



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica

Ingeniería Mecatrónica

Tesis presentada para obtener el título de:  
Licenciado en Ingeniería Mecatrónica

**Desarrollo de Pulsómetro para Monitoreo de Ritmo  
Cardíaco a Larga Distancia**

Presenta:

Rodrigo Daniel Velasco García

Asesor:

Dr. Aldrin Barreto Flores  
Facultad de Ciencias de la Electrónica

Asesor Externo:

Dr. Jacinto Carvente Rodríguez  
Facultad de Cultura Física

Septiembre 2014



*Con todo mi cariño y mi amor para las personas que hicieron todo en la vida para que yo pudiera lograr mis sueños, por motivarme y darme la mano cuando sentía que el camino se terminaba, a ustedes por siempre mi corazón y mi agradecimiento.*



# Índice general

Introducción . . . . .	1
<b>1. Marco Teórico</b>	<b>5</b>
1.1. Pulsómetros en el deporte . . . . .	5
1.1.1. Historia del Pulsómetro en el deporte . . . . .	5
1.1.2. ¿Qué es un Pulsómetro? . . . . .	6
1.1.3. Tipos de Pulsómetros en el deporte . . . . .	6
1.1.4. Beneficios del uso del Pulsómetro . . . . .	6
1.2. Entrenamiento Inteligente . . . . .	7
1.2.1. Zonas de esfuerzo físico . . . . .	7
1.3. Protocolo de comunicaciones ZigBee . . . . .	9
1.3.1. Módulo XBee . . . . .	9
1.3.1.1. Modos de operación . . . . .	10
1.3.1.2. Modos de configuración . . . . .	11
1.3.1.3. Tipos de antenas . . . . .	12
1.4. Raspberry Pi . . . . .	14
1.4.1. Historia de la Raspberry Pi . . . . .	14
1.4.2. Hardware . . . . .	15
1.4.3. Software . . . . .	16
1.4.3.1. Sistema Operativo . . . . .	16
1.4.3.2. Python . . . . .	17
1.4.3.3. TkInter . . . . .	17
<b>2. Desarrollo</b>	<b>19</b>
2.1. Medición del ritmo cardiaco . . . . .	20
2.1.1. Sensor Polar T34 . . . . .	21
2.1.2. Receptor Polar . . . . .	21
2.2. Transmisión de datos a larga distancia . . . . .	22
2.2.1. Módulos XBee . . . . .	23
2.2.1.1. Configuración Cable Virtual . . . . .	23
2.2.1.2. Configuración de los módulos . . . . .	24
2.3. Sistema de Monitoreo . . . . .	27

2.3.1.	Raspberry Pi . . . . .	28
2.3.1.1.	Configuración inicial . . . . .	28
2.3.1.2.	Elección de los periféricos . . . . .	30
2.3.2.	Obtención de la frecuencia cardiaca . . . . .	31
2.3.2.1.	Muestreo . . . . .	31
2.3.2.2.	Cálculo de la frecuencia cardiaca . . . . .	32
2.3.3.	Interfaz Gráfica con el Usuario (GUI) . . . . .	34
2.3.3.1.	GUI para la base de datos . . . . .	34
2.3.3.2.	GUI para el monitoreo de la frecuencia cardiaca	41
2.4.	Construcción final . . . . .	45
2.4.1.	Sistema para el atleta . . . . .	46
2.4.2.	Sistema para el entrenador . . . . .	48
<b>3.</b>	<b>Pruebas y Resultados</b>	<b>51</b>
3.1.	Pruebas con el sensor Polar . . . . .	51
3.2.	Resultados de las pruebas con el sensor Polar . . . . .	52
3.3.	Módulos XBee . . . . .	53
3.3.1.	Pruebas de transmisión de datos . . . . .	53
3.3.2.	Resultados de transmisión de datos . . . . .	53
3.3.3.	Pruebas de distancia de transmisión . . . . .	54
3.3.4.	Resultados de la distancia de transmisión con la serie Pro . . . . .	55
3.4.	Pruebas para el cálculo de la frecuencia cardiaca . . . . .	56
3.5.	Resultados del cálculo de la frecuencia cardiaca . . . . .	57
3.6.	Pruebas finales del sistema . . . . .	60
3.7.	Resultados finales del sistema . . . . .	62
3.8.	Comparación del sistema con otros pulsómetros . . . . .	63
<b>4.</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>67</b>
4.1.	Conclusiones . . . . .	67
4.2.	Trabajo futuro . . . . .	68
	<b>Bibliografía</b>	<b>69</b>
<b>A.</b>	<b>Características del Sistema</b>	<b>73</b>
<b>B.</b>	<b>Programa para calcular la mediana de los tiempos obtenidos</b>	<b>75</b>
<b>C.</b>	<b>Especificaciones de la Raspberry Pi</b>	<b>77</b>
<b>D.</b>	<b>Especificaciones de los módulos XBee</b>	<b>81</b>

# Introducción

En la actualidad existen varios tipos de pulsómetros o dispositivos que monitorean el ritmo cardiaco de una persona, estos dispositivos son usados en diversas situaciones y en áreas como la medicina, la rehabilitación y el deporte, siendo esta última el área de interés de esta tesis.

En la literatura sobre el deporte, artículos [1] [2] y libros [3] [4] se ha reportado que realizar el monitoreo del ritmo cardiaco durante el entrenamiento de un atleta ayuda a mejorar el desempeño, lo que da mejores resultados para los mismos. La frecuencia cardiaca es el mejor indicador de la intensidad del esfuerzo, así como del volumen de trabajo realizado en actividades físicas y deportivas que inciden directamente sobre el sistema cardiovascular[7]. Es así como el pulsómetro se convierte en una herramienta útil para deportistas puesto que permite medir la frecuencia cardiaca y así obtener información sobre el esfuerzo realizado durante el entrenamiento.

Los pulsómetros que se usan en el área del deporte se dividen en dos tipos principales, los que usan sensor de tacto y los que usan sensor de banda pectoral. Los pulsómetros que usan sensor de tacto, los que se encuentran en aparatos para hacer ejercicio, como las bandas sin fin, y en aparatos más especializados como los usados para hacer pruebas de esfuerzo o rehabilitación cardiaca a atletas profesionales. Este tipo de aparatos se usan en un ambiente controlado en donde el movimiento del atleta es restringido a una pequeña área y la comunicación del sensor al sistema de procesamiento de datos es alámbrica. La ventaja de estos dispositivos es su precisión y que además cuentan con sensores para medir otras variables como la presión arterial o frecuencia respiratoria. Su principal desventaja es que no pueden ser usados para medir estas variables en pleno entrenamiento del atleta al aire libre.

Dentro del área del deporte es el atletismo una de las áreas que más usa los pulsómetros móviles. En específico los corredores, ya sean velocistas o corredores de fondo los usan para realizar entrenamiento en pista, campo abierto, arena o cualquier otro tipo de superficie al aire libre que le permita al corredor variar la intensidad y el esfuerzo en el entrenamiento, lo que mejora su musculatura y estamina.

Los pulsómetros con sensor de banda pectoral se componen por dos elementos, un sensor colocado en una banda elástica que se coloca en el tórax, y un receptor en forma de reloj que permite visualizar las señales obtenidas por el sensor. La comunicación entre estos dos dispositivos se realiza de manera inalámbrica lo que otorga al atleta la movilidad necesaria para realizar el monitoreo mientras corre o hace ejercicio, pero su comunicación es de poco alcance, alrededor de 40cm a 1m.

