrenamiento de carrera continua.

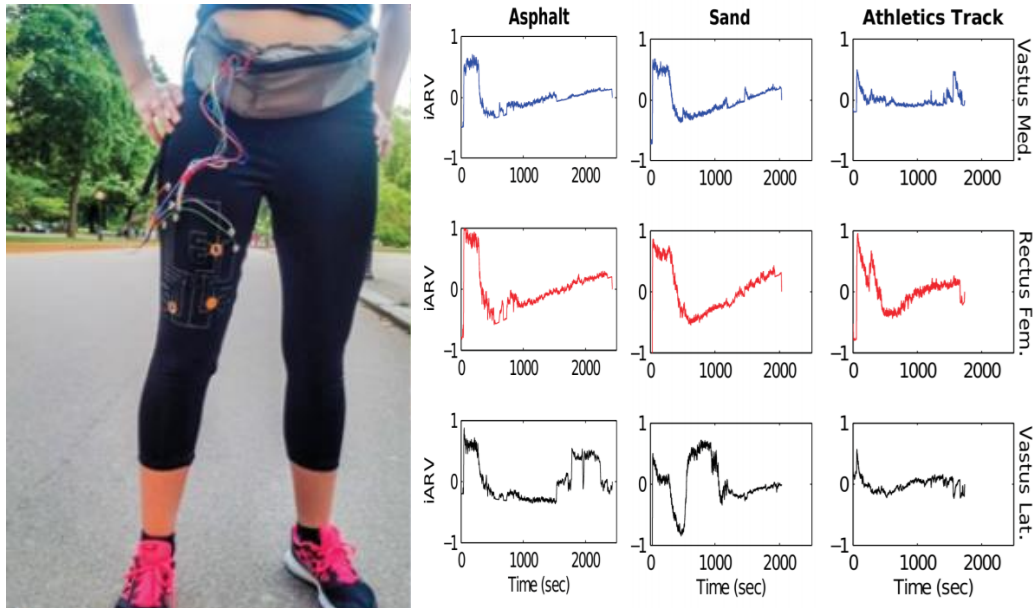


Figura 4-1 Prototipo y resultado de impulsos de los músculos vasto medio, vasto lateral y recto femoral en escenarios de asfalto, tierra y pista de atletismo [2]

4.1.2 Reconocimiento de tipo de ejercicio

Con diferentes medias de diferentes sensores es posible que el sistema reconozca que tipo de ejercicio está realizando el usuario que lo porta. Esto es lo que demostraron en la publicación “RecoFit: Using a Wearable Sensor to Find, Recognize, and Count Repetitive Exercises” (2014) en donde consiguieron reconocer el ejercicio y mostrar como respuesta el número de repeticiones sin que el usuario tuviera que intervenir.

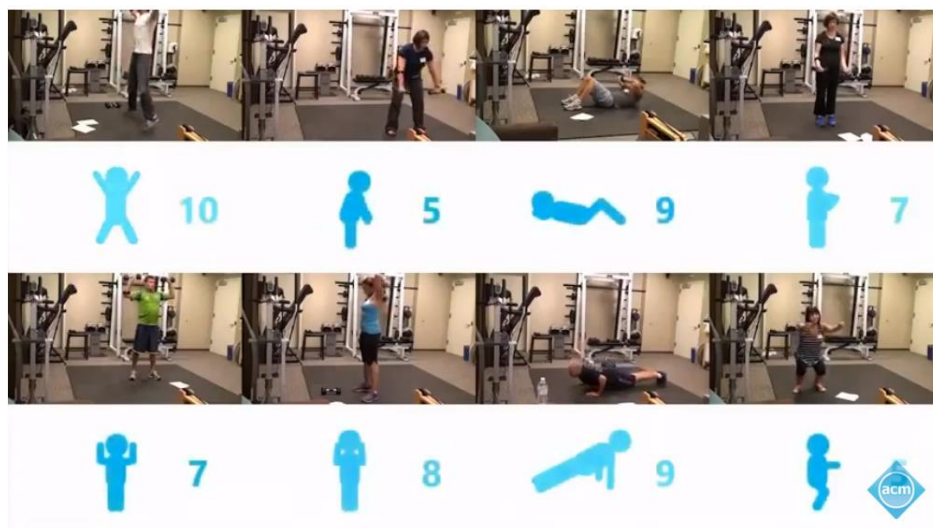


Figura 4-2 Diferentes ejercicios de peso corporal autoidentificados con sus correspondientes números de repeticiones [3]

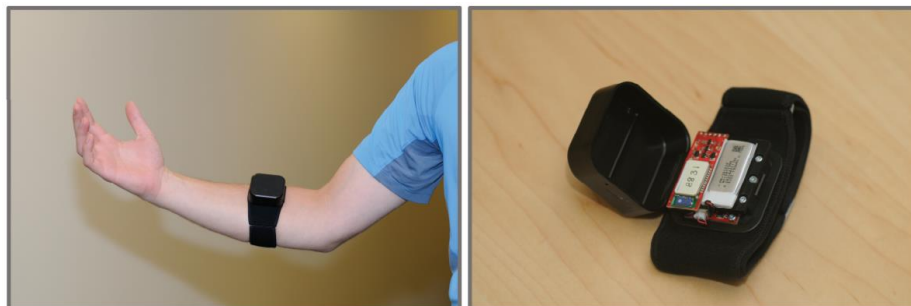


Figura 4-3 Hardware de adquisición colocado en el antebrazo del usuario mediante una cinta elástica [4]

Desarrollaron un algoritmo que identificaba los picos de la señal de aceleración y según su amplitud y frecuencia identificaba el ejercicio. Una vez identificado, las muestras utilizadas para el reconocimiento son convertidas a número de repeticiones que van aumentando conforme la señal sigue teniendo una morfología similar.

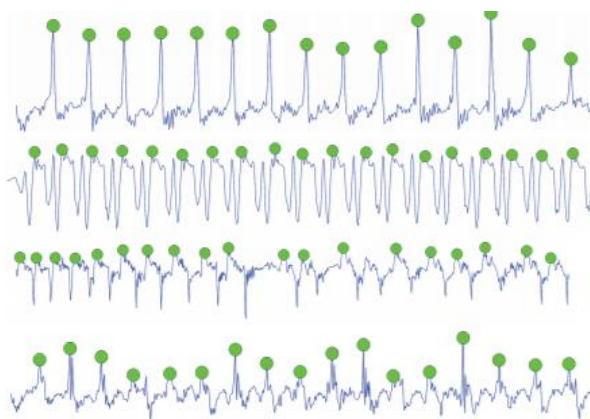


Figura 4-4 Identificación de los picos de las señales para identificar patrones [5]

4.1.3 Uso de encoder lineal en halterofilia

La halterofilia es una disciplina que también ha sido estudiada, ya que requiere de movimientos multiarticulares rápidos y precisos para poder levantar el peso deseado. Como se expuso en la introducción, CrossFit incluye con gran frecuencia esta modalidad deportiva, por lo que están muy relacionados al desarrollo de este proyecto.

Un ejemplo de estudio es *“Análisis de los parámetros de fuerza, potencia y velocidad de la barra en el arranque en levantadores de pesas juveniles de Las Tunas”* (2004) donde se utilizó un encoder lineal situado en el suelo y con el filamento sujeto a la barra. La velocidad de giro del sistema determinó la velocidad de levantamiento de varios usuarios, seguidamente se estudiaron las fases del movimiento y se compararon potencias según el perfil del levantador y la carga.



Figura 4-5 Ejemplo de uso de un encoder lineal en sentadilla trasera [6]

“Las variables cinemáticas y dinámicas del movimiento se determinaron por medio de un transductor lineal de movimiento (Real Power, Globus Italia) que consiste en un encoder lineal, rotatorio, que funciona con un sistema de dínamo, y consta de un registro mínimo de posición de 1 mm y un cable cuyo extremo se aseguró en un sitio específico de la barra de modo que no moleste la ejecución del ejercicio. El funcionamiento permite que cuando el sujeto realice el ejercicio el cable se desplace en forma vertical, según la dirección del movimiento, detectando e informando la posición de la barra cada 10 mls (10000 hz) a un interface conectado a un computadora donde el software REAL POWER 2001 versión J 62, calcula automáticamente los valores de fuerza, velocidad y potencia media y pico. (Naclerio A. F., 2001)”

4.1.4 Estimación de Carga máxima en función de la velocidad de ejecución

Otro factor que incrementó el interés por conocer más sobre este tipo entrenamientos fue la publicación del estudio realizado por Brzycki, M (1993) en el que se encontró una relación lineal entre la velocidad de ejecución y el porcentaje de peso máximo que un atleta podría levantar una vez, resultando fallido la segunda repetición consecutiva si se realizase. Este valor se llama también 1RM.

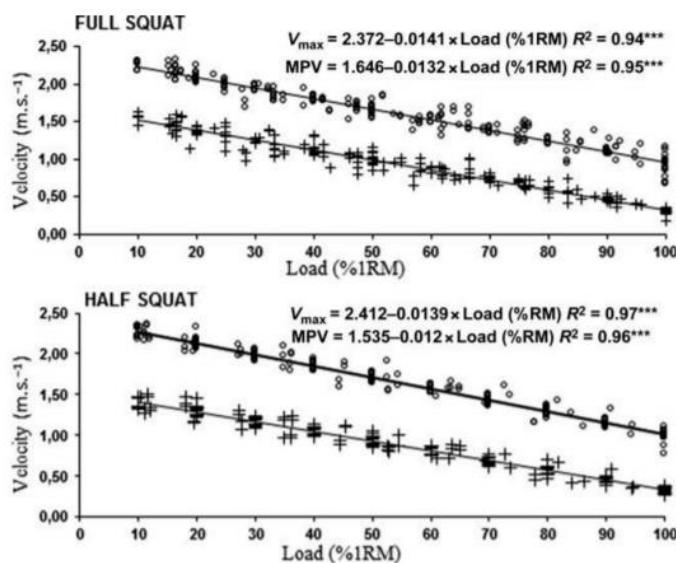


Figura 4-6 Regresiones lineales entre la velocidad de ejecución y la carga medida en porcentaje de repetición máxima en sentadilla profunda y sentadilla de 90° [7]

4-1

A partir de este estudio se empezaron a popularizarse los dispositivos que permitieran saber la velocidad de ejecución de movimientos con cargas sub-máximas para calcular de forma aproximada el peso máximo anteriormente.

La ventaja de este método es que permite calcular este valor de forma diaria, ya que varía debido a muchos factores, sin que el deportista tenga que exponerse a altas cargas muy frecuentemente, lo que podría resultar peligroso.

4.2 Modelos comerciales

- **Speed4lifts**

Se trata de un encoder lineal que dos jóvenes ingenieros españoles desarrollaron para usar en sus entrenamientos debido al alto precio de estos productos. La alta demanda en su círculo cercano los animó a comercializar su solución.

Uno de los inconvenientes que pudiera tener este dispositivo es que debe de ser colocado justo debajo del levantador y con el filamento unido a la barra en todo momento. Además, debe de ser fijado a una superficie metálica en el suelo para que el poco peso del dispositivo no produzca movimientos que estropeen la medición.



Figura 4-7 Encoder lineal Speed4lifts [8]

Normalmente en los gimnasios donde se realiza Crossfit se dispone de un suelo adaptado para amortiguar el impacto de los discos, por lo que un levantamiento de alto peso se acaba soltando la barra desde la posición más elevada y dejándola caer. Este dispositivo correría alto riesgo en esos momentos ya que podría ser dañado.

Su precio, aunque no siendo el más elevado de este tipo de dispositivos, se encuentra actualmente en 365 euros por compra mediante página web.

- **BeastSensor**

A diferencia del anterior, se trata de un módulo totalmente inalámbrico, de pequeño tamaño que mide velocidad de ejecución de diferentes movimientos. El precio del sensor es de 279 euros.

También existen packs destinados entrenamientos grupales, los cuales están diseñados para ser colocados cada uno en una estación, barra o máquina mediante el imán que viene incorporado. El atleta acudiría al teléfono móvil vinculado con dicho ejercicio, elegiría su perfil con un simple toque y a continuación realizaría el ejercicio. Cuenta con un sistema de tablero de clasificación para aumentar la competitividad de los atletas.



Figura 4-8 Beast Strength Pack 5 [9]

El coste de estos packs es de 499 ,944 y 1699 euros según el número de sensores y usuarios. Por ejemplo, la opción de precio menor estaría compuesta por un único sensor con capacidad de 10 perfiles de usuario mientras que la más cara incluye 5 sensores que pueden ser usados por 30 usuarios por entrenamiento.

Como aspecto negativo, el sensor a veces realiza medidas incorrectas de pequeños movimientos que tienes que eliminar a posteriori. Además, no es capaz de percibir si la técnica de ejecución del movimiento es correcta o si se producen descompensaciones en la simétrica del ejercicio.

- **PushBand**

PushBand sigue en la línea del anterior sensor. Está formado por un módulo compuesto por un acelerómetro, un giroscopio y conexión bluetooth. Se sitúa en una banda elástica con velcro y se coloca en el antebrazo o en el bíceps en función del ejercicio a realizar. También se puede colocar en la barra con una correa. El módulo y correas tiene un precio de 349 dólares.

Está diseñado para mostrar al usuario la velocidad pico y media del levantamiento, ayudándole así a planificar mejor su entrenamiento.

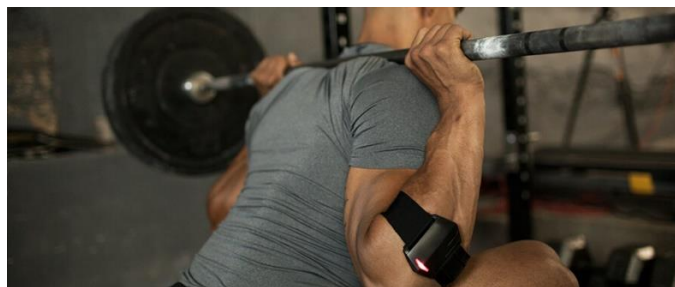


Figura 4-9 Atleta realizando un levantamiento con PushBand [10]

Uno de los apartados fuertes de este sistema es que se ha desarrollado una plataforma que permite a los entrenadores hacer un seguimiento de sus atletas. Esta opción es de pago adicional. Su coste depende del número de atletas y el nivel de estos, y se sitúa en un rango de 100 a 990 dólares (más de 3 atletas estándar y más de 50 atletas elite respectivamente).

Como aspecto negativo, no es muy preciso y en ocasiones registra más repeticiones de las realizadas, por lo que tienes que eliminarlas. Esto recalcula los parámetros devolviendo una estimación.

4.3 Conclusiones tras analizar la oferta y demanda.

Como se puede apreciar, existe gran demanda en cuanto a soluciones que permitan conocer más sobre este tipo de ejercicio físico. Lo que todavía no se ha desarrollado es un sistema WPAN que permita mediciones más precisas y de distinto tipo mediante varios nodos situados en el cuerpo o ropa del atleta. Y esto es lo que se presenta en este trabajo de fin de grado.

5 Fundamentos teóricos

5.1 WPAN

Las llamadas WPAN, acrónimo de “Wireless Personal Area Network” son redes inalámbricas de unos pocos metros de distancia, rango llamado espacio personal. Se desarrolló para eliminar cables entre ordenadores, impresoras y PDAs principalmente.

El grupo de trabajo encargado de desarrollar esta tecnología es el IEEE 802.15 y sus dos áreas de trabajo de uso más popular son los estándares 802.15.1 y 802.15.4. En ellos se basan tecnologías de uso cotidiano como son Bluetooth y Zigbee.



Figura 5-1 Logo IEEE 802.15 [11]

5.2 Zigbee

Si bien en un principio se planteó la opción de utilizar tecnología Bluetooth, determinadas características, como la diferencia de consumo de potencia o de número de nodos posibles por red, inclinaron la balanza hacia la elección de Zigbee.

Se trata de una tecnología de comunicación inalámbrica de bajo consumo y velocidad de transmisión. Fue ideada en un principio para emisión de simples órdenes a distancias típicamente de 10 metros, aunque ha seguido desarrollándose pudiendo soportar conexiones de hasta 1600 metros.

Esta baja complejidad de datos permite no necesitar alto ratio de envío. La velocidad típica de 250 kbps permite satisfacer la mayoría de las aplicaciones, pudiendo ser en ocasiones reducida a 20 kbps en caso de necesidades de sensores y de automatización.

Zigbee, es implementado por encima de las capas físicas y de enlace del estándar IEEE 802.15.4, añadiendo la capa de red para aportar funciones de enrutamiento.

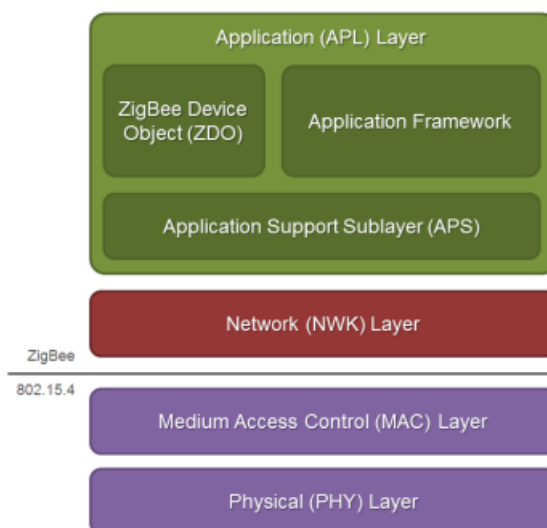


Figura 5-2 Modelo de capas de Zigbee [12]

5.2.1 Capa física (802.15.4)

El estándar 800.15.4 establece 16 canales de 5 MHz en la banda frecuencial 2.4 GHz (canales 11-26). También existe la posibilidad de uso de la banda 868 MHz y 915 MHz aunque tienen restricciones según el país. Por ello los fabricantes prefieren la banda 2.4 GHz, ya que es libre en todo el mundo.

Para la transmisión en esta banda se utiliza la codificación de espectro ensanchado por secuencia directa (DSSS) y la modulación O-QPSK.

Esta codificación utiliza un código de pseudo-ruido (secuencia PN) 32-ario, con aproximadamente la misma cantidad de bits bajos que altos, para cada 4 bits de la señal a transmitir. La modulación aplicada después transmite 2 bits por símbolo.

5.2.2 Capa MAC (802.15.4)

La capa de enlace gestiona las operaciones de descubrimiento de vecinos y de retransmisión de perdidas.

Sumado a esto, añade el protocolo acceso múltiple por detección de portadora y prevención de colisiones (CSMA-CA) para controlar el acceso radioeléctrico de los nodos.

5.2.3 Capa de red (Zigbee)

5.2.3.1 *Personal Area Network ID*

Las diferentes redes se identifican mediante el identificador único de red de área personal (PAN ID). Los dispositivos Zigbee tienen configurada un PAN ID para unirse a su correspondiente red. También pueden descubrir redes cercanas y unirse a ellas seleccionando el identificador de la red destino.