En esta área existen muchos tipos de pulsómetros, que varían de precios dependiendo de sus características, el cinturón puede ir desde los 25 hasta los 100 dólares y el reloj desde los 100 hasta los 400 dólares. Dependiendo del área del deporte al que esté dirigido el pulsómetro y lo avanzado de éste, permiten ver datos adicionales a la frecuencia cardiaca como la distancia recorrida y la velocidad, algunos tienen integrado un GPS que permite ver el trayecto recorrido y otros permiten descargar los resultados en aplicaciones para smartphones. Todos estos pulsómetros son útiles para deportistas novatos o profesionales pero tienen una limitante en común, el monitoreo durante el entrenamiento solamente puede ser hecho por el propio atleta, esto representa una desventaja para los atletas profesionales o de alto rendimiento puesto que el entrenador debe ser el encargado de realizar el monitoreo del ritmo cardiaco para optimizar el entrenamiento del atleta.

Es precisamente de esta desventaja de donde nace la necesidad de la Facultad de Cultura Física de la BUAP por crear un pulsómetro, que le permita al profesor de atletismo de esta facultad monitorear y almacenar los datos del ritmo cardiaco de uno de sus alumnos durante el entrenamiento, de manera inalámbrica y a una distancia radial que tenga un alcance de hasta 100 metros, lo que le permitirá al profesor realizar un esquema de entrenamiento específico para cada uno de sus alumnos y de esta manera obtener una mejora en el rendimiento de ellos.

El artículo [16] y el artículo [17] presentan trabajos relacionados con el desarrollo o mejora de pulsómetros para deportistas. Estos trabajos tienen el mismo concepto que el presentado en esta tesis, el de aumentar la distancia del monitoreo del ritmo cardiaco para este se haga por un tercero. Hay dos principales diferencias en el pulsómetro que se va a desarrollar en esta tesis a los que se desarrollaron en estos artículos. La primera es que se va a usar un sensor que es conocido en el área del deporte por tener buena calidad en su fabricación, por lo que puede ser usado en el entrenamiento del atleta, y buena precisión en la medición del ritmo cardiaco. La segunda es que se va a usar una computadora embebida para realzar el sistema de monitoreo lo que agrega mayor movilidad al sistema.

La presente tesis se conforma de cuatro capítulos. El primer capítulo presenta el marco teórico, se describe la importancia de realizar entrenamiento físico y los beneficios sobre realizarlo con la ayuda de pulsómetros, también se da la información necesaria sobre las herramientas que se decidió utilizar para desarrollar el sistema.

El desarrollo práctico del sistema se describe en el segundo capítulo, dividido en tres etapas: Medición del Ritmo Cardiaco, Transmisión de Datos a Larga Distancia y Sistema de Monitoreo.

En el tercer capítulo se describen las pruebas realizadas junto con los

resultados que se generaron, así como las mejoras que se decidieron hacer y los cambios implementados.

Las conclusiones que se obtuvieron en base al análisis de los resultados y al trabajo realizado se presentan en el capítulo cuatro además del trabajo futuro que se puede hacer sobre el sistema desarrollado.



# Capítulo 1

## Marco Teórico

### 1.1. Pulsómetros en el deporte

#### 1.1.1. Historia del Pulsómetro en el deporte

En 1975 el profesor de electrónica de la universidad de Oulu, Finlandia, Seppo Saynajakangas, diseño el primer sistema medidor de ritmo cardiaco inalámbrico [22], que estaba compuesto de una banda electromagnética, ajustada al toráx, que captaba las señales cardiacas y un receptor parecido a un reloj de muñeca que procesaba las señales recibidas del sensor.

El profesor inicio su investigación con la intención de desarrollar un pulsómetro que permitiera su uso en el deporte, ya que los aparatos como el Holter [30] y el ECG [31] que se tenían en ese tiempo, median la frecuencia cardiaca usando sensores foto reflectantes que se colocaban en la oreja o el dedo, y no eran adecuados para usarse en el deporte porque requerían detener el entrenamiento para realizar las mediciones.

Mediante la fundación de la compañía Polar electro OY en 1977, el profesor fue capaz de seguir con su investigación y mejora de este prototipo, hasta que en 1982 obtuvo la patente del primer Medidor de Ritmo Cardiac ECG de corta distancia serie PE 200S (MRC). En 1984 se lanzó al mercado el primer pulsómetro y tuvo un gran éxito en el mundo del deporte. Esto revolucionó la manera de realizar entrenamiento, dando paso a lo que hoy se conoce como “Entrenamiento Inteligente”. [6]

Gracias al éxito de este pulsómetro compañías como NIKE y Sanyo entre otras, sacaron al mercado sus propios pulsómetros, aunque no tuvieron tanto éxito pues la compañía Polar constantemente mejoraba la calidad y precisión de sus productos. Actualmente se pueden encontrar pulsómetros de diferentes marcas y accesibles para todo tipo de deportistas, desde el amateur hasta el profesional, pero Polar sigue siendo la marca líder en innovación y calidad.

### 1.1.2. ¿Qué es un Pulsómetro?

Un pulsómetro es un dispositivo que mediante electrodos detecta las señales generadas por los latidos del corazón y de esta manera indica los latidos del corazón que hay en un minuto, esto se conoce como la Frecuencia Cardiaca (FC). Los modelos de pulsómetros más avanzados pueden mostrar datos adicionales a la frecuencia cardiaca, como la frecuencia mínima y frecuencia máxima del atleta. El origen de los pulsómetros se dio en la medicina pues la primer aplicación de los mismos fue monitorear los latidos del corazón de los enfermos dentro de los hospitales.

En la década de los 70s gracias a la investigación del profesor Seppo Saajanakangas, se pudo reducir el tamaño de los pulsómetros y además realizar la transferencia de las señales capturadas por el sensor, hacia el sistema de procesamiento de manera inalámbrica lo que le dio mayor movilidad y flexibilidad a los pulsómetros, abriendo su campo de aplicaciones y en especial dando origen al pulsómetro deportivo.

### 1.1.3. Tipos de Pulsómetros en el deporte

En función del tipo de sensor que se ocupa para captar las señales del corazón se pueden diferenciar dos tipos de pulsómetros.

	<b>Pulsómetro con sensor de tacto</b>	<b>Pulsómetro con sensor de banda pectoral</b>
VENTAJAS	Mayor precisión	Permite monitoreo durante el entrenamiento Resistente al agua
DESVENTAJAS	Limita el movimiento Sensible a movimientos bruscos	Menor precisión Sensible a interferencia Usa batería

Tabla 1.1: Tipos de Pulsómetros

### 1.1.4. Beneficios del uso del Pulsómetro

El uso del pulsómetro en el deporte tiene beneficios para todos los tipos de usuarios.

A los principiantes les ayuda a controlar la intensidad del ejercicio que realizan ofreciendoles durante el entrenamiento los datos necesarios para ver la mejora de su condición física.

Para personas que realizan ejercicio con regularidad los ayuda a controlar la intensidad de acuerdo a su programa de entrenamiento y a conocer la cantidad de descanso que necesitan.

A los deportistas profesionales los ayuda a realizar diferentes sesiones de entrenamiento al nivel de intensidad planificado permitiéndoles seguir un programa de entrenamiento.

Y a los pacientes en recuperación les ayuda a realizar ejercicio de forma segura al nivel de intensidad recomendado por su médico.

## 1.2. Entrenamiento Inteligente

El entrenamiento inteligente [4] [5] [6] consiste en realizar ejercicio o actividad física de manera constante y controlada o supervisada por un entrenador profesional que sepa analizar la información obtenida a través de un pulsómetro y de esta manera realizar un plan de trabajo individual y especial para el atleta.

Cuando un atleta entrena sin supervisión es posible que se sobre entrene o que no entrene lo necesario para rendir al máximo. El entrenamiento inteligente evita que el atleta exceda sus límites y por el contrario ayuda a que lleve una preparación progresiva controlando los esfuerzos realizados, es decir, que el atleta realice su entrenamiento respetando las zonas de intensidad del esfuerzo y realizando cargas físicas dosificadas y seguras.

### 1.2.1. Zonas de esfuerzo físico

La frecuencia cardiaca es el mejor indicador del esfuerzo físico en una persona, es por esta razón que el pulsómetro tiene un papel importante en el entrenamiento de un atleta, ya que permite conocer la frecuencia cardiaca del atleta durante el entrenamiento y de esta manera saber si se encuentra trabajando con la intensidad necesaria para encontrarse en la zona de esfuerzo físico que requiere para su entrenamiento.

Para conocer las zonas de esfuerzo físico de un atleta se le debe determinar la frecuencia cardiaca máxima (FCmax), para esto se deben realizar pruebas como la propuesta por el Dr. Conconi llamada "Test de Conconi".[18] [19] Aunque también se puede determinar de una manera más sencilla con la siguiente fórmula[4] [5]:

$$\mathbf{FCmax} = 220 - \text{Edad del atleta} \dots \dots \dots \text{ec. 1}$$

Aunque la anterior fórmula es válida, actualmente se considera obsoleta y en base a [3] se debe ocupar la siguiente para predecir la FCmax de jóvenes en carrera:

$$\mathbf{FCmax} = 208.75 - 0.73 * \text{Edad del atleta} \dots \text{ec. 2}$$

Una vez que sabemos la frecuencia máxima se puede determinar mediante la formula de Karvonen [13], el rango de frecuencia cardiaca que pertenece a cada porcentaje de esfuerzo físico:

$$\mathbf{FC} = (\text{FCmax} - \text{FC reposo}) * \% + \text{FC reposo} \dots \text{ec. 3}$$

El esfuerzo físico se divide en cuatro zonas de intensidad:

**Zona de baja intensidad** En esta zona se maneja un esfuerzo del 50 % al 70 % de la frecuencia cardiaca máxima.

También es una zona de recuperación y se mejora la condición del corazón.

**Zona de intensidad media** En esta zona se establece un porcentaje de esfuerzo del 70 % al 80 % de la frecuencia cardiaca máxima.

En esta zona se mejora la capacidad aeróbica y se obtienen ganancias cardiovasculares importantes, por lo que si se trabaja en esta zona se tendrá una mejora sustancial.

**Zona de intensidad alta** En esta zona el porcentaje de esfuerzo oscila entre el 80 % y el 90 % de la frecuencia cardiaca máxima.

En esta zona se encuentra el umbral anaeróbico, la cual es una zona recomendable sólo para deportistas de élite.

**Zona de esfuerzo máximo** El porcentaje de esfuerzo en esta zona va del 90 % al 100 % de la frecuencia cardiaca máxima.

Esta zona de trabajo es muy dolorosa y fuerte, sólo es apta para deportistas de élite que se encuentran en competencia.

Por el enorme esfuerzo realizado esta es la zona de máximo consumo de oxígeno y produce mucho ácido láctico, al entrenar esta zona se requiere de mucho descanso entre cada serie que se realiza.



Figura 1.1: Módulo XBee

### 1.3. Protocolo de comunicaciones ZigBee

La organización ZigBee Alliance [15] formada por más de 200 empresas como Mitsubishi, Honeywell, Philips entre otras, creó el protocolo de comunicaciones ZigBee que es un protocolo basado en el estándar de comunicaciones para redes inalámbricas IEEE 802.15.4, con la finalidad de facilitar la creación de redes de sensores en el área de la industria, medicina y domótica. Este protocolo permite que dispositivos de bajo consumo realicen transferencia de datos de manera inalámbrica.

Este protocolo también presenta una velocidad de transmisión de datos de 256kbps, realiza las comunicaciones a través de un solo canal y tiene 16 canales disponibles para elegir. Es especialmente útil para crear redes complejas de hasta 65535 dispositivos. La potencia de transmisión del dispositivo y la antena utilizada determinan el alcance de la comunicación.

El protocolo ZigBee tiene la flexibilidad de realizar desde comunicación serial simple hasta configuraciones complejas como punto a punto o multi-punto.

Estas configuraciones están formadas por 3 tipos de dispositivos:

**Coordinador** Establece el canal de comunicaciones y el identificador de red para después formar la red. También es el dispositivo origen o destinatario de la información.

**Router** Determina la mejor ruta para retransmitir los datos de otros routers o end points.

**End Device** Es el punto final de la red y sólo recibe o transmite información.

#### 1.3.1. Módulo XBee

Los módulos XBee son dispositivos que usan el protocolo de comunicaciones ZigBee y gracias a esto cuentan con las siguientes ventajas:

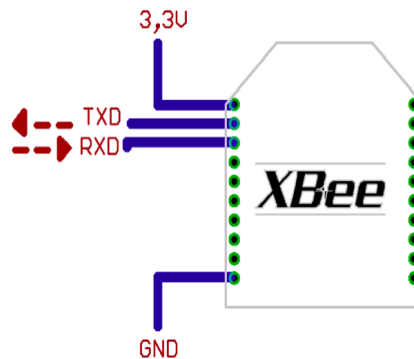


Figura 1.2: Conexión básica XBee

- Bajo costo.
- Ultra bajo consumo de potencia.
- Uso de bandas de radio libres y sin necesidad de licencias.
- Instalación barata y simple.
- Redes flexibles y extensibles.

Los módulos XBee se dividen en dos dependiendo de su alcance de transmisión: los XBee que tienen un alcance máximo de 100 mts y los XBee Pro con un alcance de hasta 1600 mts, ambos en condiciones óptimas. Además se tiene la serie 1 y la serie 2, siendo la diferencia entre éstas la complejidad de las redes que se pueden crear, siendo la serie 2 la más avanzada.

En la figura 1.2 se pueden ver las conexiones mínimas para hacer funcionar el módulo que son: alimentación que puede ir de 2.8 a 3.4 V, y las líneas de transmisión de datos.

#### 1.3.1.1. Modos de operación

Los módulos XBee tienen seis modos de operación:

**Modo Recibir/Transmitir** En este modo el módulo se encuentra transmitiendo o recibiendo datos ya sea por la antena o alguno de sus pines.

**Modo de Bajo Consumo** El módulo entra al modo de bajo consumo de energía cuando está en reposo, es decir, que el módulo no se encuentra trabajando. También se puede activar este estado por medio del pin 9.

**Modo de Comando** Este es el modo en el que se pueden configurar los parámetros del módulo XBee por medio de comandos AT que se ingresan a través del puerto UART. Para configurar el módulo desde la computadora se necesita un programa que emule la comunicación serial por el puerto USB. Para ingresar a este modo se debe ingresar el comando +++ y esperar a que el módulo devuelva OK, después de esto ya se pueden ingresar los comandos AT.