Este identificador está compuesto por dos direcciones, una de 16 bits y otra de 64 bits. La primera se vio insuficiente debido a la limitación de direccionamiento (65,535 direcciones). Había la posibilidad de que dos redes en el mismo rango poseyeran el mismo identificador. Por ese motivo, Zigbee Alliance añadió la dirección de 64 bits, que es usada en caso de conflicto de las direcciones de 16 bits.

Cuando un coordinador inicie una red, utilizará una dirección PAN de 16 bits aleatoria y una dirección de 64 bits preconfigurada o aleatoria. Los dispositivos finales y routers normalmente están configurados para unirse a redes con cualquier dirección de 16 bit siempre que coincida la dirección de 64 bits.

5.2.3.2 *Roles*

Una red Zigbee puede estar compuesta por 3 tipos de nodos: Coordinador, router y dispositivo final.

- El **coordinador** es único en la red y se encarga de establecer la red, asignar direcciones y mostrar a los demás elementos el camino a seguir para su comunicación.
- Los **routers** se encargan de multiplexar y redirigir los flujos de mensajes de los dispositivos finales con el objetivo de ampliar la red.
- Los **dispositivos finales** solo tienen conexión con su nodo padre, ya sea coordinador o router. No son capaces ni de enrutar mensajes ni de añadir nuevos dispositivos a la red.

El estándar IEEE 802.15.4 clasifica a los dispositivos como FFD (Full-Function Device) y RFD (Reduced-Function Device). Siendo el primero de ellos el único que tiene capacidad para ser configurado como coordinador de una red.

5.2.3.3 Topología de red

Existen tres tipos de topología: estrella, árbol y malla. En el caso de la primera, el coordinador es el nodo central. En las restantes, se utiliza la conexión entre routers para extender la red.

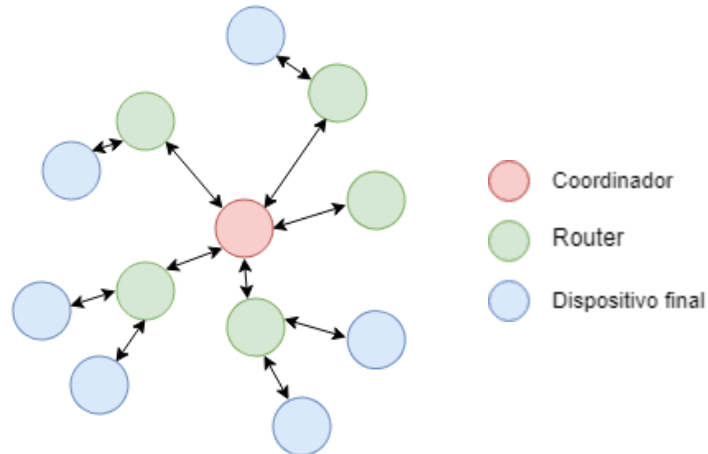


Figura 5-3 Tipología estrella

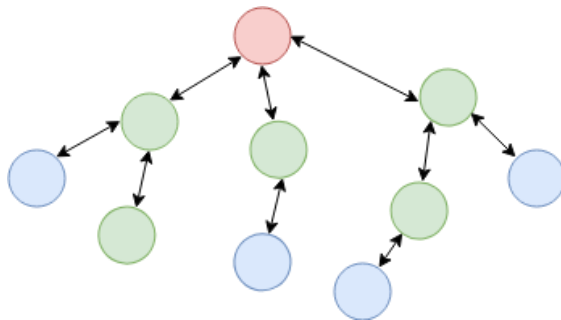


Figura 5-4 Tipología árbol

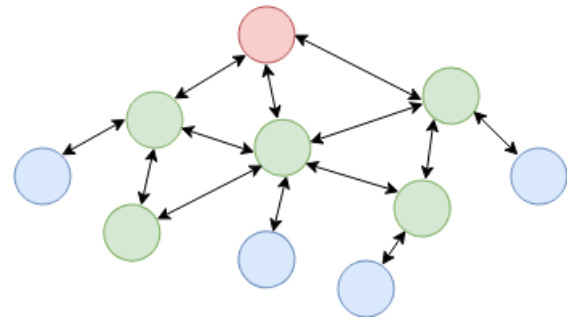


Figura 5-5 Tipología malla

La diferencia entre la tipología estrella y las demás es que todos los router tienen conexión directa con el coordinador de la red.

Como se puede apreciar en los esquemas, la topología de malla proporciona redundancia útil para evitar la pérdida de conexión por caída de algún nodo.

5.2.4 Capa de aplicación (Zigbee)

La capa de aplicación está encargada de la interconexión de dispositivos y definición de sus funciones. Está formada por los siguientes elementos:

- **“Application Support Sublayer” (APS)**: Es un nexo entre la capa de red y de aplicación. Estandariza el tipo de mensaje para permitir la interoperabilidad entre dispositivos de diferentes fabricantes. Reconoce diferentes tipos de perfiles de dispositivos según su función en la red.
- **“Application Framework”**: Es el entorno de trabajo donde se almacenan las aplicaciones en los dispositivos ZigBee.
- **“ZigBee Device Object” (ZDO)**: Aporta funciones avanzadas de gestión de red y de descubrimiento de dispositivos.

5.3 UART

Toda la comunicación no inalámbrica que se ha utilizado en este proyecto ha sido mediante mecanismos transmisor-receptor asíncrono universal. El UART es el dispositivo que se encarga de controlar los puertos y dispositivos serie de la placa. Tanto la comunicación entre los microcontroladores con los módulos transceptores como la comunicación USB entre el ordenador y el nodo coordinador de la red han intercambiado bits de esta forma.

En este tipo de comunicación, se toman los datos byte a byte y el UART de la placa microcontroladora o del ordenador se encargan de transmitir los bits de forma secuencial.

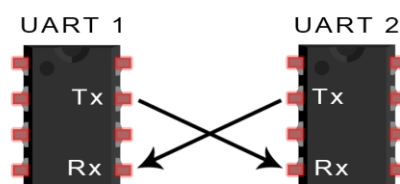


Figura 5-6 Interconexión entre UART para formar una conexión serie [13]

Para que exista comunicación de datos entre dos UART estos tienen que tener varios parámetros configurados de la misma forma. Estos son:

- El **“Baud Rate”** o velocidad de transmisión serial.
- El número de bits de datos.
- El número de bits de parada de transmisión.
- Si utiliza bit de paridad o no.

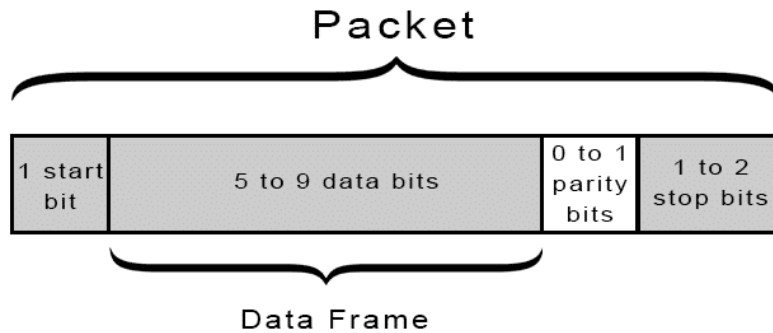


Figura 5-7 Estructura de secuencia de bits intercambiada entre UARTs [13]

La línea de transmisión serie se encuentra en valor lógico alto cuando no se está transmitiendo. En el momento de comenzar la transmisión se pone un bit a nivel bajo y al terminar, 1 o 2 bits en alto. El valor de los demás bits será dependiente de los datos a transmitir.

La velocidad de transmisión serial, a diferencia de la velocidad de bit, tendrá en cuenta estos bits extra.

6 Desarrollo hardware

Uno de los factores más importantes para la elección de microcontrolador es la comodidad para el usuario. Si este es demasiado grande podría molestar en la ejecución del movimiento por lo que probablemente causaría la adquisición de datos que difieran de los reales.

6.1 Microcontrolador Arduino Lilypad USB plus

El microcontrolador Arduino Lilypad es una opción más que aceptable. Está ideado para su inclusión en textiles al poder ser cosido gracias a la apertura de sus pines. Estos, llamados pines pétalos, se distribuyen homogéneamente por el contorno de la placa y son excelentes para prototipados iniciales mediante cables con conectores tipo cocodrilo ya que son de gran superficie y cuentan con una separación entre ella adecuada.

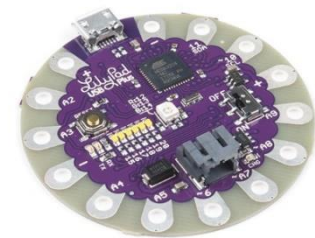


Figura 6-1 Placa Lilypad USB plus
[14]

El núcleo de la placa es el microchip ATmega32U4-M4 situado en el centro. Unas de las características de este microcontrolador son:

- Arquitectura: AVR
- Frecuencia máxima de reloj: 16 MHz
- Memoria programable: 32 kB
- Memoria RAM: 2.5kB
- Resolución conversor analógico-digital: 10 bits

La placa cuenta con 6 entradas analógicas, punto muy de interés para este proyecto ya que permite incorporar varios sensores analógicos que son los solicitados en este tipo de mediciones. Por ejemplo, permitiría conectar dos módulos acelerómetro.

6.1.1 Lilypad USB Plus vs. Arduino Nano

En un principio se consideró la posible elección de esta otra placa por su reducido tamaño y por poseer un microchip de características muy similares a las anteriormente citadas.

Gracias a la forma redondeada de Lilypad, se puede determinar que no produciría lesiones si se colocara en la ropa deportiva del atleta. Existe además la posibilidad de utilizar cordón conductor con lo que el propio circuito iría incorporado dentro de la tela y serviría a su vez de sujeción de los componentes.

Sin embargo, Arduino Nano tiene forma rectangular, por lo que posee 4 vértices que podrían causar heridas en la piel. Además, sería necesario retirar los pines puntiagudos, con los que viene de fábrica, con un desoldador. Estos pines quedarían muy próximos entre ellos y podrían producir cortocircuitos si no se soldara de forma muy cuidadosa.

Otro punto a favor de esta elección es que es que la gama Lilypad posee una placa Shield XBee de parecida morfología que sería útil para dotar al controlador con conexión inalámbrica ya sea mediante Wi-Fi, Zigbee u otros protocolos propios de fabricantes como puede ser DigiMesh.

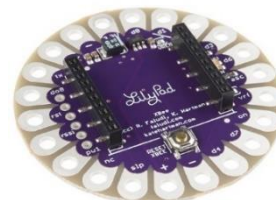


Figura 6-2 Placa Shield Xbee [15]

6.2 Microcontrolador Arduino Uno.

El proyecto cuenta con un nodo que no debe de ser colocado en el usuario y es el que se encargara de recibir los datos de los demás nodos. Así pues, no tiene mayor transcendencia su tamaño.

Se ha elegido Arduino Uno por poseer una gran cantidad de Shields comerciales para módulos de conexión inalámbrica compatibles con él.

No tiene importancia que modelo se use siempre y cuando tenga conmutador de puerto serial para poder usarlo también para configurar los módulos inalámbricos mediante la conexión USB de Arduino Uno.

En este proyecto se ha usado un modelo de Shield ya descatalogado del fabricante Libelium.



Figura 6-3 Arduino UNO, Shield y modulo Xbee [16]

6.3 Conexión inalámbrica. Modulo Xbee

Para dotar al sistema de comunicación inalámbrica se han utilizado módulos Xbee. En concreto se han utilizado módulos de la serie 2B Pro ya que se encuentran disponibles en el laboratorio de Antenas y Microondas. La diferencia de un dispositivo Pro y el modelo estándar reside en el rango de alcance, siendo mayor en las versiones Pro. Por otro lado, al pertenecer el dispositivo a la serie 2 nos indica que existe la posibilidad de crear redes en forma de malla con él. Es importante tener en cuenta que hardware de diferente serie no puede ser interconectado.

Aunque los nombres Xbee y Zigbee sean parecidos se tiene que tener en cuenta que no todos los módulos Xbee utilizan este protocolo. Como se ha comentado, los elegidos si que utilizan ese protocolo.

Tiene una velocidad de emisión al medio radioeléctrica de 250 kp/s, proveniente del protocolo 802.14.5 y una tasa de datos de 35kb/s.

Puede operar en el rango de canales 11-25 de la banda de 2,4 GHz.

La potencia de trasmisión de la antena es regulable para poder controlar consumo de potencia. El rango es de 63mW - 10mW (+18Bm - +10dBm). La sensibilidad de recepción es de -102 dBm.

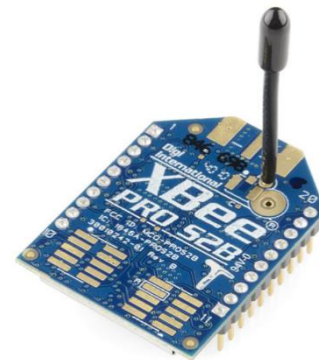


Figura 6-4 Módulo Xbee PRO S2B [17]

Esto hace que tenga una cobertura de 90-60 metros en interiores o escenarios urbanos y 3.2-1.5 kilómetros en exteriores con comunicación directa.

Dentro de cada modelo existen dos tipos de hardware: Los SMT y TH. Los primeros son para montar directamente sobre PCB soldando sus 37 patillas y los segundos tienen 20 pines que permiten ser acoplado en microcontroladores mediante placas adaptadoras comerciales.

Se alimentan con 3.3 V ya que un voltaje superior dañaría el sistema. Como el voltaje de las patillas de alimentación del microcontrolador elegido es igual, no es necesario utilizar ningún divisor de tensión.

El modelo elegido cuenta con una antena omnidireccional de ganancia 1.5 dBi formada por un monopolo articulable. Existen varios tipos de antena formadas por ejemplo por dipolos (2.1 dBi) o embebidas en el propio PCB (-0.5 dBi). También los hay con conector RP-SMA o U.FL para acoplarle una antena de mayor alcance y tamaño. Para el desarrollo de este proyecto la antena wire se ha considerado suficiente



Figura 6-5 Ejemplos de conexión de antenas en los puertos U.FL y RP-SMA [18][19]

Dispositivo Xbee	Nombre	Rango	Consumo de potencia	Frecuencia	Protocolo	Potencia de transmisión	Velocidad de transmisión	Tipo de antena
	Xbee Chip Antenna - Series 1	90 m	50mA @ 3.3v	2.4GHz	802.15.4	1mW	250kbps	Chip
	XBee U.FL Connection - Series 1	90 m	50mA @ 3.3v	2.4GHz	802.15.4	1mW	250kbps	No incluida
	XBee Wire Antenna - Series 1	90 m	50mA @ 3.3v	2.4GHz	802.15.4	1mW	250kbps	Wire
	Xbee Trace Antenna - Series 1	90 m	50mA @ 3.3v	2.4GHz	802.15.4	1mW	250kbps	PCB
	XBee 2mW PCB Antenna - Series 2	120 m	40mA @ 3.3v	2.4GHz	ZigBee Mesh	2mW	250kbps	PCB
	XBee 2mW RPSMA - Series 2	120 m	40mA @ 3.3v	2.4GHz	ZigBee Mesh	2mW	250kbps	No incluida
	XBee 2mW U.FL Connection - Series 2	120 m	40mA @ 3.3v	2.4GHz	ZigBee Mesh	2mW	250kbps	No incluida
	XBee Pro 50mW U.FL Connection - Series 2	1.6 Km	295mA @ 3.3v	2.4GHz	ZigBee Mesh	50mW	250kbps	No incluida
	XBee Pro 63mW RPSMA - Series 2B	1.6 Km	295mA @ 3.3v	2.4GHz	ZigBee Mesh	63mW	250kbps	No incluida
	XBee Pro 63mW PCB Antenna - Series 2B	1.6 Km	295mA @ 3.3v	2.4GHz	ZigBee Mesh	63mW	250kbps	PCB
	XBee Pro 63mW Wire Antenna - Series 2B	1.6 Km	295mA @ 3.3v	2.4GHz	ZigBee Mesh	63mW	250kbps	Wire

Figura 6-6 Algunos de los modelos comerciales XBee TH de las series 1, 2 y 2B [20]

Estos módulos tienen dos modos de trabajo:

- Modo transparente o AT

Trabajando en este modo, solo se emite la información tal y como llega por su puerto serie a menos que sean comandos AT. No hay forma de saber cuál es la dirección de origen de la información que se ha recibido. Por esta razón está orientado a conexiones punto a punto.

- Modo API

En este modo, la información viaja en formato de trama con cabecera, datos y CRC. Por lo que no se tiene el mismo problema que el anterior modo, ya que de la cabecera se obtiene la dirección del módulo emisor.

Byte de inicio	Longitud		Tipo de trama	Datos	Checksum
1	2	3	4	5 : n	n+1
0x7E	Byte más significativo	Byte menos significativo	Tipo de trama API	Datos (n-4 bytes)	Un byte

Figura 6-7 Estructura de tramas API

Es útil para redes multipunto en la que hay varios routers o dispositivos finales. Además, permite configurar los módulos vecinos de forma remota.

6.4 Sensor. Acelerómetro

En el mercado existe una inmensa variedad de acelerómetros. Uno de los factores más importantes a la hora de elegir es el rango de aceleración medible el cual se expresa en múltiplos de la gravedad terrestre, cuyo valor es $9,8 \text{ m/s}^2$. Los hay de incluso $\pm 200g$.

En este proyecto en concreto, al tratarse de movimientos corporales, no se trata de un factor limitante, por lo que se podrá elegir un acelerómetro teniendo en cuenta la comodidad de uso.

Siguiendo la línea de las opciones elegidas hasta ahora, el fabricante Sparkfun incorpora el acelerómetro de 3 ejes ADXL335 de Analog Devices en un PCB de dimensiones

reducidas y compatible con la opción de ser cosido a tejido. Según el fabricante el rango típico es de ± 3.6 g y su rango mínimo garantizado es de ± 3 g.



Figura 6-8 Acelerómetro ADXL335 [21]

Es de baja potencia y está testado con una tensión de 3 V por lo que es perfecto para un sistema de bajo consumo como el que se presenta en este proyecto. Como la alimentación partirá del microcontrolador, será sometido a 3.3V. Por lo que se producirán unos pequeños cambios de medición controlados según informa el datasheet del fabricante.

En su hoja de características se detallan posibles aplicaciones a las que se puede destinar su uso. Entre ellas se encuentran: Dispositivos móviles, sistemas de juego, estabilizadores de imágenes y dispositivos destinados a propósitos deportivos o del ámbito de la salud. Esto corrobora que se trata de un dispositivo apto para este proyecto.

6.4.1 Calibración acelerómetro

Se ha medido con un multímetro el voltaje que devuelve el acelerómetro por cada uno de sus pines en condiciones de 0g. Para ello se coloca en una superficie plana y se procede a la medida de los pines correspondientes a los ejes X e Y. Para el eje Z se coloca el acelerómetro por ejemplo en una pared vertical para que se encuentre en una situación de 90° respecto al vector de la gravedad terrestre y se procede a la medición.