**Modo Transparente** En este modo todo lo que ingresa por el pin *Data in* se transmite como paquete RF y todo lo que llega como paquete RF es enviado por el pin *Data out*. Es el modo más sencillo de operación y viene por defecto en los módulos XBee.

**Modo de operación API** Este modo es similar al modo Transparente pero en éste los datos son colocados en *frames* que indican operaciones y eventos dentro del módulo.

**Idle** Si el módulo no está transmitiendo, recibiendo, ahorrando energía o en el modo de comandos, entonces se dice que se encuentra en un estado IDLE.

### 1.3.1.2. Modos de configuración

Los modos de configuración son los tipos de redes que soporta el protocolo de comunicaciones inalámbricas Zigbee y por tanto los módulos XBee.

**Modo de Conexión Transparente** Ésta es la configuración más sencilla, básicamente todos los datos que llegan al puerto UART son transmitidos al módulo que se desee y todos los datos que lleguen al módulo se pasan al puerto UART. Dependiendo del número de puntos de acceso y su interacción podemos diferenciar cuatro tipos de conexión transparente:

**Punto a Punto** Este tipo de configuración reemplaza la comunicación serial por cable a inalámbrica, lo único que se debe hacer es darle una dirección al módulo y la dirección del módulo al que se debe conectar.

**Punto a Multipunto** Este tipo de configuración permite enviar un paquete RF de un módulo a dos o más módulos, y recibir en un módulo paquetes RF de dos o más módulos.

**Broadcast** Para enviar datos de un módulo a varios módulos en una misma red se usa esta configuración.

**Cable Virtual IO** Esta configuración permite crear una comunicación del pin de un módulo al pin correspondiente de otro módulo.

**Modo de Conexión NonBeacon Peer to Peer** Este modo de configuración permite la conexión de todos los módulos de la red con todos los módulos, haciendo que éstos puedan actuar como maestros o esclavos.

**Modo de Conexión NonBeacon con Coordinador** Es una red punto a multipunto pero en esta configuración existe un módulo central que funciona como administrador de la red.

**Modo de Conexión API** Este modo de configuración agrega *frames* a los paquetes RF, que contienen información para hacer más robusto el envío de datos.

### 1.3.1.3. Tipos de antenas

Los módulos XBee manejan cuatro tipos de antena:

- Antena de chip
- Antena de PCB
- Antena de cable
- Antena de conector

Para elegir la antena adecuada se deben comparar las características de éstas. En la tabla 1.2 se puede apreciar la comparación de las antenas y en las figuras 1.3 y 1.4 se pueden ver los radios de alcance de éstas.[18]

Módulo	Tipo de Antena	Distancia al aire libre	Distancia en lugares cerrados
XBee	Chip	143m (470ft)	24m (80ft)
XBee	Cable	258m (845ft)	24m (80ft)
XBee Pro	Chip	151m (1690ft)	43m (140ft)
XBee Pro	Cable	1335m (4382ft)	43m (140ft)

Tabla 1.2: Tipos de Antenas

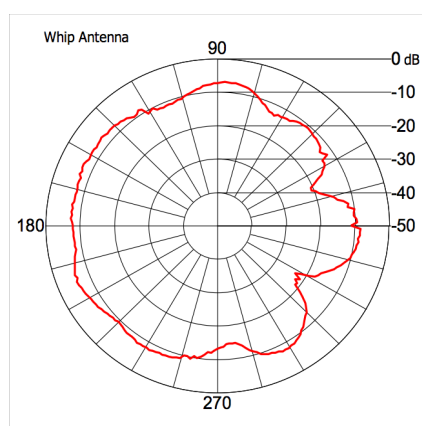


Figura 1.3: Radio de Alcance de la Antena de Cable

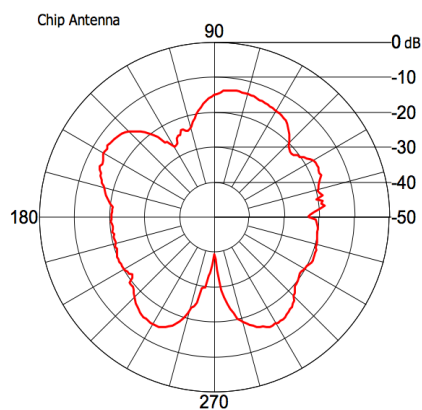


Figura 1.4: Radio de Alcance de la Antena de Chip

En la generación de la tabla se tuvieron en cuenta los siguientes aspectos:

- La antena de PCB tiene un alcance 5 % menor al de la antena de cable.
- La antena de conector tiene un comportamiento casi igual al de la antena de cable.

Al comparar las antenas podemos llegar a las siguientes conclusiones:

- La antena de cable tiene mayor alcance que la de chip en espacios abiertos, pero se comportan de manera similar en espacios cerrados.
- Los módulos XBee Pro tienen mayor alcance.
- Los módulos XBee y XBee Pro tienen mayor alcance en espacios abiertos.
- La distancia de alcance de la antena de cable es constante e independiente a la posición del módulo XBee.

## 1.4. Raspberry Pi

### 1.4.1. Historia de la Raspberry Pi

En 2006 los investigadores del Laboratorio de Computación de la Universidad de Cambridge: Eben Upton, Rob Mullins, Jack Lang y Alan Mycroft crearon una fundación y empezaron a trabajar en la creación de una computadora pequeña y de bajo costo con la idea de mejorar el nivel de educación de los niños en el Reino Unido. Su propuesta surgió en base a que observaron que los jóvenes cada vez saben menos sobre el funcionamiento de las computadoras y sólo se dedican a consumir.[22]

Del 2006 al 2008 crearon varios prototipos de esta mini PC, pero fue hasta el 2008 cuando los precios de los procesadores para dispositivos móviles con suficiente poder multimedia empezaron a bajar de precio. Esto hizo posible la creación de esta mini PC.

En el 2011 lograron crear el prototipo de la Raspberry Pi y en ese mismo año la fundación firmó un contrato para que las compañías Element 14/Premier Farnell y RS Electronics empezaran a manufacturar la tarjeta. El 29 de febrero del 2012 se lanzaron al mercado las primeras unidades de la Raspberry Pi agotándose a unas horas de su lanzamiento pero desde ese momento

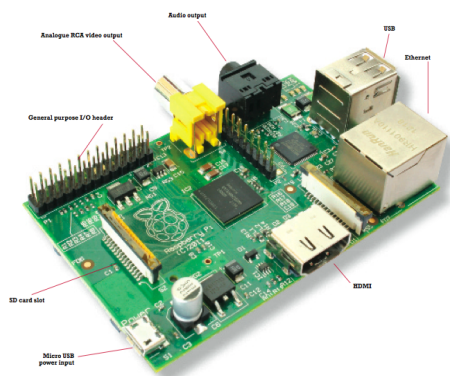


Figura 1.5: Raspberry Pi

se empezaron a aceptar órdenes de compra para la tarjeta que se surtieron a finales de ese mismo año. A mediados del 2013 la Raspberry Pi alcanzó el millón de unidades vendidas y para finales del mismo año alcanzó los dos millones.

### 1.4.2. Hardware

La Raspberry Pi es una computadora del tamaño de una tarjeta de crédito que funciona a base de un procesador central (CPU) ARM1176JZF-S con velocidad de 700MHz, que puede llegar hasta 1GHz utilizando *overclock*, el iPhone 3 ocupaba éste procesador, que se encuentra dentro de un chip Broadcom BCM2835, también cuenta con un procesador gráfico (GPU) VideoCoreIV y 512 MB de memoria RAM en el modelo B y 256 MB en el modelo A. En el apéndice D se pueden ver a detalle las características de la Raspberry Pi.

El modelo A tiene la mitad de memoria RAM, un solo puerto USB y no tiene controlador Ethernet por lo que cuesta menos que el modelo B que tiene controlador Ethernet y dos puertos USB.

La tarjeta cuenta con un puerto micro USB que sólo sirve para alimentación, también tiene un puerto para conectar una tarjeta SD que sirve como disco duro y es en donde se instala el sistema operativo. La tarjeta cuenta con salida de audio y video HDMI y salidas de audio y video analógicas para poder conectar la Raspberry Pi a cualquier televisión. También cuenta con pines GPIO o pines de entrada/salida general que funcionan a 3.3V, además de dos puertos SPI a los que se pueden conectar una cámara y un *display*.

### 1.4.3. Software

#### 1.4.3.1. Sistema Operativo

La tarjeta Raspberry Pi usa sistemas operativos(SO) basados en Linux, lo que los hace de licencia libre reduciendo el costo de usar la tarjeta. El sistema más usado por su flexibilidad y confiabilidad es Raspbian, que es la versión de la distribución Debian optimizada para la Raspberry Pi. Además de Raspbian hay versiones de otros distribuidores de linux para la Raspberry Pi como Pidora, Arch linux ARM, RISC OS. Una lista de los sistemas operativos que se pueden usar en la Raspberry Pi son los siguientes:

- Android
- Arch Linux ARM
- Debian Whezzy Soft-Float
- Firefox OS
- Gentoo Linux
- Google Chromium OS
- Kali Linux
- Open webOS
- PiBang Linux
- Pidora
- QtonPi
- Raspbian
- Slackware ARM
- Plan 9
- RISC OS
- OpenELEC

- Raspbmc
- Xbian

#### 1.4.3.2. Python

Python es un lenguaje de programación muy dinámico que te permite trabajar de manera rápida aplicaciones de diversos tipos. Debido al compilador de byte optimizado y al soporte de librerías, los códigos en Python se ejecutan mucho más rápido.

Python está disponible para la mayoría de los sistemas operativos, como Windows, Mac y Linux. Raspbian para la Raspberry Pi tiene instalado Python como el lenguaje de programación principal.

Python tiene licencia Open Source lo que hace que su uso y distribución sean gratuitos, incluso para fines comerciales.

Las principales características de Python son:

- Muy claro y limpio en su sintaxis.
- Habilidades de introspección.
- Orientación a objetos intuitiva.
- Expresiones naturales para código.
- Soporte de paquetes jerárquicos.
- Tipos de datos dinámicos.
- Gran cantidad de librerías y módulos para casi cualquier aplicación.
- Facilidad de crear aplicaciones embebidas.

#### 1.4.3.3. TkInter

TkInter es la librería estándar y más usada de Python para usar el kit de herramientas Tk para crear GUIs o Graphical User Interfaces. Es una librería basada en la orientación a objetos que permite realizar GUIs de manera rápida y sencilla. TkInter está disponible para usarse en los sistemas

operativos más importantes, como Linux, Mac y Windows. La interfaz de TkInter usa varios módulos de Python y para ser usada en Python solamente se debe de importar la librería al principio del código realizado.

# Capítulo 2

## Desarrollo

El desarrollo se dividió en tres etapas:

**Medición del Ritmo Cardíaco**

**Transmisión de Datos a Larga Distancia**

**Sistema de Monitoreo**

Las primeras dos etapas forman parte del sistema que va a portar el atleta. Parte de la segunda etapa y la última etapa comprenden el sistema que va a tener el entrenador. En la figura 2.1 se puede ver la división de las tres etapas.

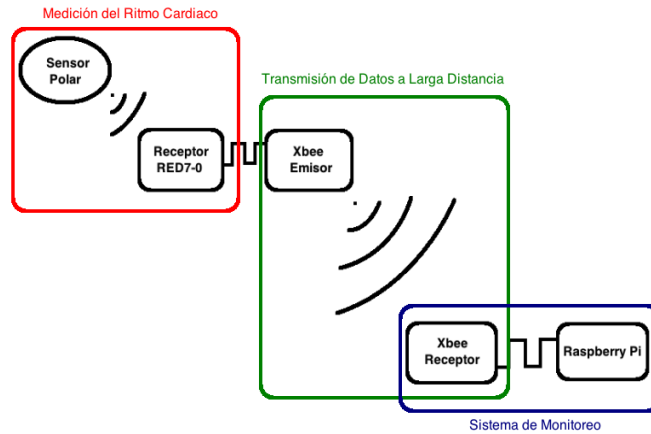


Figura 2.1: Diagrama de Bloques de las Tres Etapas de Desarrollo

## 2.1. Medición del ritmo cardíaco

En esta etapa se va a desarrollar el sistema que va a captar los pulsos del corazón. El sistema se quiere usar en el área del deporte, por lo cual se debe asegurar que el sensor sea cómodo y que no interfiera con el entrenamiento del atleta y que también sea de buena calidad y tenga una buena precisión. También debe de ser de bajo consumo ya que el sistema que va a portar el atleta se va a alimentar mediante una batería y además debe permitir adquirir los datos de manera sencilla para después pasarlos al transmisor de larga distancia.

Por estas razones se decidió realizar el monitoreo de los pulsos cardiacos usando el sensor de banda pectoral T34 de la marca Polar. Para recibir las señales del transmisor T34 se va a usar el chip RED7-0 que nos permite captar la señal no decodificada del sensor hasta a 1 m de distancia. Ambos trabajan a una frecuencia de 5kHz a 5.3kHz. En la figura 2.2 se puede apreciar el esquema de esta etapa.

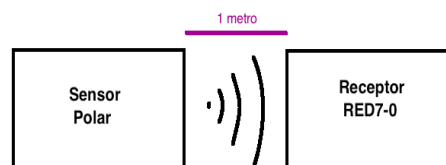


Figura 2.2: Diagrama de la Medición de Ritmo Cardíaco

### 2.1.1. Sensor Polar T34

Polar es la marca de pulsómetros deportivos con más antigüedad y es la más reconocida por su calidad e innovación, constantemente mejora sus productos para satisfacer las necesidades de sus clientes. Cuenta con una amplia gama de pulsómetros, bandas pectorales y relojes de monitoreo, que tienen diferentes características dependiendo el área del deporte al que están enfocados. Dentro de los productos que ofrece, se encuentra un sensor de banda pectoral que emite las señales sin decodificar, lo que permite captarlas ocupando un receptor compatible con el protocolo de comunicación que ocupa la marca. El sensor T34 es delgado, ligero y a prueba de agua, además tiene gran precisión y transmite las señales sin decodificarlas permitiendo que sean captadas por un receptor compatible con el transmisor, esto lo hace ideal para su uso en el desarrollo de esta etapa.