El resultado ha sido de 1.62 V, 1.63 V y 1.70 V en los ejes X, Y y Z respectivamente. Si los comparamos con los valores que proporciona la tabla de características de la figura se puede apreciar que son mayores que los nominales, esto es porque el acelerómetro se está alimentando con 3.3 V en vez de con 3 V.

El siguiente paso es hacer la conversión de dichos voltajes a niveles o “steps” de ADC. Esto se consigue mediante una regla de tres que relaciona el nivel de nivel máximo 1023 con el voltaje al que se alimenta el acelerómetro.

$$\frac{V_s}{V_{0g}} = \frac{1023}{ADC_{0g}}$$

En este caso los valores de nivel ADC son 509, 512 y 527. A los valores medidos por el Lilypad se le deberá sustraer dicho valor de condición 0g.

Un parámetro necesario que nos aporta el datasheet del elemento es la sensibilidad, medida en mv/g. Mirando la tabla de características podemos encontrar que el valor de sensibilidad es 300mv/g o 0.3V/g.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT					
Measurement Range	Each axis	±3	±3.6		g
Nonlinearity	% of full scale		±0.3		%
Package Alignment Error			±1		Degrees
Interaxis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity ¹			±1		%
SENSITIVITY (RATIOMETRIC)²					
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	V _S = 3 V	270	300	330	mV/g
Sensitivity Change Due to Temperature ³	V _S = 3 V		±0.01		%/°C
ZERO g BIAS LEVEL (RATIOMETRIC)					
0 g Voltage at X _{OUT} , Y _{OUT}	V _S = 3 V	1.35	1.5	1.65	V
0 g Voltage at Z _{OUT}	V _S = 3 V	1.2	1.5	1.8	V
0 g Offset vs. Temperature			±1		mg/°C
NOISE PERFORMANCE					
Noise Density X _{OUT} , Y _{OUT}			150		µg/√Hz rms
Noise Density Z _{OUT}			300		µg/√Hz rms
FREQUENCY RESPONSE⁴					
Bandwidth X _{OUT} , Y _{OUT} ⁵	No external filter		1600		Hz
Bandwidth Z _{OUT} ⁵	No external filter		550		Hz
R _{FILT} Tolerance			32 ± 15%		kΩ
Sensor Resonant Frequency			5.5		kHz
SELF-TEST⁶					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		µA
Output Change at X _{OUT}	Self-Test 0 to Self-Test 1	-150	-325	-600	mV
Output Change at Y _{OUT}	Self-Test 0 to Self-Test 1	+150	+325	+600	mV
Output Change at Z _{OUT}	Self-Test 0 to Self-Test 1	+150	+550	+1000	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	V _S = 3 V		350		µA
Turn-On Time ⁷	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	°C

Figura 6-9 Hoja de características acelerómetro ADXL335 [22]

Así pues, teniendo la relación de V/g y la relación steps/V, incluimos la siguiente fórmula en el código Arduino para obtener los valores de aceleración medidos en gravedades terrestres por cada uno de los ejes.

$$Aceleracion (g) = \frac{(Valor\ ADC - ADC\ 0g) * Vs}{1023\ steps * \frac{0.3v}{g}}$$

Estas operaciones se verán reflejadas en el código fuente Router.ino

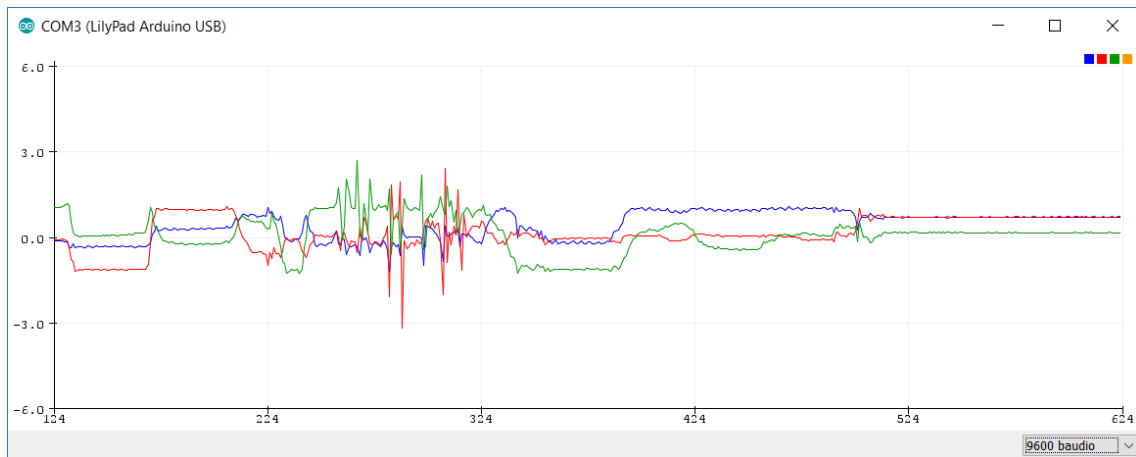


Figura 6-10 Gráfica de aceleraciones de los 3 ejes cartesianos durante movimientos aleatorios de prueba

7 Desarrollo software

7.1 Arduino IDE

Para la programación de los microcontroladores se ha usado el software gratuito Arduino IDE. El lenguaje de programación usado es C++, aunque Arduino incorpora unas librerías, también llamadas “core”, que facilitan la programación de los pines de entrada y salida y los puertos de comunicación, además de librerías para operaciones específicas. El propio IDE carga automáticamente estas librerías y nos permite añadir librerías desarrolladas por la comunidad de usuarios.

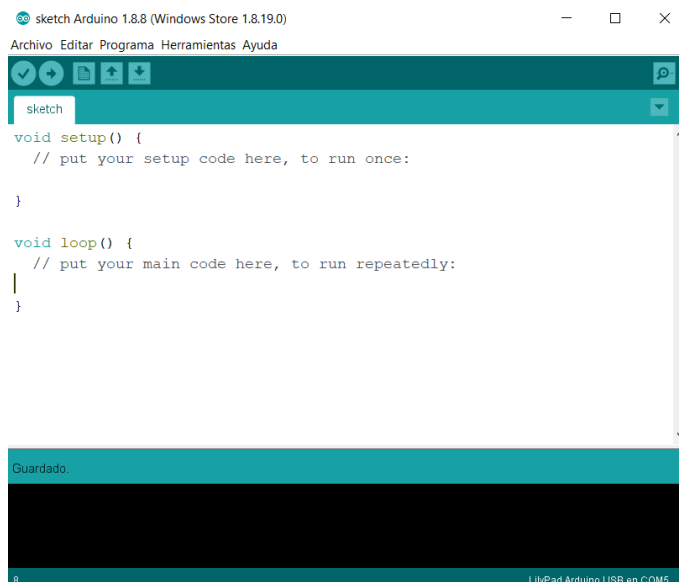


Figura 7-1 Arduino IDE con un proyecto en blanco

El código está dividido en la función `setup()` y `loop()` a diferencia del lenguaje C++ estándar que utiliza la función `main()`.

A la hora de volcar el código se debe conectar la placa mediante el cable USB y seleccionar el puerto desde el menú Herramientas.

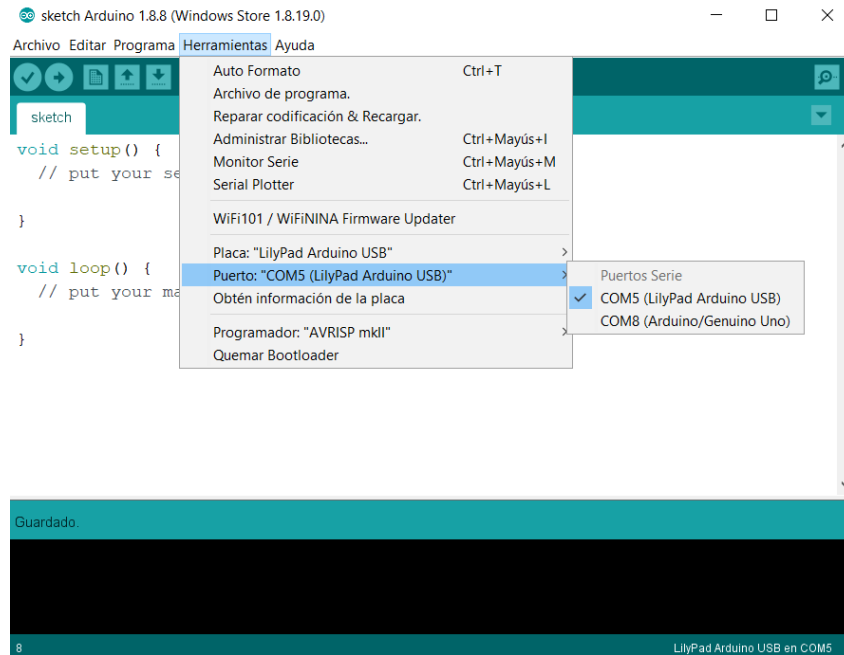


Figura 7-2 Selección de puerto serie.

Como nuestro proyecto está compuesto por dos tipos de microcontroladores. Deberemos seleccionar el tipo de placa antes de proceder al volcado de código.

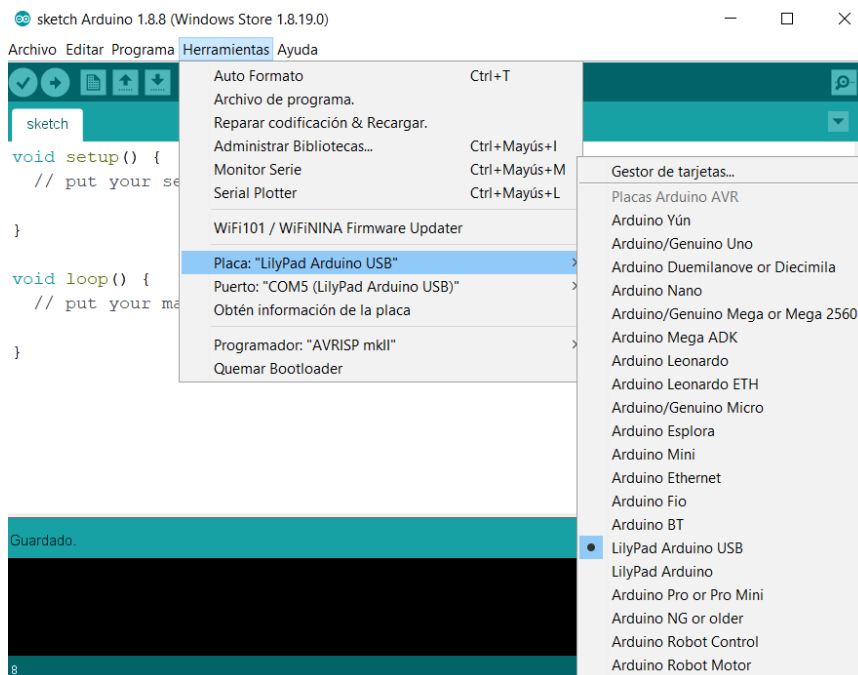


Figura 7-3 Selección de tarjeta microcontroladora.

7.2 XCTU

Para la configuración de los módulos inalámbricos Zigbee se ha usado el software XCTU, desarrollado por el fabricante DIGI .

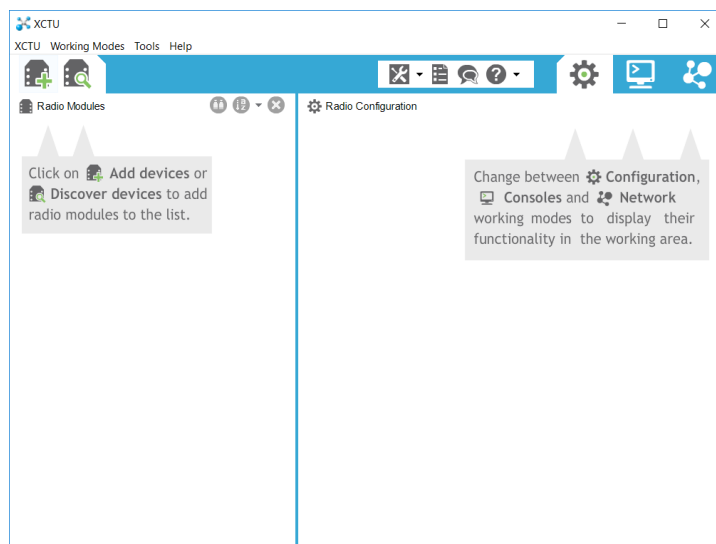


Figura 7-4 Entorno XCTU

En primer lugar, se debe añadir o escanear el módulo XBee. La diferencia de estos dos procesos es que la segunda opción nos permite seleccionar varias opciones de configuración para poder descubrir el módulo. En cambio, la primera es más rápida y útil en el caso de saber a priori la configuración del módulo. Para estos procesos es necesaria la conexión a un puerto USB mediante el circuito llamado "XBee explorer".

Otra opción, utilizada en este proyecto, es la utilización de la placa Arduino uno y el Shield XBee en lugar del XBee explorer. Para ello es necesario quitar el microprocesador y cambiar el conmutador correspondiente al Serial a la opción USB, ya que, de otro modo XCTU no encontrará el módulo.

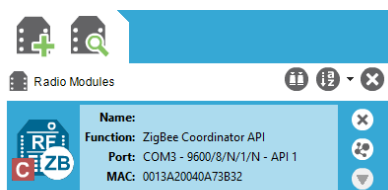


Figura 7-5 Modulo Coordinador añadido

Una vez se añade el módulo, XCTU ofrece las siguientes opciones:

- Configuración de parámetros del módulo.

Al acceder a esta opción aparecen listados una serie de parámetros que pueden variar según el modo y rol en el que se encuentre el dispositivo Xbee.

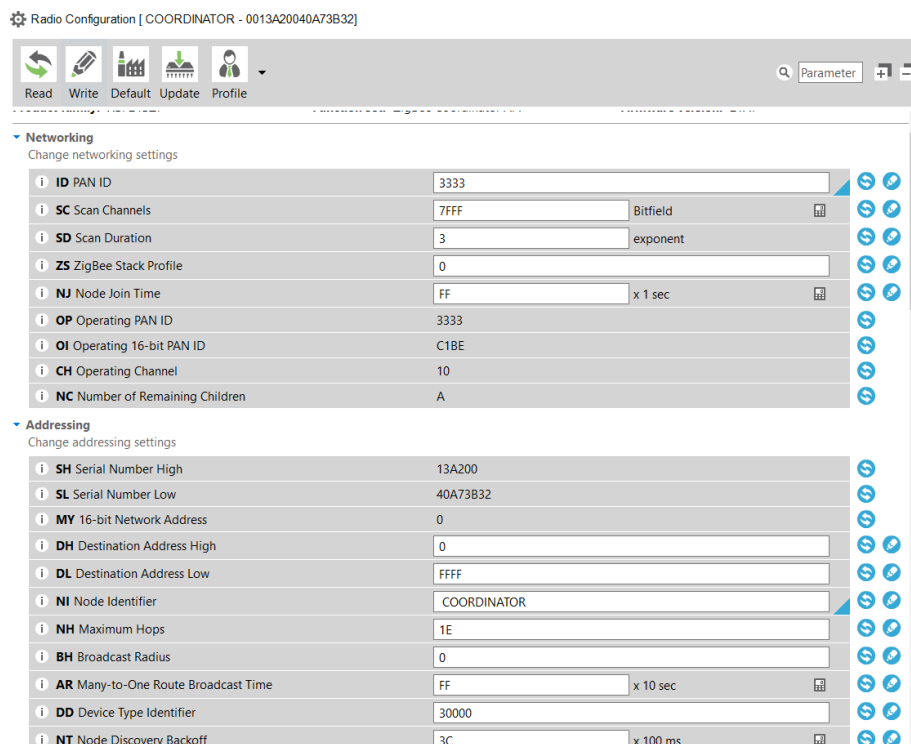


Figura 7-6 Opciones de configuración del módulo Xbee en modo coordinador

Al modificar un parámetro se debe transmitir mediante la opción “write”. Se puede hacer campo a campo o en general. Cada parámetro modificado aparece en verde hasta que el cambio es escrito al módulo. De forma similar, aparecen marcados en color azul los campos que han sido modificados respecto a la configuración por defecto.

Si se desea cambiar el modo de comunicación o el rol del módulo, se debe usar la opción “Update firmware”. Esta abrirá un desplegable en el que se debe seleccionar el número de serie de la familia de producto, la función compuesta por modo de comunicación y por el rol que se le desea asignar y la versión de firmware.

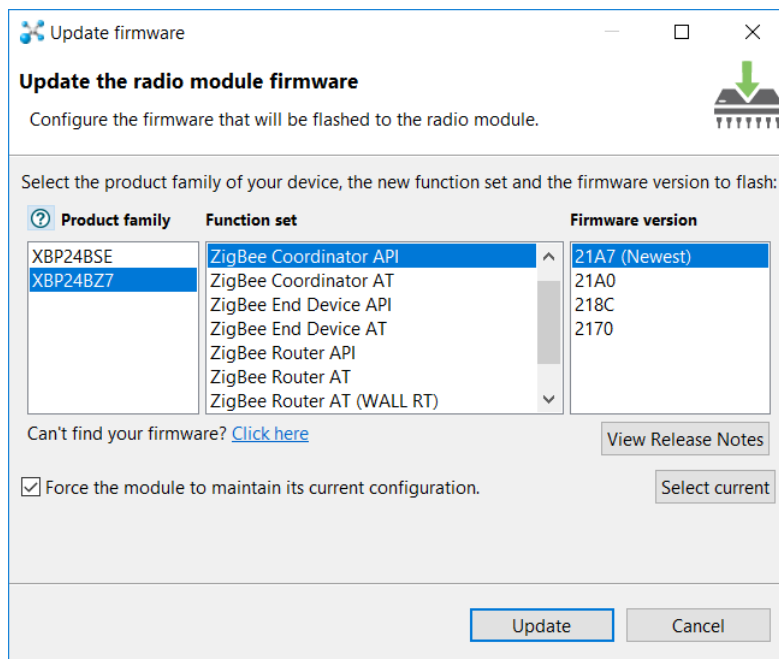


Figura 7-7 Selección de función a asignar y versión del firmware.

- Consola de paquetes.

La apariencia de esta ventana depende del modo de comunicación en el que se encuentre. En el modo AT solo se puede escribir caracteres y leer caracteres tanto en ASCII como en hexadecimal. También se puede entrar en modo comando desde esta consola.

En modo API se puede observar las tramas recibidas y enviadas, pudiendo filtrar según el tipo de paquete. Accediendo a una trama se puede observar la estructura y revisar su contenido.

También existe la opción de mandar paquetes con diferentes comandos de forma única o periódica

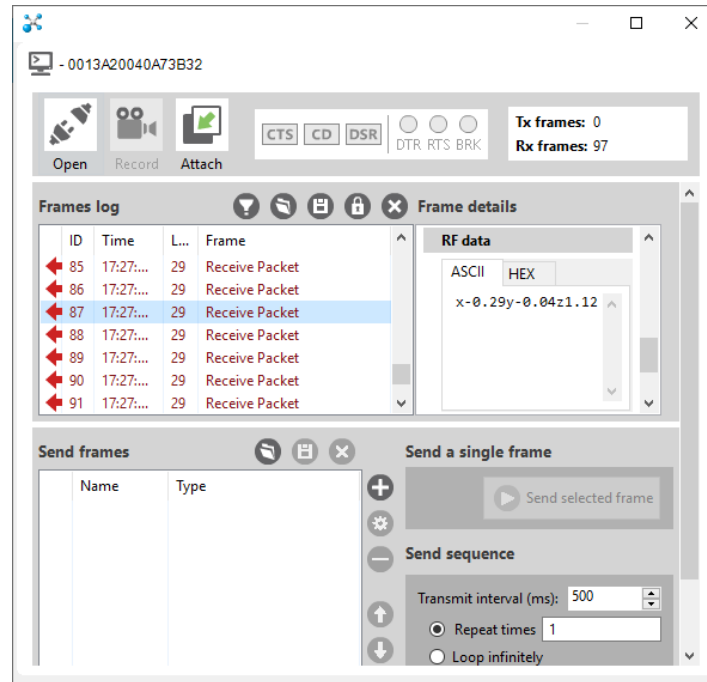


Figura 7-8 Consola de paquetes.

La opción *copy packet information* permite copiar la trama en el portapapeles. Se incluye a continuación un ejemplo de esta información.

Receive Packet (API 1)

7E 00 1D 90 00 13 A2 00 40 9F 39 08 CE 66 01 78 2D 30 2E 32 39 79 2D 30 2E 30 34 7A 31 2E 31 32 53

Start delimiter: 7E

Length: 00 1D (29)

Frame type: 90 (Receive Packet)

64-bit source address: 00 13 A2 00 40 9F 39 08

16-bit source address: CE 66

Receive options: 01

RF data: 78 2D 30 2E 32 39 79 2D 30 2E 30 34 7A 31 2E 31 32

Checksum: 53

Esto es de gran utilidad para empezar a familiarizarse con el formato de la trama API las primeras ocasiones que se utiliza este tipo de comunicación.

- Configuración remota.

Mediante esta opción se puede localizar los nodos que se encuentran en la misma red y permite modificar su configuración de forma remota sin la necesidad de estar colocando todos los nodos en el explorer. La única condición es que se encuentren conectados a la misma red y que el módulo padre se encuentre en modo API.

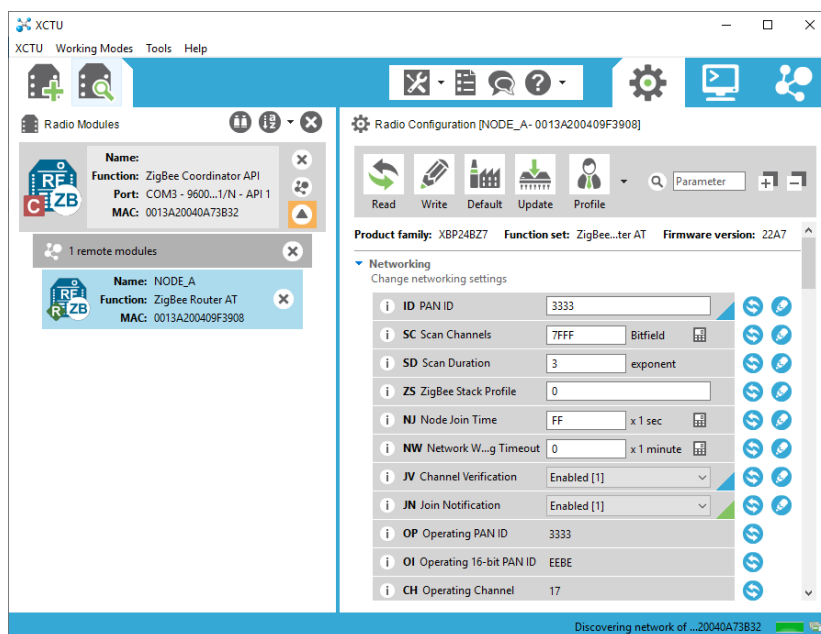


Figura 7-9 Opciones de configuración de un módulo remoto.

Una vez se cargue la configuración, de la misma forma que en modo local, XCTU se encargará de enviar las tramas del tipo “Remote AT Command Request” con los parámetros de forma inalámbrica mediante el nodo padre.

El inconveniente de esta opción es que las operaciones de escritura son mucho más lentas. Por lo tanto, no es recomendable para cambios de firmware o modo de módulos.

- Esquema de red.

Esta sección representa la red mediante un esquema que interconecta los módulos que se encuentran operativos. De forma visual se puede ver las direcciones de los nodos y una representación por colores se puede identificar la calidad de la conexión.

7.3 Matlab

Se ha elegido Matlab por ser una herramienta matemática que facilita la labor de representar, procesar, guardar y cargar datos. Especialmente de interés en este proyecto es la facilidad de manipulación de datos en vectores.

Otro factor fundamental para su uso es que incluye una herramienta llamada “*App Designer*” que permite crear interfaces gráficas de usuario, comúnmente llamadas GUI (*Graphical user interface*), de manera muy sencilla e intuitiva mediante programación orientada a objeto.

Al añadir un elemento al espacio de diseño, este aparecerá automáticamente definido en el código fuente. Después se especifican los campos de ese objeto en la pestaña configuración. Por último, se puede añadir funciones asociadas a ese objeto que varían según el tipo de elemento al que pertenezcan.

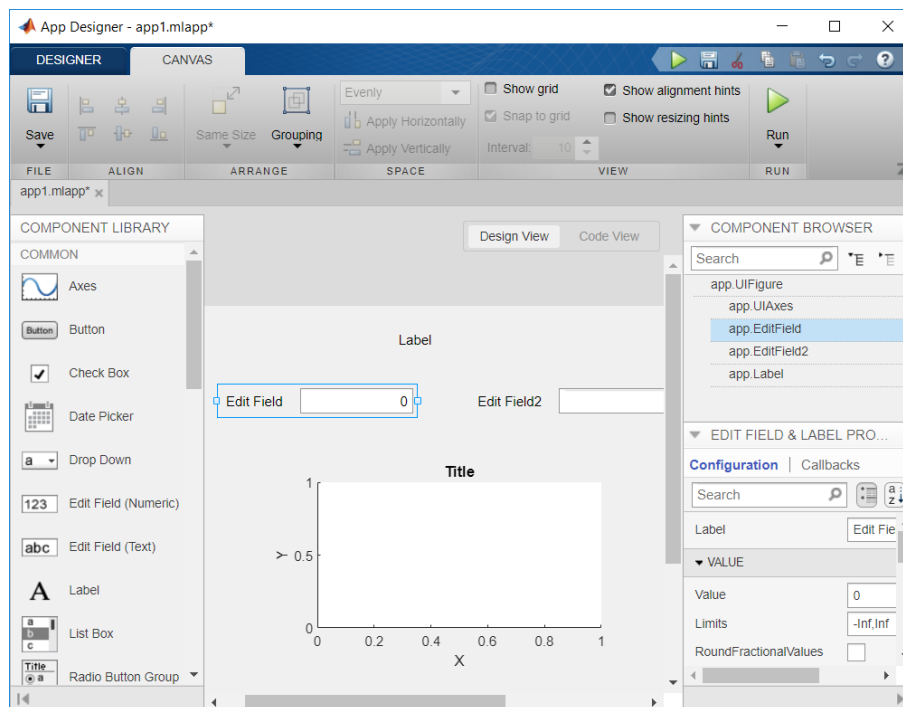


Figura 7-10 Entorno de “App Designer”

8 Primera conexión Zigbee punto a punto en modo transparente.

Uno de los primeros pasos que se realizaron, mientras se esperaba la llegada del hardware adquirido, fue la prueba de conexiones sencillas mediante dos módulos Xbee en modo transparente. Esta prueba sirvió de ayuda para familiarizarse con el entorno XCTU y el uso de los modulos y para comprobar el funcionamiento del modo de comunicación transparente.

Para ello se configuraron los módulos como coordinador AT y router AT. Y se deben fijar como direcciones de destino las direcciones de 64 bits del otro dispositivo.

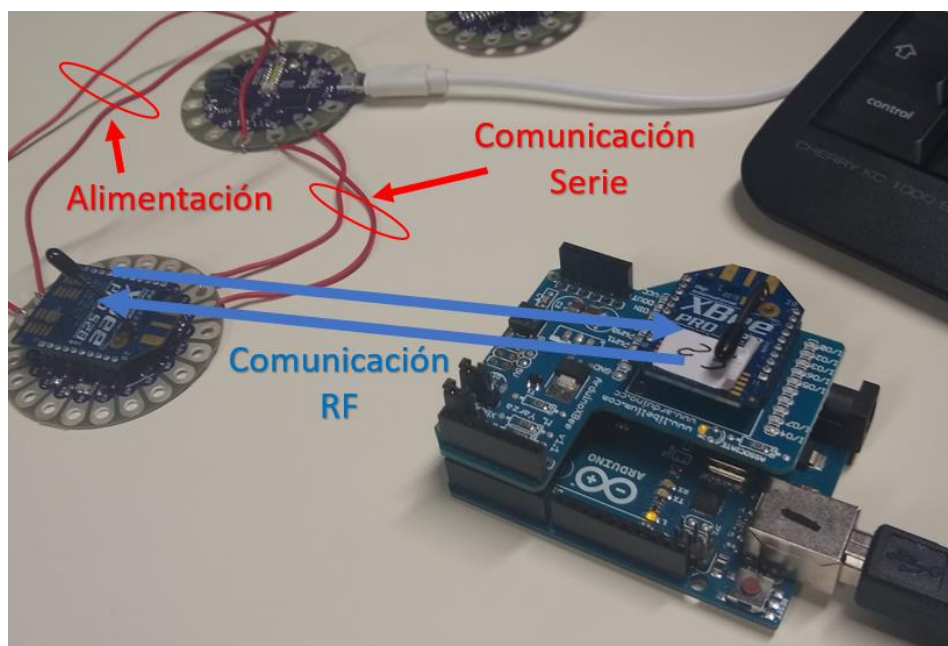


Figura 8-1 Hardware correspondiente a la primera conexión Zigbee en modo AT

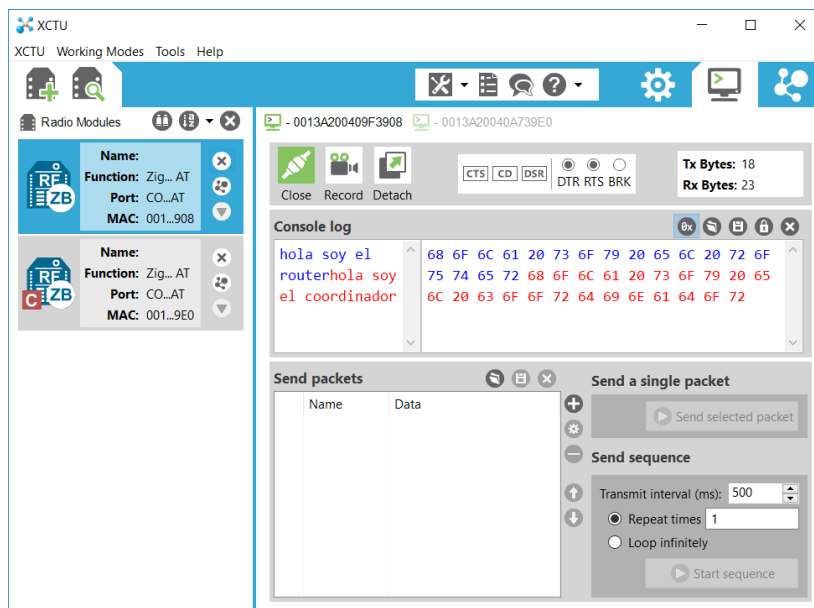


Figura 8-2 Ejemplo de intercambio de cadenas de texto entre dos módulos en modo transparente.

Pronto se comprendió que no era el modo adecuado para este proyecto, ya que solo se envían los datos en limpio, por lo que no existe forma de conocer el origen de la información. Este modo está más encaminado a conexiones punto a punto simples. Como se ha explicado antes, la intención del proyecto desde un principio ha sido la creación de una red que soporte varios nodos. Por lo que no se siguió trabajando en este modo.

9 Conexión punto a punto en modo API

En este escenario, la comunicación inalámbrica estará formada por un coordinador XBee en modo API montado sobre el microcontrolador Arduino Uno y un router Xbee en modo AT montado sobre la placa Shield Xbee.



Figura 9-1 Esquema de sistema punto a punto

El acelerómetro proporciona las medidas de aceleración en los 3 ejes cartesianos. Por tanto, el módulo Lilypad los obtiene mediante 3 lecturas analógicas de los puertos a los que se le conectan.

Una vez el microcontrolador ha obtenido y procesado los datos, los transfiere mediante UART a los pines tx y rx del módulo XBee.

Y por último son transmitidos de forma inalámbrica al coordinador de la red, que se encarga de leerlos y mandarlos por el puerto serie USB para su posterior procesamiento mediante la herramienta matemática Matlab.

Como se ha comentado, se han usado los microcontroladores Arduino Lilypad USB plus y Arduino Uno. Ninguno de ellos posee dos puertos serie físicos, por lo que se necesita la utilización de la librería SoftwareSerial para definir el segundo puerto serie. Se podría haber empleado la tarjeta Arduino Leonardo ya que posee dos puertos serie, pero no se ha considerado imprescindible.

```
#include <SoftwareSerial.h>      #include <SoftwareSerial.h>  
SoftwareSerial xbee(0,1);        SoftwareSerial xbee(SDA, SCL);
```

Figura 9-2. Definición de un nuevo puerto serie. Arduino Uno y Lilypad USB plus respectivamente.

Una vez definidos los puertos en cada una de las placas, con sus correspondientes pines de recepción y transmisión, se usan las funciones `read()` y `write()` para leer y escribir bytes.

En la trama se envían los tres ejes cartesianos y su correspondiente valor de aceleración en una cadena sin espacios. Esto se ha hecho así para poder comprobar de un rápido vistazo, tanto en ASCII como en hexadecimal, que los datos que se están transmitiendo son correctos.

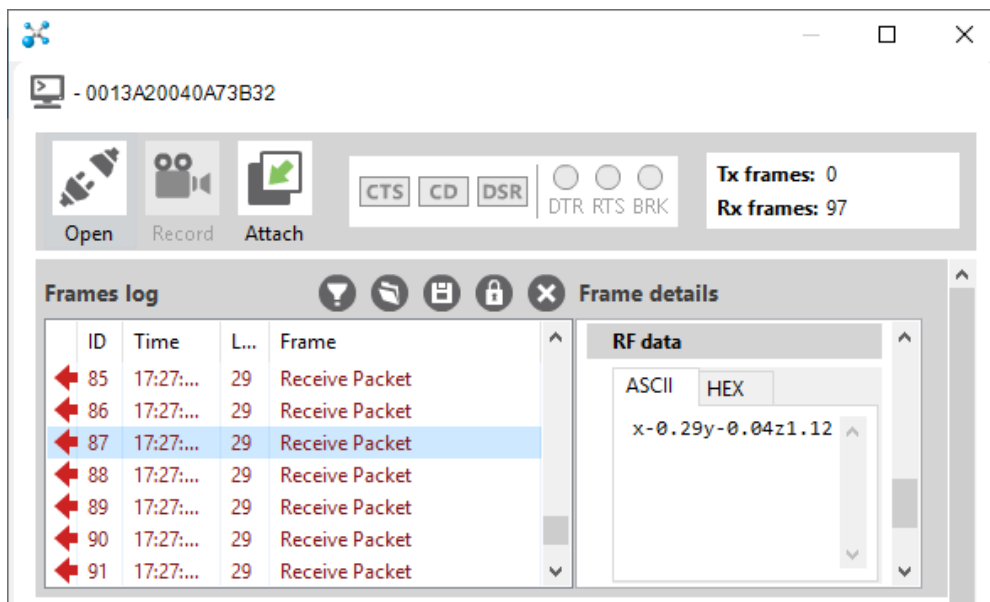
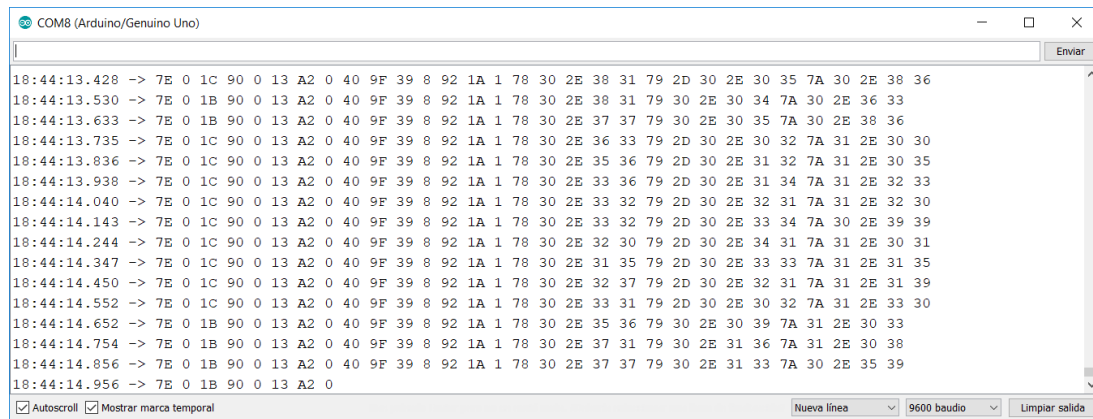


Figura 9-3 Paquetes recibidos en el coordinador.

Como se puede observar en XCTU, las tramas llegan de forma correcta y explorando su contenido se encuentra un ejemplo de datos del acelerómetro en el formato que ha sido enviado.

El siguiente paso es tener acceso a esos paquetes para poder enviarlos a Matlab. Para ello se usará el puerto serie del Arduino Uno. En el sketch subido a este microcontrolador, se procede a la lectura byte a byte del paquete y se escribirá en el puerto serie configurado.