Figura 2.3: Sensor de Banda Elástica T34

### 2.1.2. Receptor Polar

Hay dos circuitos integrados que son capaces de recibir las señales no decodificadas transmitidas por el sensor Polar T34, el circuito integrado RED7-0 y el RMCM01. Ambos reciben las señales desde 40 cm hasta 1 m y por medio de un pin de salida comunican la señal de manera serial, indicando con un “1” cada latido del corazón. El chip RMCM01 también es capaz de recibir y decodificar la señal de otros sensores de la marca Polar y puede realizar comunicación UART con algún otro dispositivo como un microcontrolador por ejemplo.

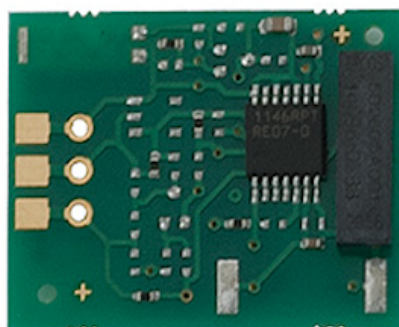


Figura 2.4: Receptor RED7-0

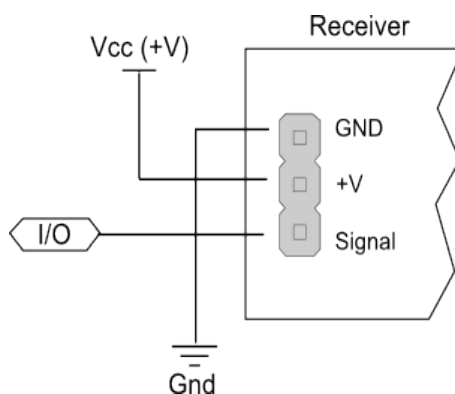


Figura 2.5: Diagrama de Conexión del Receptor RED7-0

El chip RED7-0 tiene un pin de salida TTL y es el único que por el momento se puede encontrar en el mercado de manera individual, ya que Polar no permite la venta del chip RMCM01 de manera individual, solamente puede ser vendido incluido en una tarjeta de desarrollo con toda una interfaz. Este chip se acopla a las necesidades y requerimientos del sistema a desarrollar ya que se alimenta de 3.3 a 5 V, este último es el voltaje de alimentación de los módulos XBee, y también es de bajo consumo con sólo 200  $\mu\text{A}$  a 5 V, lo que lo hace ideal para un sistema portátil.

## 2.2. Transmisión de datos a larga distancia

En la figura 2.6 se puede apreciar el diagrama de esta etapa. Para la transmisión de los datos a larga distancia se propuso usar el módulo XBee, que usa el protocolo de comunicaciones Zigbee. Esto permitió al dispositivo

trasmitir datos a largas distancias y con bajo consumo, lo que es un requisito importante ya que el sistema debe ser portátil. Esto permite tener un maestro y varios esclavos lo que le da oportunidad de crecimiento a este sistema en un futuro, ya que se podrían incorporar más transmisores, es decir hacer el monitoreo de dos o más atletas con un solo sistema de monitoreo.

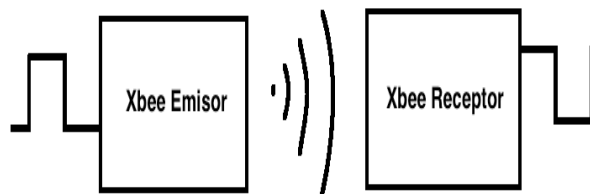


Figura 2.6: Diagrama de Transmisión de Datos a Larga Distancia

### 2.2.1. Módulos XBee

El módulo XBee Serie 1 permite la transmisión de datos a una distancia de hasta 100 mts (300 ft) en espacios abiertos o en óptimas condiciones, esta es la razón que se usó para seleccionar este módulo puesto que la distancia radial máxima que se quiere alcanzar es de 100 mts.

Este módulo funciona con un voltaje de alimentación de 2.8 a 3.4 V y consume una corriente en modo de transmisión de 45 mA, lo que lo hace de bajo consumo e ideal para esta etapa del sistema ya que va a ser alimentado por una batería. Además permite realizar la comunicación con el receptor RED7-0 de manera sencilla y transmitir los datos a larga distancia de manera transparente usando la configuración de Cable Virtual.

Otra ventaja de estos dispositivos es que tienen diferentes tipos de antenas y que permiten escoger la mejor antena para el sistema. Si vemos la comparación que se hizo en la tabla 1.2 y analizamos las figuras 1.3 y 1.4, la antena que se acopla a los requerimientos del sistema es la antena de cable. Ésta tiene un radio de transmisión casi constante en todas direcciones y permite al módulo XBee estar en cualquier posición y seguir transmitiendo sin problemas.

#### 2.2.1.1. Configuración Cable Virtual

La configuración Cable Virtual del módulo XBee permite hacer transmisión de datos de un pin del módulo transmisor al pin correspondiente del módulo receptor de manera transparente, es decir, lo que llega al transmisor sale del receptor. Esto es conveniente para ahorrar energía en esta etapa ya

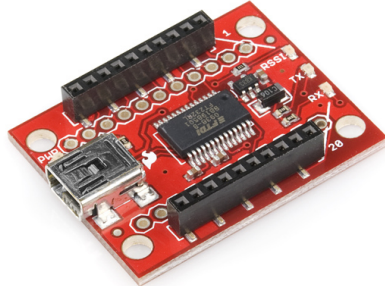


Figura 2.7: XBee Explorer

que no se tiene que hacer ningún tipo de procesamiento en el módulo transmisor ni se requiere de tener un  $\mu\text{C}$  extra en el sistema que va a usar el atleta, lo que mantiene el consumo bajo y el peso disminuye. Los módulos XBee tienen 10 pines que se pueden configurar como entradas y salidas digitales y algunos de estos pueden ser entradas analógicas o salidas PWM.

#### 2.2.1.2. Configuración de los módulos

Para realizar la configuración de los módulos se necesita un XBee Explorer USB. Este dispositivo permite pasar de comunicación USB a serial, esto es necesario para poder configurar los módulos XBee con la computadora, ya que los módulos XBee se configuran ingresando comandos a través del puerto serial.

Además del XBee Explorer se necesita un programa que te permita acceder a la terminal de la computadora para usar el puerto serial. X-CTU es el programa oficial para configurar los módulos XBee. Ésta es una aplicación para Windows que permite hacer una emulación de la terminal de la computadora para poder realizar comunicación serial con hardware. El desarrollo del sistema se realizó usando una computadora MAC por lo cual se buscó una aplicación que permitiera tener acceso a la terminal y realizar comunicación serial. CoolTerm es una aplicación *open source* que da acceso a la terminal y permite usar el puerto serial para realizar comunicación serial con hardware.