Se puede utilizar la consola de Arduino IDE para comprobar que se está mandando la información por el puerto serie de forma correcta.



```
18:44:13.428 -> 7E 0 1C 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 38 31 79 2D 30 2E 30 35 7A 30 2E 38 36
18:44:13.530 -> 7E 0 1B 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 38 31 79 30 2E 30 34 7A 30 2E 36 33
18:44:13.633 -> 7E 0 1B 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 37 37 79 30 2E 30 35 7A 30 2E 38 36
18:44:13.735 -> 7E 0 1C 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 36 33 79 2D 30 2E 30 32 7A 31 2E 30 30
18:44:13.836 -> 7E 0 1C 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 35 36 79 2D 30 2E 31 32 7A 31 2E 30 35
18:44:13.938 -> 7E 0 1C 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 33 36 79 2D 30 2E 31 34 7A 31 2E 32 33
18:44:14.040 -> 7E 0 1C 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 33 32 79 2D 30 2E 32 31 7A 31 2E 32 30
18:44:14.143 -> 7E 0 1C 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 33 32 79 2D 30 2E 33 34 7A 30 2E 39 39
18:44:14.244 -> 7E 0 1C 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 32 30 79 2D 30 2E 34 31 7A 31 2E 30 31
18:44:14.347 -> 7E 0 1C 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 31 35 79 2D 30 2E 33 33 7A 31 2E 31 35
18:44:14.450 -> 7E 0 1C 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 32 37 79 2D 30 2E 32 31 7A 31 2E 31 39
18:44:14.552 -> 7E 0 1C 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 33 31 79 2D 30 2E 30 32 7A 31 2E 33 30
18:44:14.652 -> 7E 0 1B 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 35 36 79 30 2E 30 39 7A 31 2E 30 33
18:44:14.754 -> 7E 0 1B 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 37 31 79 30 2E 31 36 7A 31 2E 30 38
18:44:14.856 -> 7E 0 1B 90 0 13 A2 0 40 9F 39 8 92 1A 1 78 30 2E 37 37 79 30 2E 31 33 7A 30 2E 35 39
18:44:14.956 -> 7E 0 1B 90 0 13 A2 0
```

Figura 9-4 Conjunto de tramas API escritas byte a byte en el puerto serie USB.

Como se puede apreciar en la imagen, los bytes recibidos siguen el formato de trama API de Zigbee. Al terminar la trama, se realiza un salto de línea, carácter que será usado en Matlab a la hora de leer datos del puerto serie.

Los bytes de la trama se escriben prácticamente a la vez, pero la velocidad de llegada alta ha hecho que la última trama salga interrumpida en la captura de la imagen.

La longitud de esta trama depende del signo del sentido de las aceleraciones de los ejes, por ello pueden ser de tamaño variable, esto se tendrá en cuenta a la hora de procesar la trama.

Una vez en Matlab, se crea el objeto serial en el puerto correspondiente al Arduino Uno y se le asigna un delimitador de lectura correspondiente al salto de línea cuyo valor decimal es 10.

```
s = serial('COM8','Terminator',10);
```

Se ha creado un bucle el cual se repetirá las mismas veces que tramas deseemos registrar. En cada interacción, se lee el puerto hasta encontrar el salto de línea con la función fscanf().

Una vez ya se tiene la trama, se procesa y se obtiene los valores de aceleración en formato de coma flotante de doble precisión que se pueden ir graficando conforme se leen las tramas.

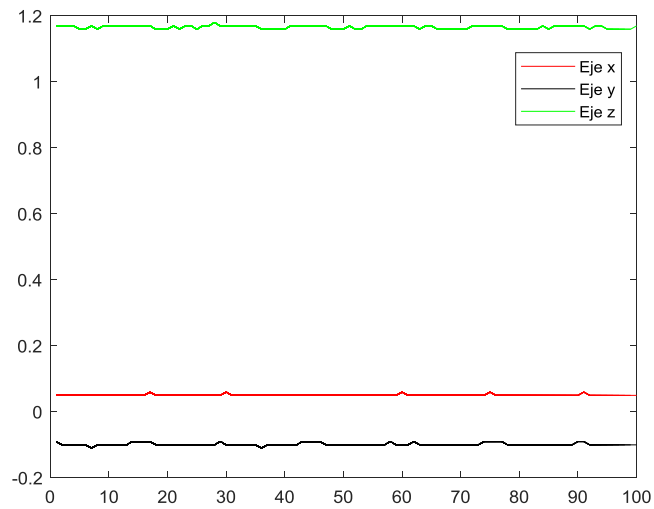


Figura 9-5 Representación de valores de aceleración cuando el acelerómetro está inmóvil.

En la representación gráfica se puede observar que los Ejes X e Y tienen valores cercanos a 0 g mientras que el eje Z tiene valores superiores a 1 g. Esto es debido a que el acelerómetro se encontraba en reposo en una posición en la que el eje Z estaba alineado con la gravedad.

Tras realizarse comprobaciones con movimientos simples y comprobar la veracidad de las medidas ya se puede proceder a estudiar como colocar el dispositivo en un usuario para realizar medidas simples.

10 Prototipo funcional inalámbrico

Hasta ahora todas las pruebas se han realizado con el circuito sobre una mesa, por lo que se necesita dotarle al sistema de movilidad para seguir con pruebas más próximas a su función final.

Con afán de reducir las dimensiones del sistema, se ha agrupado la placa Lilypad y el Shield XBee en dos planos paralelos mediante la colocación de unos separadores con tornillos plásticos. Para ello se han usado algunos de los agujeros que ambas placas tienen y que no se van a usar para el desarrollo de este proyecto.

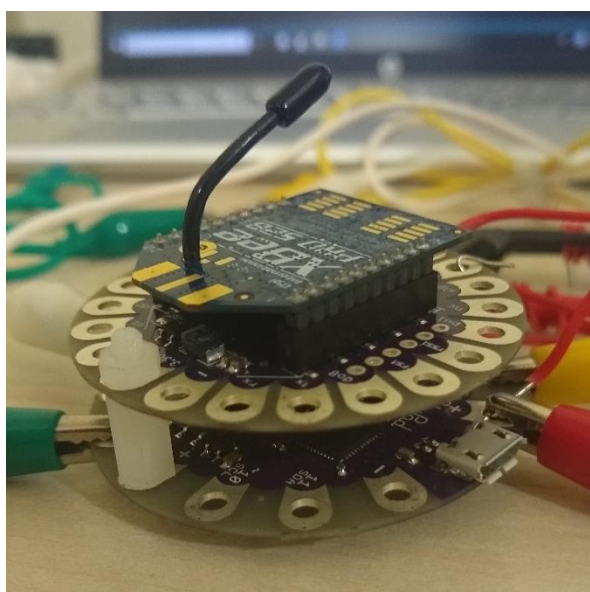


Figura 10-1 Montaje mediante tornillos plásticos. Módulo Xbee y Lilypad USB plus

10.1 Sistema de alimentación

Para darle independencia total, se requiere una fuente de alimentación, ya que, no se podrá usar la conexión USB que se venía utilizando hasta el momento. Este puede ser un tema crítico, ya que puede aumentar drásticamente el tamaño del sistema.

Se han considerado dos opciones. La primera, una pila de botón de 3.3 voltios que sería soldada a los pines positivo y negativo. Pero en función de su descarga el voltaje proporcionado sería menor y podría no ser lo suficiente. Además, se perdería la funcionalidad del interruptor de encendido y apagado por lo que cada vez que se dejara de usar el dispositivo sería necesario retirar la pila.

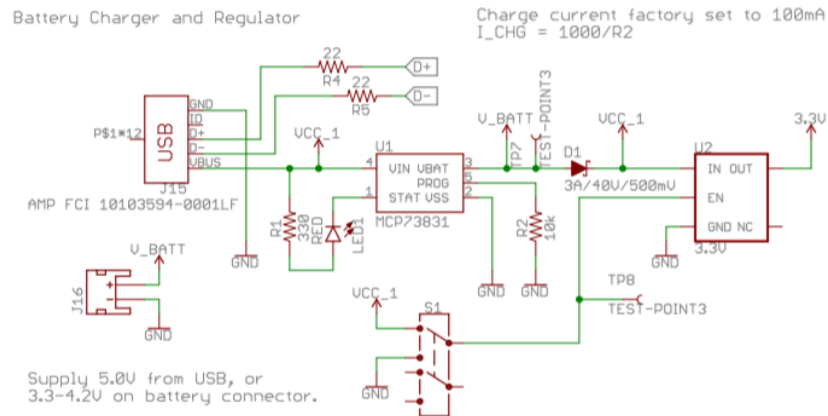


Figura 10-2 Esquemático de regulador de carga y puerto de batería de LilyPad USB plus [23]

La otra opción ha sido utilizar una batería tipo Lipo de 3,7 V con conector tipo JST de tamaño muy reducido. El fabricante de la placa recomienda un modelo, pero no se envía a España por las políticas del país de origen.

No obstante, da una serie de directrices para elegir batería: Asegurarse de que sea tipo Polímero de litio (Lipo) recargable de 3.7V y que la orientación de los cables de tierra y positivo sea la correcta.

El modelo en cuestión elegido se utiliza en drones de reducidas dimensiones. Cumple tanto las especificaciones del esquemático como las recomendaciones del fabricante, pero tiene la polaridad invertida. Esto es fácilmente solucionado alternando el orden de los cables en el conector.



Figura 10-3 Batería de dron de pequeñas dimensiones.

La batería es colocada en el hueco existente entre las dos placas, quedando así integrada sin aumentar las dimensiones al sistema.

10.2 Colocación en usuario

El siguiente paso ha sido la colocación del sistema en el cuerpo de una persona, para que empiece a cobrar forma de wearable. Como se comentó en el apartado de descripción del hardware, la línea de productos Lilypad está diseñados para ser cosida a prendas por lo que se optó por esta solución.

En una de las reuniones con personal dedicado a la práctica y entrenamiento de este deporte se indicó que las zonas que más importancia cobraban en estos movimientos halterófilos eran las comprendidas entre la zona de la cadera y la planta del pie.

De ahí surgió la idea de utilizar las rodilleras de compresión que se suelen usar en la práctica de este deporte para alojar el módulo. Estas prendas son robustas y ajustadas, por lo que evitan que el acelerómetro se mueva resultando en lecturas erróneas.



Figura 10-4 Atleta en CrossFit Games 2017 [24]

Se ha colocado el acelerómetro en la parte frontal de la prenda, que corresponde con la parte frontal de la rodilla. Se ha elegido esta posición ya que registraría más rango de movimiento.

El módulo de comunicación y el microcontrolador se han colocado en un lateral alejado de la articulación para evitar posibles molestias en el movimiento o roces con la barra en el momento de subirla desde el suelo.

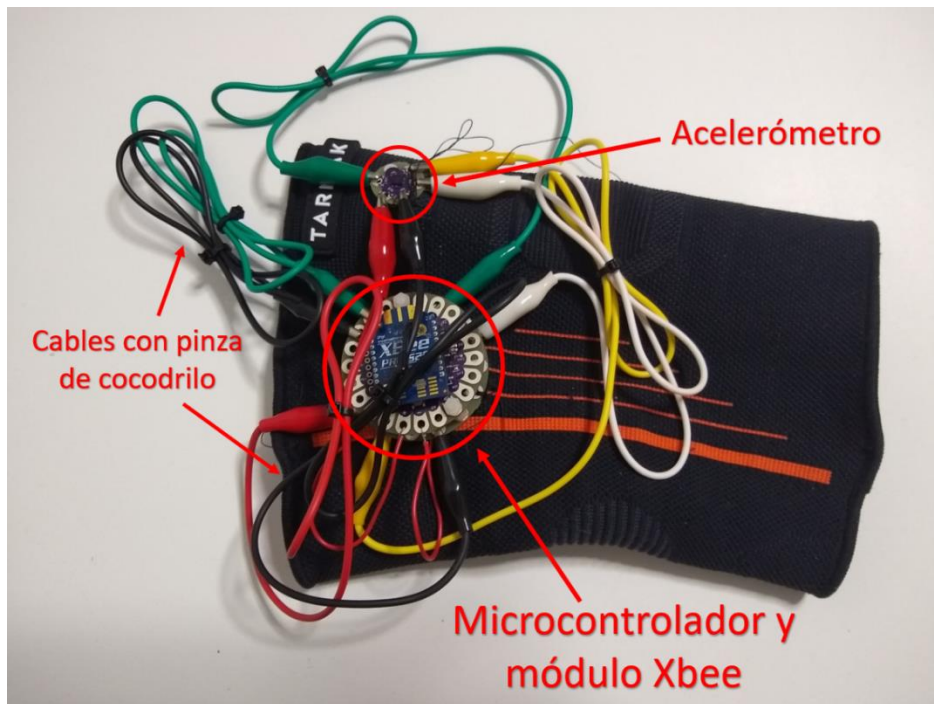


Figura 10-5 Prototipo colocado en rodillera de compresión.

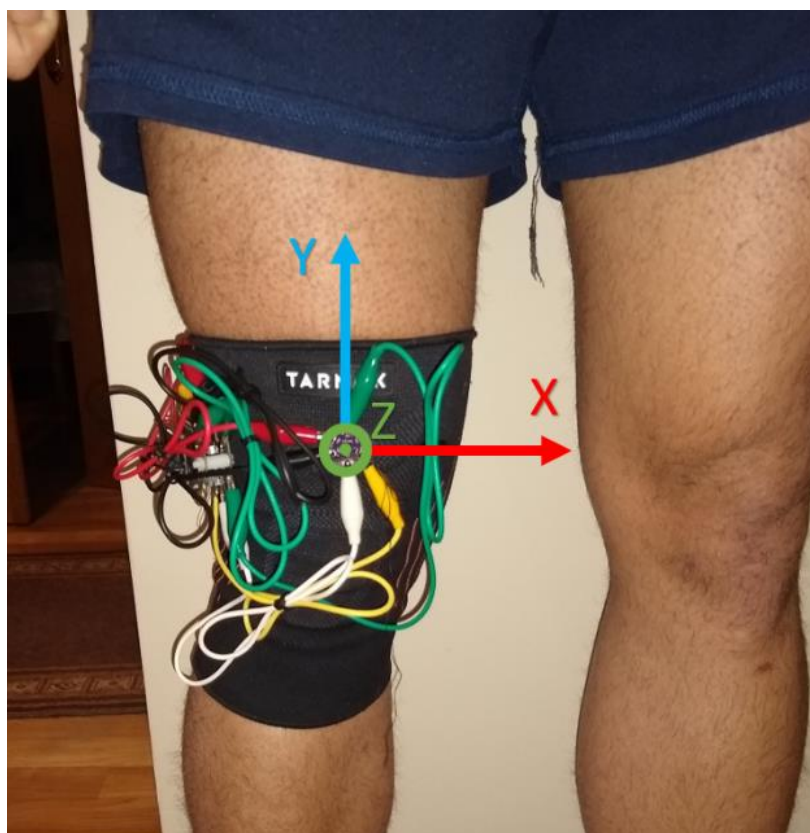


Figura 10-6 Prototipo colocado en rodilla. Ejes del acelerómetro.

11 Primeras medidas de movimientos en usuario

Tras tener listo el prototipo, se ha probado en el laboratorio algún ejercicio básico como son las sentadillas. En una de las pruebas se ha realizado 3 repeticiones simultaneas de dicho ejercicio para comprobar si obtiene algún patrón de movimiento.

Al representar en Matlab los valores de aceleración de los 3 ejes respecto al número de muestras podemos ver que el tanto en el eje Y como en el Z obtenemos una periodicidad clara. El eje X sin embargo registra pequeñas oscilaciones laterales. Este resultado es debido a la forma en la que se ha colocado el acelerómetro.

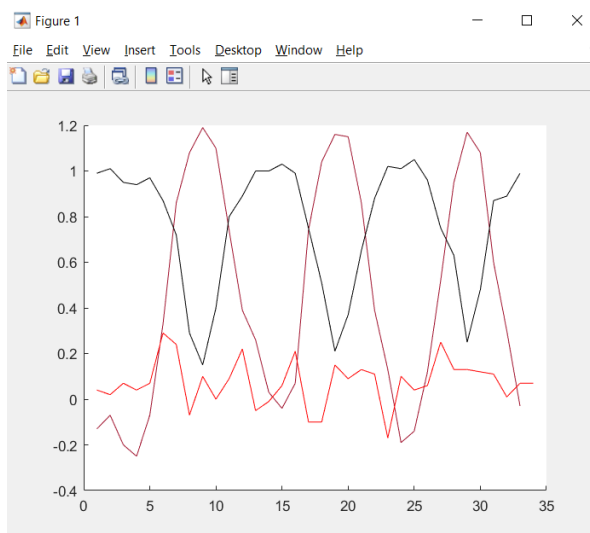


Figura 11-1 Valores de Aceleraciones (g) en el eje de las ordenadas y numero de la muestra en el eje de abscisas. Obtenidos de 3 sentadillas simples.

En posición de inicio del movimiento, estando el sujeto de pie, la aceleración existente es la debida a la gravedad terrestre. Esta va descendiendo ya que el eje Y deja de estar alineado con la vertical al comenzar a realizarse la sentadilla.

Sin embargo, en el eje Y ocurre lo contrario, de estar perpendicular al inicio del movimiento, acaba estando alineado a la vertical lo que hace que la componente de aceleración gravitacional incremente el valor de la aceleración.

Es decir, se está produciendo una rotación del acelerómetro respecto al eje X, o también llamada rotación Roll, de 90° aproximadamente.

11.1 Ejemplo de aplicación. Validación de repetición en competición de powerlifting.

En competiciones oficiales de levantamiento de peso existen una serie de reglas para validar si un movimiento es correcto o no. Por ejemplo, en la modalidad deportiva de “*powerlifting*” (no confundir con halterofilia) se realiza un ejercicio el cual se ejecuta con la barra apoyada en los hombros por detrás de la nuca.

En este ejercicio, se debe bajar con el peso en posición de sentadilla hasta una profundidad en la cual la articulación de la cadera quede por debajo de la articulación de la rodilla.

Si no se llegara a ese punto no se podría dar por valido el ejercicio, y de nada serviría que el atleta consiguiera elevar de nuevo la carga ya que se consideraría que no se ha cumplido con el rango de movimiento exigido.

Actualmente esto es revisado por un jurado compuesto por 3 árbitros y en ocasiones suelen encontrarse situaciones de discrepancia entre ellos. Tanto es así, que se ha llegado a plantear la inclusión de un sistema de video arbitraje similar al que se empezó a usar en el campeonato mundial de fútbol de Rusia (2018).

THE BACK SQUAT

- Shoulder-width stance
- Bar rests on upper back
- Hands just outside shoulders
- Full grip on the bar
- Elbows pointed back
- Hips descend back and down
- Hips descend lower than knees
- Lumbar curve maintained
- Heels down
- Knees in line with toes



CrossFit

Figura 11-2 Indicaciones de movimiento “Back Squat”. CrossFit. [25]

En CrossFit también se realiza este ejercicio y las directrices oficiales incluyen la regla anteriormente explicada, por lo que se aplica en competiciones oficiales de este tipo de deporte.

Se ha querido aprovechar el emplazamiento del acelerómetro en el prototipo para aplicar este reglamento.

Como se puede apreciar en el dibujo, la repetición es totalmente correcta, ya que la rodilla se encuentra más elevada que la unión de la pierna en la cadera. A esta posición corporal se denominará “punto crítico” a partir de ahora, en favor de hacer más sencilla la explicación.