Lo primero que se debe de hacer es abrir una ventana de la aplicación (figura 2.8), después en el menú opciones se debe de elegir el puerto al que se conecto el XBee Explorer, el nombre del puerto debe empezar con `usbserial` como se ve en la figura 2.9. Una vez que se elige el puerto, se sale del

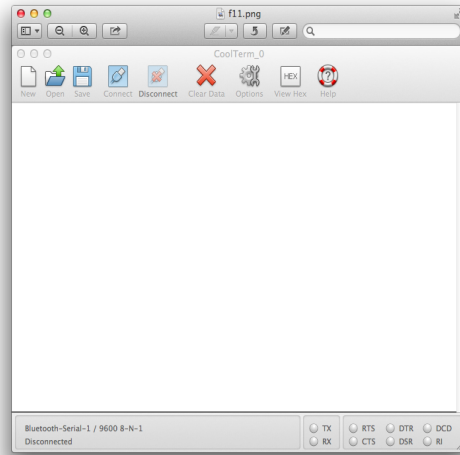


Figura 2.8: Ventana de CoolTerm

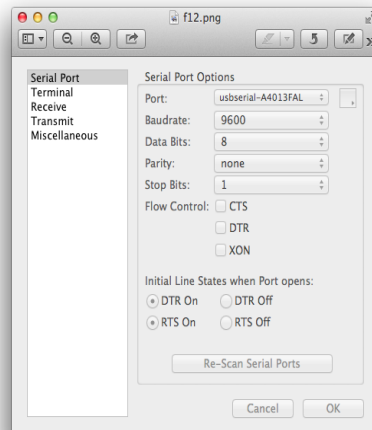


Figura 2.9: Ventana de Opciones de CoolTerm

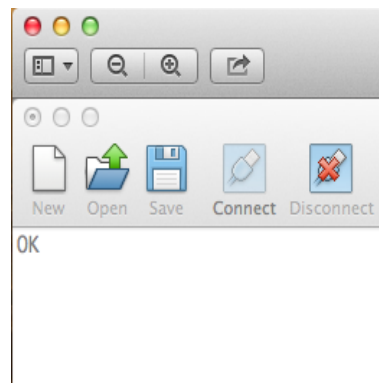


Figura 2.10: Inicio de Comunicación con XBee

menú opciones y se conecta al dispositivo por medio del botón conect. Posteriormente se debe iniciar la comunicación con el XBee, para esto debemos ingresar tres signos + seguidos(+++), sin presionar Enter y esperar a que el XBee responda con un OK como se ve en la figura 2.10

El XBee se configura mediante comandos AT que es una abreviación para *attention*, hay una gran variedad de comandos. Cada comando empieza con las letras AT, después se escriben dos caracteres que indican el comando que se va a usar y después los caracteres para configurar las opciones de este comando. A continuación se ven los más importantes:

**AT ID** Este comando se utiliza para darle un ID de red local al XBee.

**AT MY** Con este comando se le asigna una dirección de 16 bits al módulo.

**AT DH/DL** Este comando le indica al XBee con que módulo se debe comunicar, ya sea para recibir o enviar datos.

**AT WR** Con este comando se guarda la configuración hecha al módulo.

**AT CN** Con este comando se sale del modo de configuración.

Si solamente se escribe el comando sin los caracteres para configuración del comando, el módulo XBee responde con el valor que dicho comando tiene.

Aparte de los anteriores comandos, se necesitan algunos más para configurar los módulos XBee en modo Cable Virtual:

**AT DX** Configura el pin X como entrada digital/analógica ó salida digital.

**AT IR** Con este comando se establece la frecuencia de muestreo.

**AT IT** Indica la cantidad de muestras que se van a guardar en el buffer antes de enviarlas.

Entonces para configurar los módulos que va a usar este sistema, se ingresan los siguientes comandos:

Para el emisor:

```
+++  
ATID2171  
ATMY1  
ATDL2  
ATD43  
ATIRE  
ATIT1  
ATWR  
ATCN
```

Y para el receptor los siguientes.

```
+++  
ATID2171  
ATMY2  
ATDL1  
ATD44  
ATWR  
ATCN
```

Una vez que se configuran los módulos XBee con estos comandos, el módulo emisor va a transmitir la señal que tenga en la entrada digital 4 y el receptor va a mostrar la señal en la salida digital 4.

## 2.3. Sistema de Monitoreo

En esta etapa se desarrolló un sistema portátil y de fácil uso que permite al entrenador tener una base de datos de los atletas con los que está trabajando, ver el ritmo cardiaco del atleta mientras éste se encuentra entrenando y guardar los datos del ritmo cardiaco en un archivo de texto para después analizarlos.

En la implementación se usó la computadora Raspberry Pi, debido a que es una computadora de bajo consumo que permite una integración sencilla con dispositivos electrónicos y que además usa un sistema operativo basado en Linux por lo que su uso no tiene costo. Para crear la GUI se usó el lenguaje de programación Python y la librería TkInter que tienen licencia de uso libre.

### 2.3.1. Raspberry Pi

La tarjeta Raspberry Pi modelo B se alimenta a 5 V y consume de 700 mA a 1 A, cuenta con dos puertos USB, un puerto Ethernet, una salida de audio y video digital HDMI, salidas de audio y video analógicas, pines de entrada y salida digitales GPIO que funcionan a 3.3 V, que es el voltaje al que funciona el módulo XBee y un puerto para conectar una tarjeta SD en la que se instala el sistema operativo. Estas características hacen que la tarjeta Raspberry Pi fuese ideal para desarrollar el software del sistema.

#### 2.3.1.1. Configuración inicial

Los periféricos necesarios para hacer funcionar la Raspberry Pi son los siguientes:

- Cable micro USB-USB, que va a servir para conectar la tarjeta a alimentación.
- Cable HDMI o cable de video analógico para conectar la tarjeta a un monitor o televisión.
- Teclado y ratón USB.
- Tarjeta SD de mínimo 4Gb de 4ª generación.

Lo primero que se debe de hacer es instalar el sistema operativo en la tarjeta SD. La manera más sencilla de hacer esto es ingresar a la página oficial de la fundación [www.raspberrypi.org](http://www.raspberrypi.org), ingresar a la parte de descargas y descargar el archivo “New Out Of Box Software (NOOBS)” que te permite instalar cualquier sistema operativo oficial en la Raspberry Pi. Lo único que se debe de hacer es descomprimir el archivo .zip dentro de la tarjeta SD y después conectar los periféricos a la Raspberry Pi, algo importante a recordar es que lo último que se debe conectar es la alimentación.

En la figura 2.12 se ve la ventana que aparece al conectar la Raspberry Pi, en esta ventana se puede escoger el sistema operativo o sistemas a instalar en la Raspberry Pi. El sistema operativo que se va a usar es Raspbian, que es el más usado por su flexibilidad. Una vez que se instala el sistema operativo y encendemos



Figura 2.11: Periféricos para la Raspberry Pi

la Raspberry Pi aparecerá una ventana de configuración que se puede ver en la figura 2.13, en la que debemos de cambiar algunas opciones de acuerdo a los requerimientos que se tienen, como zona horaria e idioma. Una vez que se configuran las opciones se debe de reiniciar la tarjeta y entonces va a aparecer el escritorio de Raspbian.

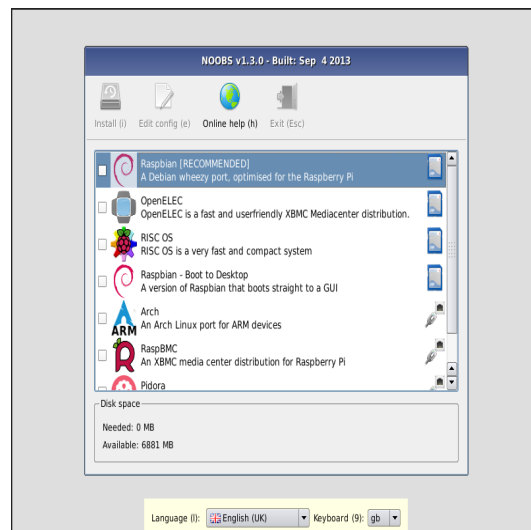


Figura 2.12: Ventana de Inicio NOOBS

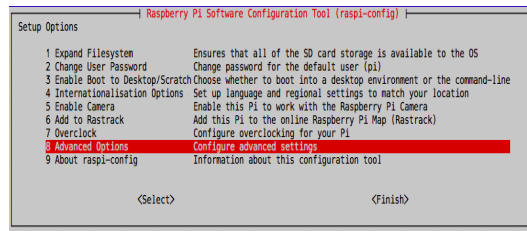


Figura 2.13: Ventana de Configuración Inicial Raspbian

### 2.3.1.2. Elección de los periféricos

El sistema de monitoreo se propuso hacer compacto y portátil pero el monitor y la fuente de alimentación son los periféricos que evitan que esto sea posible, esto requirió que se tenían que buscar los periféricos ideales para esta tarea.