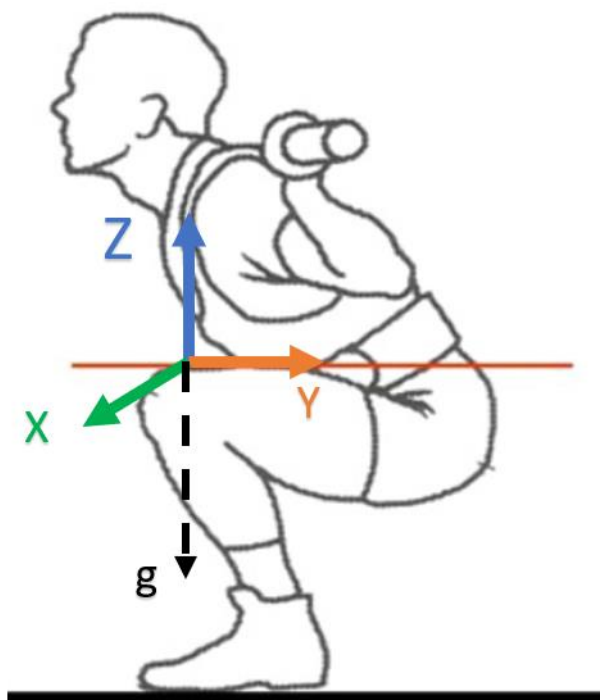


Figura 11-3 Representación anatómica humana durante el ejercicio “back squat” en el instante de punto crítico [26]

En este escenario, el acelerómetro colocado en la rodilla se encontraría con el eje Z alineado con el plano vertical. Y los ejes Y y X estarían en paralelo con la línea roja del dibujo.

Se quiere intentar registrar durante la ejecución del movimiento si se llega al punto crítico.

Para ello se calculará el ángulo ρ formado por el vector de aceleración y el vector de gravedad terrestre, (G_x, G_y, G_z) y $(0, 0, 1)$ respectivamente, expresados en la unidad múltiplo de la gravedad terrestre.

$$\cos \rho = \frac{Gz}{\sqrt{Gx^2 + Gy^2 + Gz^2}}$$

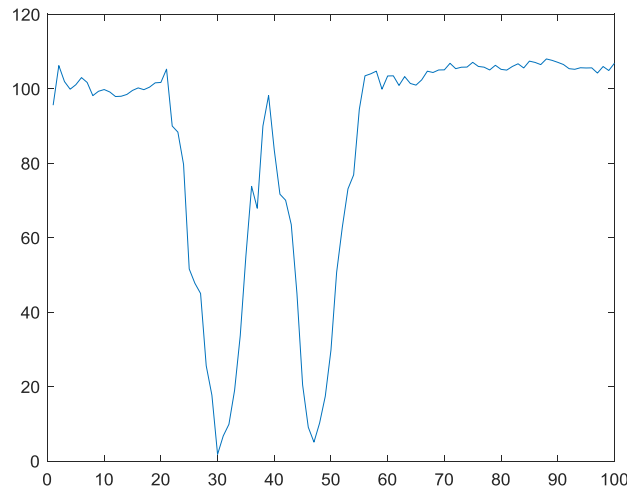


Figura 11-4 Ángulo medido en grados entre el eje Z y el vector de aceleración de la gravedad terrestre.

En la gráfica se observan dos movimientos de subida y bajada, en el que el primero es más profundo y el eje Z del acelerómetro llega a estar casi totalmente alineado con el plano vertical. Esto podría ser una aproximación de un movimiento válido.

El problema es que, si se sobrepasa la alineación, el ángulo no se mostraría como ángulo negativo, sino que el resultado sería positivo por lo que no se podría saber si se sobrepasa el punto crítico.

Pronto se comprendió que en la posición inicial del acelerómetro los ejes se encuentran rotados, por lo que el eje Y está actuando como eje Z, el X como Y y el Y como el X. Por lo que interesa comprobar el ángulo del eje Y con la vertical. Con estas nuevas consideraciones, el ángulo β sería 90° en el punto crítico.

$$\cos \beta = \frac{Gy}{\sqrt{Gx^2 + Gy^2 + Gz^2}}$$

En un principio el ángulo será próximo a 0° porque se encontrará casi alineado y conforme avance el movimiento se ira acercando a los 90° de la posición de la figura 11-3

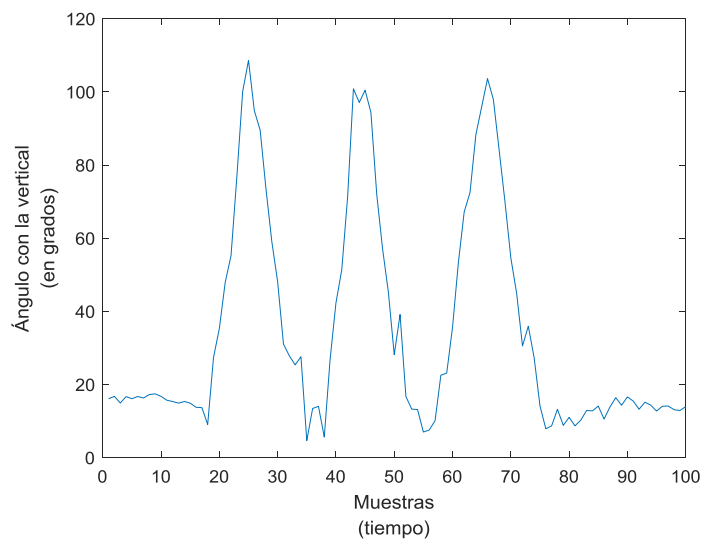


Figura 11-5 Ángulo medido en grados entre el eje Y y el vector de aceleración de la gravedad terrestre

Esta vez se han registrado 3 sentadillas lo más profundas posibles, pero de forma correcta, para forzar sobrepasar esa situación de 90° y como se puede observar en la gráfica, las 3 sobrepasan ese nivel. Así pues, se puede determinar que son 3 repeticiones validas según los reglamentos oficiales.

Se ha considerado oportuno realizar este pequeño experimento con una medida sencilla relacionada con un movimiento de Crossfit en concreto para demostrar que los acelerómetros no solo pueden medir aceleraciones, sino que son interesantes para utilizar en cálculos vectoriales obteniendo así su posición.

Se podría añadir un segundo acelerómetro en el otro extremo de la articulación si fuera de interés para medir el ángulo formado por esta unión. Similar a como se realizó de forma cableada en una maqueta de madera en *“Study of the Lower Limb’s Angle During Weightlifting Exercises Using an Accelerometer-Based System”*(2014).

12 Red multipunto

Una vez conseguido una conexión punto a punto correcta y haber sido capaces de procesar los datos, el siguiente paso es añadir más nodos a la red.

Para ello añadimos otro nodo con la misma configuración.

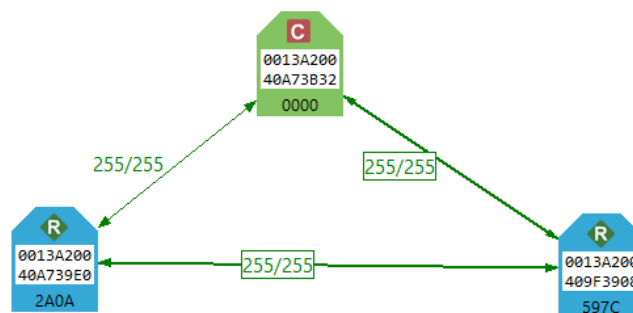


Figura 12-1 Esquema de red obtenido en XCTU.

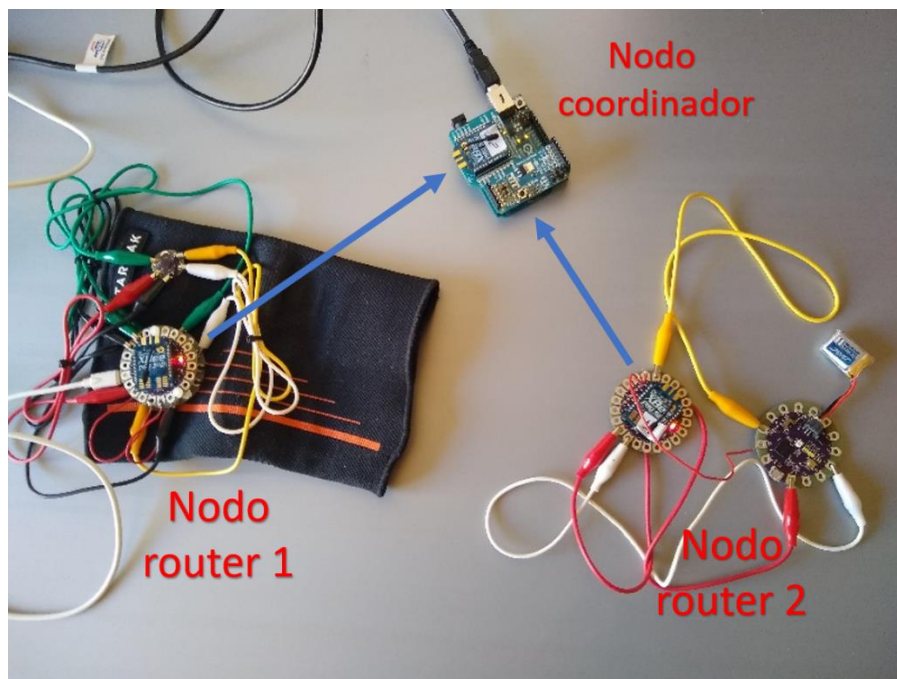


Figura 12-2 Elementos reales que forman la red.

Aunque al observar la imagen de red proporcionada por XCTU dé la sensación de que se trata de una red en forma de malla, no es así. XCTU marca las conexiones posibles, pero

al haberse configurado los módulos router con la dirección destino cross, la verdadera tipología de red es la de tipo estrella. No es de interés de momento que los nodos colocados en el cuerpo se comuniquen entre ellos.

Se han realizado pruebas y se ha recibido tramas API de los dos destinos, perfectamente identificables gracias a la identificación de nodo que poseen estos paquetes.

El nuevo nodo no se ha compactado ni colocado en ninguna prenda ya que es necesario antes comprobar si el sistema se comporta de forma adecuada en pruebas de campo. Además, habría que estudiarse más a fondo la dinámica de los ejercicios para saber en qué parte del cuerpo del atleta colocarlo.

13 Adecuación del sistema para pruebas de campo.

En este apartado se describen las medidas que se tomaron para dotar al sistema de un funcionamiento más eficaz para la posible prueba en un gimnasio de CrossFit.

13.1.1 GUI en Matlab

Para la facilitar el proceso de adquisición de datos se ha desarrollado una interfaz gráfica en Matlab mediante la herramienta “APP Designer”. En este entorno se rellenarán varios aspectos que puedan ser de interés para el posterior procesado de la muestra.

Estas variables son:

- El tipo de ejercicio que se realiza.
- El peso, medido en kilogramos, con el que se ha realizado dicho ejercicio.
- El modo de ejecución. Si se ha realizado una única repetición o varias encadenadas.

Todos estos parámetros son almacenados en variables en el espacio de trabajo que es guardado en archivo “.mat”.

Cuando se quiera analizar las medidas a posteriori se podrán cargar mediante el comando “load” junto al nombre de la muestra. Tras esto se tendrá acceso a los tres vectores compuestos por las muestras de aceleraciones y a todas las demás variables que se introduzcan en los espacios editables de la interfaz gráfica.

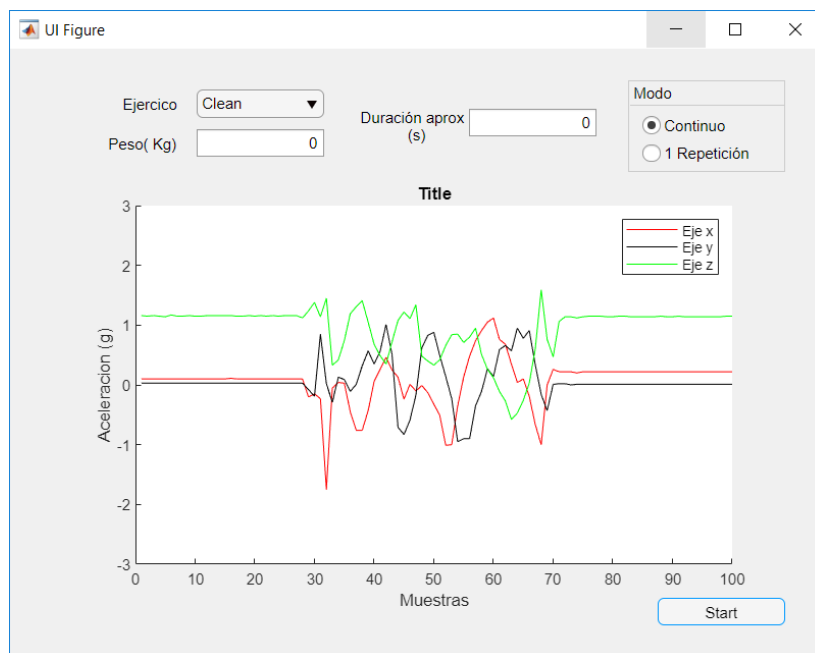


Figura 13-1 GUI creada en Matlab mediante la herramienta “App Designer”

El interfaz cuenta con un espacio donde se pueden ver los valores que se van adquiriendo en cada eje, de esta forma, se puede observar si las medidas están siendo correctas.

13.1.2 Sustitución de cables.

Los cables con pinza de cocodrilo han sido de gran ayuda para los prototipados rápidos que se han estado realizando hasta este punto, pero no serían cómodos para la práctica deportiva.

Por esta razón se consideró sustituirlos por cables más cortos que fueran soldados a los pines. Para ello se utilizó un cable flexible plano o cable FFC compuesto por 5 vías, 2 de ella para la alimentación y 3 para para las salidas asociadas a los ejes de los acelerómetros.

Se retiró la cubierta de los 6 cables a la vez con un alicate pelador de cables multifuncional ajustable para evitar dañar los finos filamentos. A continuación, se separaron las vías en los dos extremos y se soldaron a los pines correspondientes.

Se dejó una largura de cable superior para que los movimientos del atleta no supusieran tirantez y acabaran arrancando los puntos de soldadura o la rotura de la vía.

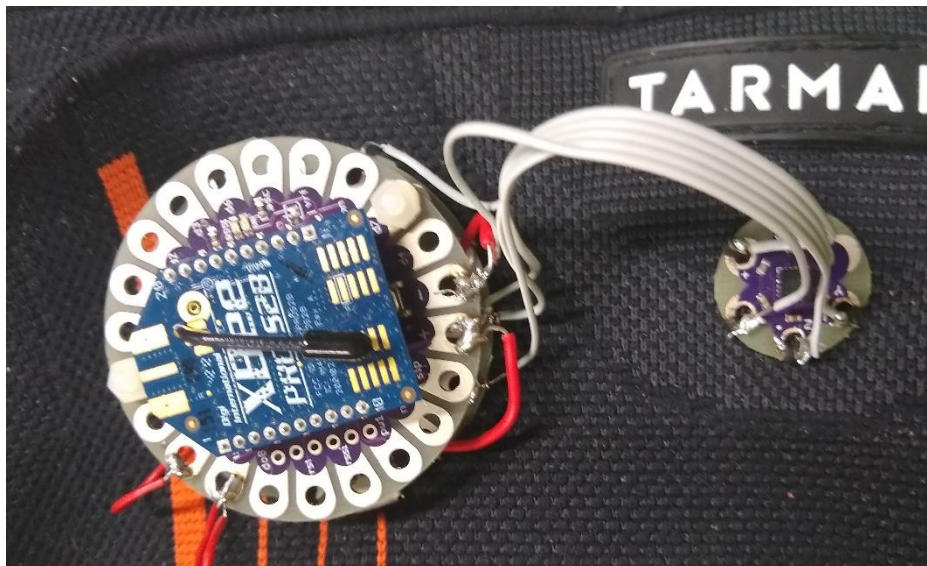


Figura 13-2 Nodo router colocado en rodillera.

Se consiguió reducir las dimensiones considerablemente además de aportarle una estética limpia y ordenada, aspecto fundamental para hacerle sentir cómodo al atleta.

14 CrossFit Runa

Desde un primer momento se pensó en la posibilidad de realizar pruebas de campo. Además de la obtención de datos más reales y de mayor interés para su futuro análisis, servirían para comprobar la acogida que tienen las posibles soluciones tecnológicas aplicadas a este deporte.

Por ello se acudió a Pablo Colin, entrenador Crossfit certificado, atleta y dueño de CrossFit Runa. Aparte de proporcionar puntos de vista muy interesantes, también nos ofreció el material deportivo y el espacio para la realización de las pruebas que se van a detallar en este apartado.



Figura 14-1 Vista parcial de CrossFit Runa

Una vez en el gimnasio se colocó el nodo coordinador sobre un cajón de madera a una altura de unos 60 cm. A lado se colocó el ordenador personal por cuyo puerto serie USB se leyeron los datagramas mediante el software Matlab. En este equipo se utilizó la GUI desarrollada que agilizó enormemente el proceso, ya que todas las medidas eran automáticamente guardadas e identificadas para su posterior revisión.

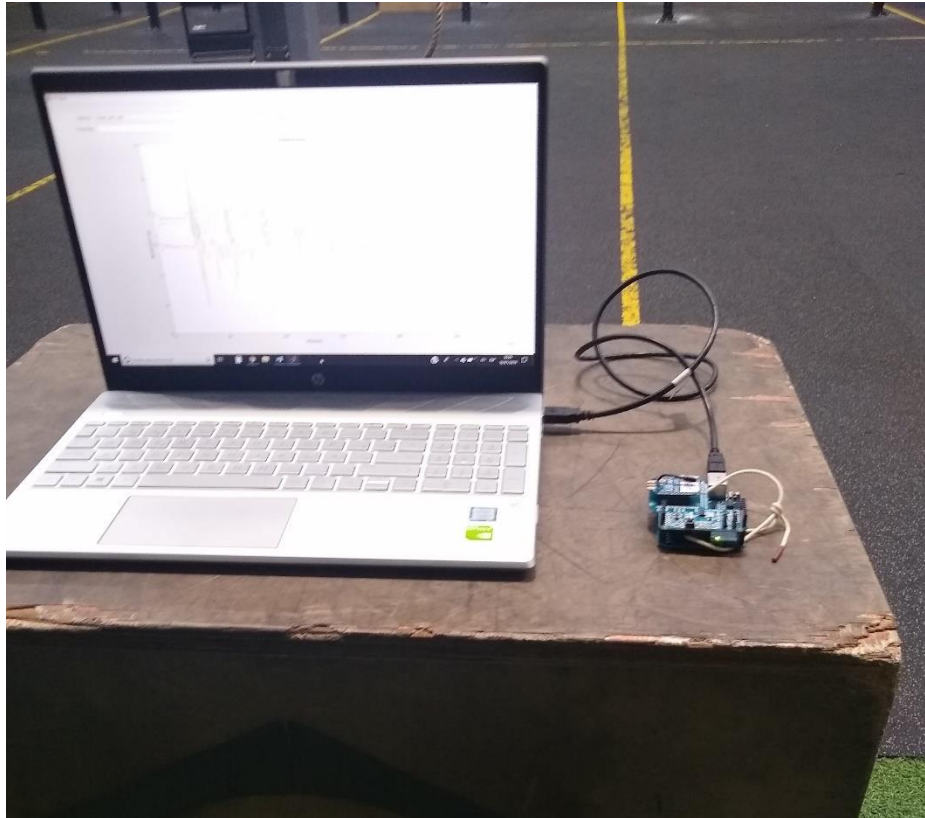


Figura 14-2 Nodo coordinador junto al ordenador personal con el software Matlab

El nodo router encargado del muestreo de aceleraciones fue colocado en la rodilla del atleta como muestra la *Figura 14-3*



Figura 14-3 Atleta con el dispositivo de medida colocado.

14.1 Resultados de mediciones

Se realizaron pruebas en dos tipos de modo de ejecución. El primer modo trata de una sola repetición, con pausas estáticas en la posición de inicio y final para conocer también la posición del sensor en dichos instantes.

El segundo es el modo continuo, es decir, repeticiones del mismo ejercicio para observar la periodicidad de los datos.

14.1.1 “Snatch” o Arrancada.

En este ejercicio, se parte de una posición recogida, con las rodillas flexionadas, ya que la barra se encuentra en el suelo. En un solo movimiento el atleta debe de ser capaz de colocar la barra por encima de su cabeza y extender completamente brazos y piernas.

Con pesos muy altos es importante que, tras el tirón inicial, el atleta se coloque debajo de la barra en el momento preciso en el que la barra se encuentre con aceleración cero (*Figura 14-4a*) Si este movimiento se produjese demasiado tarde podría desembocar en que no se consiguiese realizar el ejercicio.



Figura 14-4 Atleta en posición de recepción de la barra y en posición final del ejercicio “snatch” de izquierda a derecha respectivamente.

La repetición única de este ejercicio con sus correspondientes pausas arrojó las 3 siguientes señales.

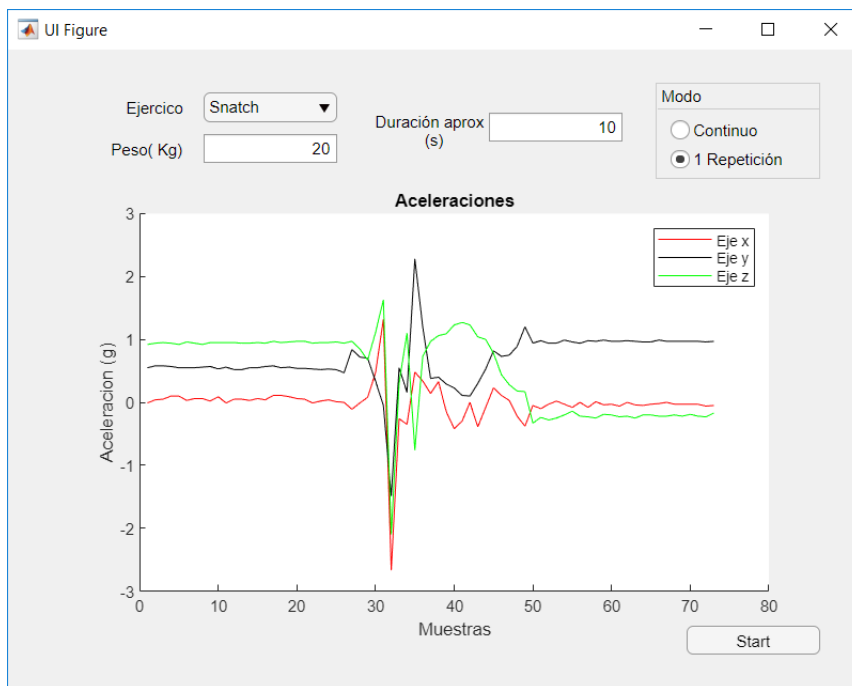


Figura 14-5 Gráfica de aceleraciones del movimiento “snatch” realizado una única vez

La posición inicial y la posición final son claramente diferenciables en la gráfica ya que el acelerómetro está en una posición diferente. Estos valores constantes diferentes de cero, aunque el atleta se mantenga completamente inmóvil son debidos a la contribución de la aceleración de la gravedad.

El siguiente paso fue pedir al atleta que repitiera el movimiento varias veces consecutivas sin ningún tipo de pausa. Como se puede comprobar en las gráficas se puede ver la similitud entre la repetición única y cada una de las 3 repeticiones en modo continuo.

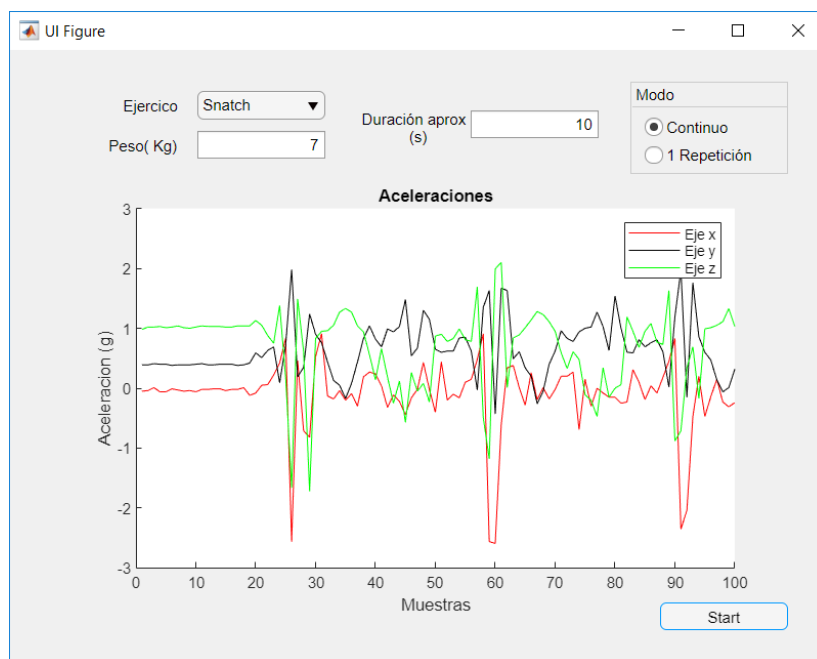


Figura 14-6 Gráfica de aceleraciones del movimiento “snatch” en modo continuo

Sorprende que una de las mayores aceleraciones observadas sea la referida al eje X, el cual se encuentra situado en el plano horizontal. Esto es debido a la rápida apertura que las rodillas realizan para que el atleta pueda bajar a la posición de recepción de la barra.

Como se ha comentado, este movimiento tiene que ser rápido para evitar que la barra pierda la inercia de subida que ha adquirido tras el tirón inicial. De otro modo, la barra rápidamente empezaría a descender y empujaría al atleta hacia abajo, que no podría soportar la recepción del peso acelerado.

14.1.2 “Clean and jerk” o dos tiempos

Este ejercicio, al contrario que en el anterior, se realiza en dos fases. La primera fase se llama “clean” o cargada y la segunda “jerk” o enviñ. Cada fase comienza por un tirón, el primero para conseguir montar la barra sobre los hombros desde el suelo y el segundo para elevar la barra desde la posición final de la primera fase hasta situar la barra por encima de la cabeza con brazos y piernas extendidos.

Puede realizarse con un tiempo de descanso entre las dos fases o hacerse de forma más fluida para aprovechar la inercia de la primera fase.



Figura 14-7 Atleta en el instante de recepción de la barra en el movimiento "clean"

Primero se realizó una medición de cargada. No se realizó medición de la segunda fase por ser menos interesante a nivel de la articulación que se estaba monitorizando.

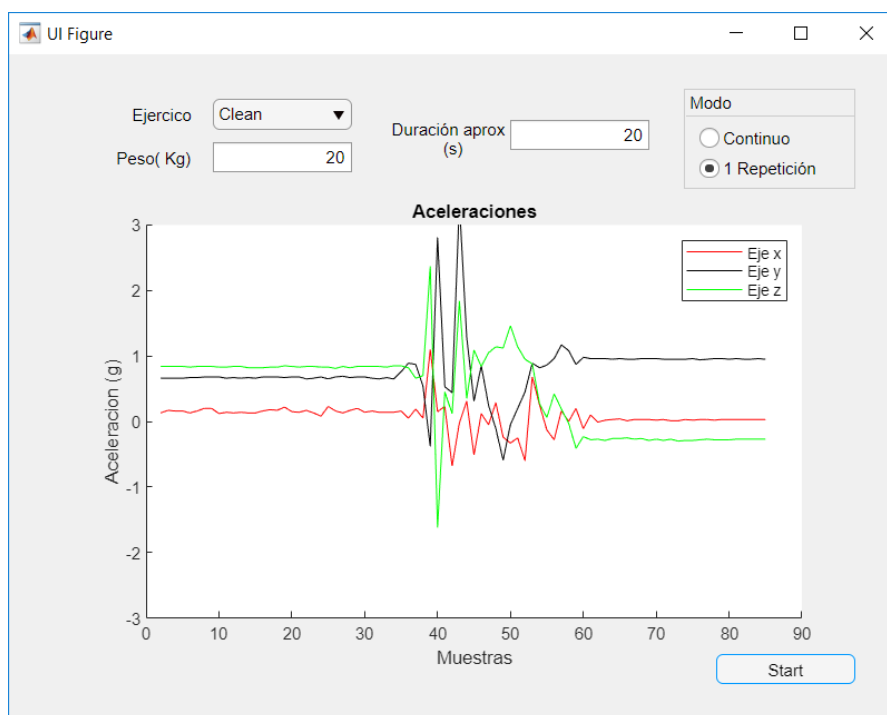


Figura 14-8 Gráfica de aceleraciones del movimiento "clean" realizado una única vez

En la siguiente medida, en modo continuo, fue registrado el movimiento completo con sus dos fases encadenadas por repetición. Este movimiento se suele realizar de esta forma con poco peso para añadir mayor componente cardiovascular al entrenamiento.

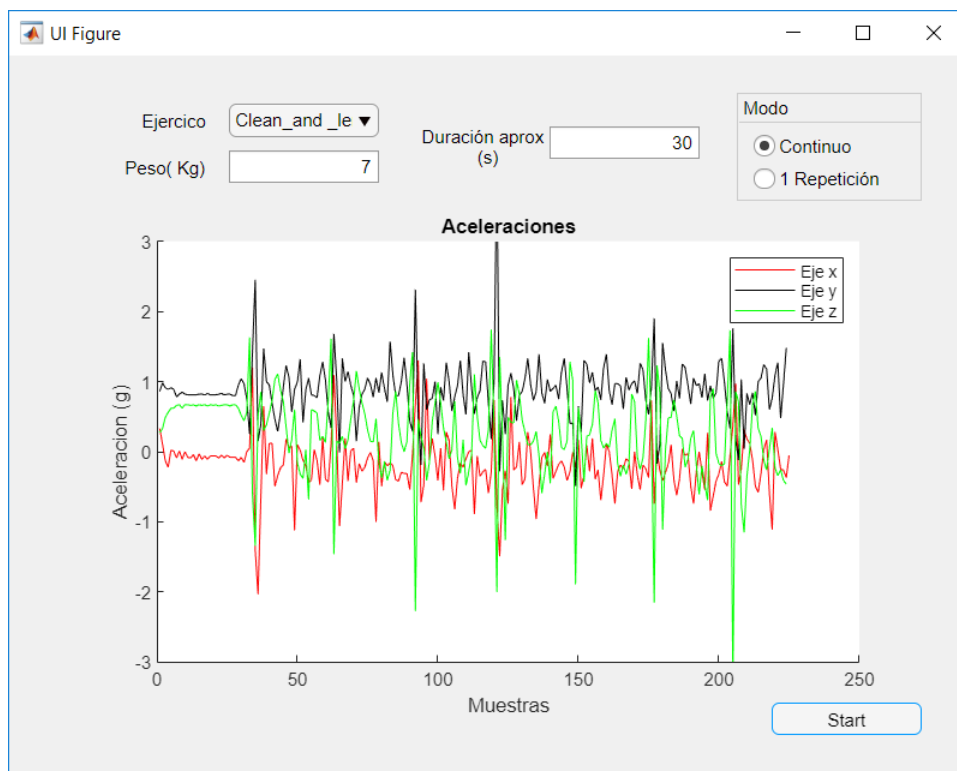


Figura 14-9 Gráfica de aceleraciones del movimiento "clean and jerk" realizado en modo continuo

Como se puede apreciar esta aporta duración y unos picos en el eje X más pequeños que los procedentes a la primera fase, la cual tiene picos positivos en el eje Y y negativos en el eje Z.

Es interesante observar la variación de la amplitud de los picos. En algunos casos excede el límite del rango y en otros es casi inexistente como ocurre en la muestra 150 del eje Y.

Viendo los resultados de cada una gráfica parece razonable relacionar el número de picos con las repeticiones realizadas. Estos picos pueden ser identificados mediante funciones de MATLAB. Características como la frecuencia de repetición y la distribución de los valores de las muestras podrían ser relacionadas con el tipo de ejercicio.

Añadir más nodos con sus correspondientes medidas procesadas en más partes del cuerpo aumentarían la fiabilidad del reconocimiento de ejercicio.

Con poco más de desarrollo se podrían aplicar a competiciones cronometradas en las que se tuvieran que realizar un número prefijado de repeticiones de diferentes tipos. Básicamente como cualquier entrenamiento CrossFit, pero con la posibilidad de crear una clasificación digital.

15 Conclusión

Durante el desarrollo de este proyecto se ha visto muy viable la utilización de esta tecnología para la obtención de datos sobre dinámicas de este tipo de entrenamiento. Se han conseguido datos muy interesantes y que contienen grandes posibilidades de posterior procesado.

Además, se ha conseguido desarrollar un prototipo de dispositivo, de bajo coste y consumo que no ha obstaculizado en absoluto la práctica de este deporte. Este puede ser colocado cosido en cualquier prenda de compresión o en tiras de velcro, lo que permite ser colocado en casi cualquier parte del cuerpo de manera muy sencilla.

Las medidas en el gimnasio, aparte de para obtener medidas de aceleraciones de movimientos con una ejecución adecuada, han servido para comprobar la acogida que tiene la tecnología en este marco. Se ha obtenido una respuesta muy positiva de profesionales de este deporte.

Estos comentaron que no existe actualmente nada dedicado a este tipo de entrenamiento. Las aproximaciones más cercanas son dispositivos encargados de medir movimientos muy lineales y de baja complicación. Estos dispositivos a los que se refirieron fueron los que se habían descrito en el apartado “Modelos comerciales” del capítulo “Estado del arte”

Si se considerase seguir con el desarrollo habría que plantearse la aplicación de este, si se le quiere dar un propósito de investigación o si se quiere desarrollar prototipos destinados a usuarios finales que deseen conocer más acerca de sus entrenamientos.

Se observa que entre los deportistas de CrossFit se admira la velocidad que tienen algunos atletas para realizar cientos de movimientos complejos con alto peso o la capacidad que tienen para realizar entrenamientos predefinidos en poco tiempo. Esto podría ser algo que se pudiera mostrar a los espectadores, en demostraciones o en competiciones que se organizan en todos los gimnasios, mediante pantallas, en un servicio web o en una app móvil. Incluso podría estar presente esta tecnología en competiciones oficiales, ofreciendo a los miles de espectadores una experiencia sin precedentes.

16 Líneas Futuras

Una vez obtenido un prototipo de nodo, se podría seguir diseñando más nodos. Se podría por ejemplo dotar a una prenda como una camiseta de compresión de varios nodos repartidos por su superficie. Para ello se debería realizar un estudio más en profundidad para determinar que otros puntos del cuerpo son apropiados para obtener medición.

Sería de gran interés añadir más tipos de sensores, como por ejemplo sensores de ritmo cardiaco en muñequera o en un nodo en el pecho. Esto nos permitiría correlacionar las pulsaciones con el tipo de ejercicio o la cantidad de carga.

Sensores mioeléctricos también serían interesantes si se quiere profundizar en el estudio del movimiento. Podrían arrojar datos muy interesantes sobre posibles descompensaciones o erróneas inactivaciones musculares que pudieran desembocar en lesiones de los músculos antagonistas, es decir, los que recibirían esa carga extra.

El prototipo aguanta pruebas de movimientos controlados, pero no deja de ser un elemento grande debido a que las placas son de prototipado. Se podría diseñar un PCB con un microcontrolador más pequeño y embeber todo el nodo en una carcasa. Para esto, por ejemplo, se podría realizar un diseño ergonómico que se pudiera imprimir en una impresora 3D.

Ligado a este posible nuevo prototipo, se podría considerar oportuno la utilización de la otra versión de módulos Xbee, los SMT, que se integran directamente al PCB mediante puntos de soldadura.

Si se considerase la opción de sistema destinado a gimnasios o a atletas, una opción a tener en cuenta sería la utilización de herramientas de Matlab como clasificadores, que una vez entrenados con una muestra de datos lo suficientemente grande, pudieran determinar qué ejercicio se está realizando para adaptar la salida de información al mismo.

Quizá una aplicación interesante sería el registro de varios atletas de modo simultaneo. De este modo se podría comprobar en tiempo real la diferencia entre ellos. Pudiéndose contar repeticiones, medir potencias, o controlar el ritmo cardiaco de ellos.

Los gimnasios de esta práctica son muy interesantes y suelen tener una composición similar. Son locales amplios con estructuras metálicas que sirve tanto para colocación de barras con peso como para ejercicios gimnásticos con barra. Se podría realizar pruebas de trazados de rayos teniendo en cuenta los materiales de estas estructuras y suelo, posibles espacios diáfanos, las personas que puedan estar entrenando, etc. Se podría obtener un modelo aproximado del canal para calcular las pérdidas de transmisión.

Aunque los módulos inalámbricos son de bajo consumo, existe la posibilidad de configurar para que emitan a menor potencia. Tras los resultados de experimento de trazado de rayos se podría estudiar hasta qué punto se podría reducir esta potencia para seguir manteniendo conexión en el escenario objetivo. Después se deberían realizar múltiples medidas para comparar los datos teóricos y prácticos. Reducir la potencia de emisión aumentaría la vida útil de las baterías.

17 Hojas de características

Acelerómetro ADXL 335.

<https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf>

Microprocesador ATmega32U4. <https://cdn.sparkfun.com/datasheets/E-Textiles/Lilypad/ATmega16U4-32U4.pdf>

Esquemático de la placa Lilypad USB plus. <https://cdn.sparkfun.com/datasheets/E-Textiles/Lilypad/Lilypad Arduino USB v12.pdf>

Esquemático de Lilypad Xbee. <https://cdn.sparkfun.com/datasheets/E-Textiles/Lilypad/LilyPad XBee v10.pdf>

Modulo XBee Pro S2B. <https://www.digi.com/support/productdetail?pid=4549>

18 Referencia de imágenes

- [1] Atleta en CrossFit Games 2015.
<https://boxlifemagazine.com/2015-crossfit-games-recap/>
- [2] Prototipo y resultado de impulsos de los músculos vasto medio, vasto lateral y recto femoral en escenarios de asfalto, tierra y pista de atletismo.
https://www.researchgate.net/publication/301857324_Wearable_Embroidered_Muscle_Activity_Sensing_Device_for_the_Human_Upper_Leg
- [3] Diferentes ejercicios de peso corporal autoidentificados con sus correspondientes números de repeticiones. Captura de video demostración:
https://www.youtube.com/watch?v=zpa4rVGIO68&list=LL0q2u-_oohQCn9pqpXWn2gg&index=31&t=0s
- [4] RecoFit. Hardware de adquisición colocado en el antebrazo del usuario mediante una cinta elástica. http://delivery.acm.org/10.1145/2560000/2557116/p3225-morris.pdf?ip=130.206.159.13&id=2557116&acc=ACTIVE%20SERVICE&key=DD1EC5BCF38B3699%2E2A477AE0AF0D64C7%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&_acm_=1547997336_b042b01721aa2392517099dc9d515570
- [5] Identificación de los picos de las señales para identificar patrones.
http://delivery.acm.org/10.1145/2560000/2557116/p3225-morris.pdf?ip=130.206.159.13&id=2557116&acc=ACTIVE%20SERVICE&key=DD1EC5BCF38B3699%2E2A477AE0AF0D64C7%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&_acm_=1547997336_b042b01721aa2392517099dc9d515570
- [6] Ejemplo de uso de un encoder lineal en sentadilla trasera.
https://bikettraining.es/wp-content/uploads/2018/02/IMG_9477web.jpg
- [7] Regresiones lineales entre la velocidad de ejecución y la carga medida en porcentaje de repetición máxima en sentadilla profunda y sentadilla de 90°.
https://www.researchgate.net/publication/282149641_Movement_velocity_as_a_measure_of_exercise_intensity_in_three_lower_limb_exercises
- [8] Encoder lineal Speed4lifts. Disponible en: <https://www.speed4lifts.com/es/>
- [9] Beast Strength Pack 5. Disponible en:
<https://www.thisisbeast.com/en/products/beast-coach-package>
- [10] Push Band. Disponible en: <https://www.trainwithpush.com/push-band>
- [11] Imagen IEEE 802.15 IEEE 802.15 Working Group for Wireless Specialty Networks (WSN). <http://www.ieee802.org/15/>
- [12] Modelo de capas de Zigbee :

https://www.digi.com/resources/documentation/Digidocs/90002002/Content/Reference/r_zb_stack.htm?TocPath=zigbee%20networks|_____3

[13] Esquema conexión de dos UART y secuencia de bits

<http://www.circuitbasics.com/basics-uart-communication/>

[14] Placa Arduino Lilypad USB plus. Disponible en:

<https://www.sparkfun.com/products/14631>

[15] Lilypad Xbee. Disponible en: <https://www.sparkfun.com/products/12921>

[16] Arduino Uno+Shield+Módulo Inalámbrico.

<https://www.sparkfun.com/products/retired/8471>

[17] Módulo XBee <https://xbee.cl/xbee-pro-63mw-wire-antenna/>

[18] Ejemplo de antena para conector U.FL. <https://robu.in/product/zigbee-xbee-pro-module-s2c-802-15-4-63mw-3km-antenna-xbp24cz7sit-004/>

[19] Ejemplo de antena para conector RP-SMA.

http://mx.melinterest.com/?r=site/search&seller_id=87511030&seller_nickname=VIC03

[20] Tabla de modelos comerciales basada en: https://www.sparkfun.com/pages/xbee_guide

[21] Acelerómetro Lilypad. Disponible en: <https://www.sparkfun.com/products/9267>

[22] Fragmento de la hoja de características del acelerómetro ADXL 335. <https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf>

[23] Parte del esquemático de la placa Lilypad USB plus. https://cdn.sparkfun.com/datasheets/E-Textiles/Lilypad/Lilypad_Arduino_USB_v12.pdf

[24] Atleta en CrossFit Games 2017 <https://fitness-und-gesundheit.de/bericht-everroll-ist-offizieller-sportboden-der-crossfit-games-11149.html>

[25] Directrices del ejercicio *Back squat*. <https://journal.crossfit.com/article/the-back-squat>

[26] “*Powerlifting Squat Rules*”. <https://www.powerliftingtowin.com/powerlifting-squat-rules/>

19 Bibliografía y referencias

SparkFun Electronics®. LilyPad USB Plus Hookup Guide. Niwot, Colorado. [Consulta 16 de noviembre 2018] Disponible en: <https://learn.sparkfun.com/tutorials/lilypad-usb-plus-hookup-guide/all>

SparkFun Electronics®. LilyPad Basics: Powering Your Project. Niwot, Colorado. [Consulta 21 de diciembre 2018] Disponible en: https://learn.sparkfun.com/tutorials/lilypad-basics-powering-your-project?_ga=2.104961931.969227032.1545159815-2069018203.1544373134

Enrique Crespo. Aprendiendo Arduino [Blog]. Primeros pasos en Arduino. [Consulta 2 de Enero 2019] Disponible en: <https://aprendiendoarduino.wordpress.com/category/c/>

Enrique Crespo. Aprendiendo Arduino [Blog]. Aprendiendo Arduino. Guía XBee [Consulta: 12 de diciembre 2018]. Disponible en: <https://aprendiendoarduino.wordpress.com/tag/xbee/>

TE Connectivity. Choose the right type of accelerometer. [Consulta 20 de octubre 2018] Disponible en: <https://www.te.com/usa-en/industries/sensor-solutions/insights/types-of-accelerometers.html>

CrossFit®. CrossFit, Inc Washington DC , USA. Disponible en: <https://www.crossfit.com/>

Digi International Inc. XBee®/XBee-PRO S2C Zigbee® [Consulta 27 de diciembre 2018] Disponible en: <https://www.digi.com/resources/documentation/digidocs/pdfs/90002002.pdf>

SparkFun Electronics®. Exploring XBee and XCTU. Niwot, Colorado [Consulta 12 de octubre 2018] Disponible en: https://learn.sparkfun.com/tutorials/exploring-xbees-and-xctu?_ga=2.147760638.836473004.1546865599-2069018203.1544373134

DIAZ, R.A. y MARTÍNEZ, J.C. Study of the Lower Limb's Angle During Weightlifting Exercises Using an Accelerometer-Based System (2014) Cartagena, Colombia. DOI: 10.1109/CIIMA.2014.6983443

MARK PEDLEY. Tilt Sensing Using a Three-Axis Accelerometer. NXP Freescale Semiconductor [Consulta: 4 de enero 2019]. Disponible en: https://cache.freescale.com/files/sensors/doc/app_note/AN3461.pdf

DAN MORRIS [et al.] 2014 "RecoFit: Using a Wearable Sensor to Find, Recognize, and Count Repetitive Exercises" New York, USA. ACM. ISBN: 978-1-4503-2473-1

BRZYCKI, M. Strength testing: Predicting a one-rep max from repstofatigue. (1993). JOHPERD 64:88-90.

CONCENICAO, F [et al.] Movement velocity as a measure of exercise intensity in three lower limb exercises. Journal of sports sciences. (2016) PMID: 26395837 DOI: 10.1080/02640414.2015.1090010.

MATLAB. Classification learner Matlab Video [Online Video] Disponible en: https://www.youtube.com/watch?v=GZ3KUPqA1JM&list=LL0q2u-_oohQCn9ppqXWn2gg&index=11&t=0s

RIBAS MANERO, RBR [et al.] , Wearable embroidered muscle activity sensing device for the human upper leg. in Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS. vol. 2016-October, 7592111, Institute of Electrical and Electronics Engineers Inc., pp. 6062-6065, 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2016, Orlando, United States, 16/08/2016. DOI: 10.1109/EMBC.2016.7592111

20 Código Fuente

20.1 Microprocesador Lilypad USB plus (Router.ino)

```
#include <SoftwareSerial.h>
SoftwareSerial xbee(SDA, SCL); //Definiendo los pines rx y tx
respectivamente
const int xpin = A5; //Definiendo pines de entrada de
aceleracion
const int ypin =A3;
const int zpin = A2;

double vref=3.30;
int steps=1023;
double sesnibility=0.3;

long startTime=millis();
void setup() {
// initialize the serial communications:
pinMode(xpin, INPUT);
pinMode(ypin, INPUT);
pinMode(zpin, INPUT);
Serial.begin(9600);
xbee.begin(9600);
}

void loop() {

double rx=analogRead(xpin);
delay(1);
double ry=analogRead(ypin); //Lectura de los puertos.
delay(1);
double rz=analogRead(zpin);

double converter= steps*sesnibility/vref; // Convertir a unidades
de gravedad

double GX= (rx-509.00)/converter;
double GY= (ry-512.00)/converter; //Se resta el valor ADC 0g
double GZ= (rz-527.00)/converter;

//Un paquete
String xyz="x"+String(GX,2)+"y"+String(GY,2)+"z"+String(GZ,2);
xbee.print(xyz);
delay(80);

long currentTime=millis();
long Time= currentTime-startTime;
Serial.println(Time);
startTime=currentTime;
}
```

20.2 Microprocesador Arduino Uno (CoordinatorXbee.ino)

```
#include <SoftwareSerial.h>
SoftwareSerial xbee(0,1);
byte lengthValue[2];
```



```
void setup() {
  Serial.begin(9600);
  xbee.begin(9600);
}
void loop() {

while(xbee.available()>16){

  if (xbee.read() == 0x7E) { //Comprueba el inicio de la trama
    Serial.print(0x7E,HEX); // Como ha sido leído se
imprime manualmente
    Serial.print(" ");
    lengthValue[0] = xbee.read();
    lengthValue[1] = xbee.read();
    int intLength = lengthValue[1] + (lengthValue[0] << 8); //Se
calcula el tamaño del paquete
    Serial.print(lengthValue[0],HEX ); Serial.print(" ");
    Serial.print(lengthValue[1],HEX ); Serial.print(" ");

    for (int i=0;i<intLength;i++){ //Se lee byte a byte en
funcion del paramtro de tamaño
    Serial.print(xbee.read(),HEX);
    //Serial.print(bytes[i],HEX);
    Serial.print(" ");

    }
    //Serial.print(xbee.read(),HEX); //El valor del checksum no se
envía, ///pero se podría
//enviar para comprobar posibles fallos
xbee.read(); //El checksum aunque no se encie hay
//que leerlo
    Serial.println(); //Salto de línea delimitador de trama
  }
}
}
```

20.3 GUI en Matlab (InterfazGrafica.mlapp)

```
classdef InterfazGrafica < matlab.apps.AppBase
  % Properties that correspond to app components
  properties (Access = public)
    UIFigure          matlab.ui.Figure
    UIAxes            matlab.ui.control.UIAxes
    StartButton       matlab.ui.control.Button
    EjercicioDropDownLabel matlab.ui.control.Label
    Ejercicio         matlab.ui.control.DropDown
    PesoKgEditFieldLabel matlab.ui.control.Label
    Peso              matlab.ui.control.NumericEditField
    DuracinaproxLabel matlab.ui.control.Label
    Duracion          matlab.ui.control.NumericEditField
    Modo              matlab.ui.container.ButtonGroup
    ContinuoButton    matlab.ui.control.RadioButton
  end
end
```

```
        RepeticinButton          matlab.ui.control.RadioButton
    end
    methods (Access = private)
        % Button pushed function: StartButton
        function StartButtonPushed(app, event)
    close all;

    instrreset;

    %%ejercicio='clean';

    vectorx=[];
    vectory=[];
    vectorz=[];

    duracion=app.Duracion.Value;
    tramas=duracion/0.1;

    s = serial('COM8', 'Terminator',10); %% Creación de objeto de puerto
serie para la comunicacion por USB con Arduino Uno

    fopen(s); % initiate arduino communication

    for counter =1:tramas

        caracter=fscanf(s); %%Lee los caracteres del puerto serial hasta el
delimitador, son almacenados en una celda
        C = strsplit(caracter, ' '); %Los distribuye en celdas suprimiendo el
espacio en blanco
        [vectorx, vectory, vectorz]= xyzDataExtract(C, counter, vectorx,
vectory, vectorz);

        plot(app.UIAxes,1:size(vectorx,2),vectorx, 'r');

        plot(app.UIAxes,1:size(vectory,2),vectory, 'k');

        plot(app.UIAxes,1:1:size(vectorz,2),vectorz, 'g');

        legend(app.UIAxes, 'Eje x', 'Eje y', 'Eje z');

        drawnow %% Permite que el objeto UIAxis este continuamente
dibujandose con cada nuevo dato

    end

    peso=app.Peso.Value;
```

```
ejercicio=app.Ejercicio.Value;
continuo=app.ContinuoButton;
if continuo==true
modo='Continuo';
else
modo='Repeticion';
end

%% Guardar variables de datos sin sobreescribir las anteriores.
load('numeroMuestra')
numeroString= num2str(numeroMuestra);

nombreMuestra= strcat(ejercicio,modo,numeroString);
nombre=strcat(nombreMuestra, '.mat');

numeroMuestra= numeroMuestra+1;
save numeroMuestra.mat numeroMuestra;

save(nombre , 'vectorx', 'vectory', 'vectorz', 'ejercicio', 'peso',
'modo' );

%% Cierre de conexion

fclose(s);
delete(s);

instrreset; %%Libero el puerto por si quiero usarlo en Arduino IDE

end
% Value changed function: Ejercicio
function EjercicioValueChanged(app, event)
value = app.Ejercicio.Value;

end
% Callback function
function PesoValueChanged(app, event)
value = app.Peso.Value;

end
% Callback function
function StopButtonPushed(app, event)
start=false;

end
end
% App initialization and construction
methods (Access = private)
% Create UIFigure and components
function createComponents(app)
% Create UIFigure
app.UIFigure = uifigure;
app.UIFigure.Position = [100 100 640 480];
```

```
app.UIFigure.Name = 'UI Figure';
% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
title(app.UIAxes, 'Aceleraciones')
xlabel(app.UIAxes, 'Muestras')
ylabel(app.UIAxes, 'Aceleracion (g)')
app.UIAxes.PlotBoxAspectRatio = [1 0.599659284497445
0.599659284497445];
app.UIAxes.YLim = [-3 3];
app.UIAxes.ColorOrder = [0 0.4471 0.7412;0.851 0.3255
0.098;0.9294 0.6941 0.1255;0.4941 0.1843 0.5569;0.4667 0.6745
0.1882;0.302 0.749 0.9294;0.6353 0.0784 0.1843];
app.UIAxes.NextPlot = 'add';
app.UIAxes.Position = [66 41 512 337];
% Create StartButton
app.StartButton = uibutton(app.UIFigure, 'push');
app.StartButton.ButtonPushedFcn = createCallbackFcn(app,
@StartButtonPushed, true);
app.StartButton.Position = [509 28 100 22];
app.StartButton.Text = {'Start'; ''};
% Create EjercicioDropDownLabel
app.EjercicioDropDownLabel = uilabel(app.UIFigure);
app.EjercicioDropDownLabel.HorizontalAlignment = 'right';
app.EjercicioDropDownLabel.Position = [84 427 48 22];
app.EjercicioDropDownLabel.Text = 'Ejercicio';
% Create Ejercicio
app.Ejercicio = uidropdown(app.UIFigure);
app.Ejercicio.Items = {'Clean', 'Clean_and _jerk', 'Snatch',
'Kettlebell_swing', ''};
app.Ejercicio.ValueChangedFcn = createCallbackFcn(app,
@EjercicioValueChanged, true);
app.Ejercicio.Position = [147 427 100 22];
app.Ejercicio.Value = 'Clean';
% Create PesoKgEditFieldLabel
app.PesoKgEditFieldLabel = uilabel(app.UIFigure);
app.PesoKgEditFieldLabel.HorizontalAlignment = 'right';
app.PesoKgEditFieldLabel.Position = [73 396 59 22];
app.PesoKgEditFieldLabel.Text = 'Peso( Kg)';
% Create Peso
app.Peso = uieditfield(app.UIFigure, 'numeric');
app.Peso.Position = [147 396 100 22];
% Create DuracinaproxLabel
app.DuracinaproxLabel = uilabel(app.UIFigure);
app.DuracinaproxLabel.HorizontalAlignment = 'center';
app.DuracinaproxLabel.Position = [273 407 94 27];
app.DuracinaproxLabel.Text = {'Duración aprox '; '(s)'};
% Create Duracion
app.Duracion = uieditfield(app.UIFigure, 'numeric');
app.Duracion.Position = [361 412 100 22];
% Create Modo
app.Modo = uibuttongroup(app.UIFigure);
app.Modo.Title = 'Modo';
```

```
app.Modo.Position = [486 384 123 72];
% Create ContinuoButton
app.ContinuoButton = uiradiobutton(app.Modo);
app.ContinuoButton.Text = 'Continuo';
app.ContinuoButton.Position = [11 26 70 22];
app.ContinuoButton.Value = true;
% Create RepeticinButton
app.RepeticinButton = uiradiobutton(app.Modo);
app.RepeticinButton.Text = {'1 Repetición'; ''};
app.RepeticinButton.Position = [11 4 89 22];
end
end
methods (Access = public)
% Construct app
function app = InterfazGrafica
% Create and configure components
createComponents(app)
% Register the app with App Designer
registerApp(app, app.UIFigure)
if nargin == 0
clear app
end
end
% Code that executes before app deletion
function delete(app)
% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end
```

20.4 Función para leer los valores de la trama (xyzDataExtract.m)

```
function [vectorx, vectory, vectorz]=xyzDataExtract(C, counter,
vectorx, vectory, vectorz )

%% Es importante tener en cuenta que solo funcionara si el tipo de
respuesta API está configurada como nativa en XCTU

%% Coordenada X negativa

if char(hex2dec(C(17)))=='-'
vectorx(counter)=str2double(char(hex2dec(C(17:21)))); %% Si
es el valor es negativo

if char(hex2dec(C(23)))=='-'
vectory(counter)=str2double(char(hex2dec(C(23:27))));
%% Si el valor es negativo

if char(hex2dec(C(29)))=='-'

vectorz(counter)=str2double(char(hex2dec(C(29:33))));
```

```
        else

vectorz(counter)=str2double(char(hex2dec(C(29:32))'));
        end

        else
        vectory(counter)=str2double(char(hex2dec(C(23:26))'));
%% So el valor es positivo

                if char(hex2dec(C(28)))=='-'

vectorz(counter)=str2double(char(hex2dec(C(28:32))'));
                else

vectorz(counter)=str2double(char(hex2dec(C(28:31))'));
                end

        end

        %% Coordenada x positiva
        else
        vectorx(counter)=str2double(char(hex2dec(C(17:20))'));    %% Si
el valor es positivo

                if char(hex2dec(C(22)))=='-'
        vectory(counter)=str2double(char(hex2dec(C(22:26))'));
%% Si el valor es negativo

                if char(hex2dec(C(28)))=='-'

vectorz(counter)=str2double(char(hex2dec(C(28:32))'));
                else

vectorz(counter)=str2double(char(hex2dec(C(28:31))'));
                end

        else
        vectory(counter)=str2double(char(hex2dec(C(22:25))'));
%% So el valor es positivo

                if char(hex2dec(C(27)))=='-'

vectorz(counter)=str2double(char(hex2dec(C(27:31))'));
                else

vectorz(counter)=str2double(char(hex2dec(C(27:30))'));
                end

        end

end

end
```