Para el monitor se requiere tener una pantalla pequeña y que sea *touch* para que el entrenador pueda interactuar con las interfaces de los programas. En la fuente de alimentación se necesita una batería de 5 V y 1 A por medio de un puerto USB. La pantalla *touch* PiTFT es una pantalla de 2.8 pulgadas que se conecta a un puerto SPI y dos pines GPIO de la Raspberry Pi, esta pantalla ofrece las características que se necesitan para hacer al sistema de monitoreo portátil. La batería Anker Astro 3E ofrece 5 V y hasta 3 A por dos puertos USB y una capacidad de 10000 mAh lo que la hace ideal para alimentar el sistema de monitoreo.

Estos elementos representan la solución para una implementación completamente compacta y portátil, pero debido a que no pudieron ser adquiridos no fueron usados en el desarrollo del sistema, aunque éste está pensado para el uso de estos elementos.



Figura 2.14: Batería Anker Astro 3E

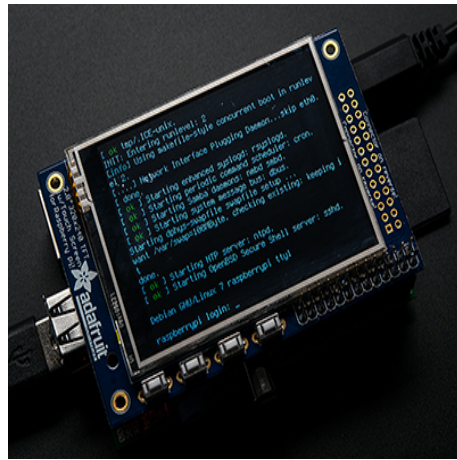


Figura 2.15: Pantalla touch PiTFT

### 2.3.2. Obtención de la frecuencia cardiaca

El receptor RED7-0 nos indica por medio de un “1” lógico cada vez que un pulso del corazón es detectado y esta misma señal es la que se va a obtener del módulo XBee receptor a través de la salida digital 4 que se encuentra en el pin 11 del dispositivo. Éste se va a conectar directamente a una de las entradas o salidas digitales de la Raspberry, por conveniencia se ha escogido el pin 6 del dispositivo o GPIO 18, y ahora lo que se debe de hacer es crear un programa que detecte cada vez que haya un “1” lógico en este GPIO. Para esto se decidió usar el lenguaje de programación Python pues es sencillo de usar y además permite usar la librería TkInter para crear la interfaz gráfica. El programa que se realizó lleva la cuenta de los pulsos recibidos y después de un tiempo indica cual es la frecuencia cardiaca.

#### 2.3.2.1. Muestreo

En la realización del muestreo se usaron dos técnicas: polling e interrupciones.

Utilizando la técnica polling se debe preguntar o comprobar el estado del pin cada cierto tiempo, en nuestro caso fué cada 0.001 segundos, este tiempo se eligió teniendo en cuenta que el tiempo de duración de la señal que entrega la salida del XBee es de 0.013 segundos.

La técnica por interrupciones no comprueba el estado de un pin, por el contrario detecta cada vez que hay un cambio en la señal ya sea de “0” a “1” o de “1” a “0”. En nuestro caso se detecta el cambio en la señal cuando pasa de “0” a “1”, el programa se interrumpe y atiende la acción que esta interrupción genere. Después de realizar una sencilla prueba en la que se ejecutaron por 10 minutos los dos programas y se compararon los resultados de ambos, se decidió por la técnica por interrupciones pues con ésta no se pierden datos y no consume tiempo

de procesamiento en espera. El siguiente es el programa para el manejo de las interrupciones:

```
def my_callback(channel): 1
    p = p + 1
    tiempo_actual = time.time()
    tiempo_entre_pulsos = tiempo_actual - tiempo_anterior
    tiempo_anterior = tiempo_actual
    tiempo_total = tiempo_total + tiempo_entre_pulsos

GPIO.add_event_detect(18, GPIO.RISING,
callback=my_callback) 2
```

### Código 2.1 Implementación por Interrupciones

En el programa anterior se registra el tiempo que transcurre entre cada interrupción y funciona de la siguiente manera:

- 1** Primero se define una función que es llamada cuando ocurre una interrupción. Dentro de esta función se aumenta una variable p para indicar la cantidad de pulsos que se han detectado y se registra el tiempo entre cada pulso y el tiempo total que es la suma de los tiempos entre pulsos.
- 2** En esta línea se define el pin que se va a ocupar para detectar las interrupciones y la función que va a ser llamada cuando una interrupción ocurra.

#### 2.3.2.2. Cálculo de la frecuencia cardiaca

Una vez que se adquieren los datos, el siguiente paso es calcular la frecuencia cardiaca, que son los pulsos que hay en un minuto.

Para calcularla se ocuparon dos métodos, el primero va a registrar los pulsos y aumentar una variable con cada pulso recibido y después de 1 min va a indicar el valor de esta variable, con esto se sabe el número exacto de pulsos que hubo en ese minuto. Pero ya que el sistema se va a usar en el área del deporte y más en específico en el atletismo, no es muy útil saber cual fue la cantidad de pulsos recibidos después de cada minuto.

Para que el sistema sea funcional en esta área se debe poder saber la frecuencia cardiaca en un tiempo menor a un minuto. La prueba más rápida en atletismo es la de 100 metros planos, de manera profesional el menor tiempo en que se realiza la prueba es de 9 segundos, por lo cual el sistema debe poder indicar la frecuencia cardiaca mínimo dos veces durante esta prueba, lo que le será útil al entrenador para saber la frecuencia cardiaca.

Teniendo en cuenta este requerimiento el programa tiene que ser capaz de predecir en al menos 4 segundos cual será la cantidad de pulsos que se van a recibir en un minuto, es decir predecir la frecuencia cardiaca.

Para hacer esta predicción el programa va a medir el tiempo que hay entre cada pulso y tomar una muestra de 5 pulsos para hacer un promedio de este tiempo y así asignar este valor a un rango de frecuencia cardiaca.

Una ventaja de esta propuesta es que la velocidad es directamente proporcional a la frecuencia cardiaca, es decir, mientras mayor frecuencia cardiaca por parte del deportista el valor se obtendrá más rápido. Con esto se asegura que en la prueba más rápida de atletismo se pueda ver la frecuencia cardiaca al menos tres veces durante la prueba.

Los valores mínimo y máximo de frecuencias que va a tener el sistema se dan de la siguiente manera. La frecuencia máxima de una persona está dada por la ecuación 2 que se vio en el capítulo 2.

Si tomamos en cuenta que el sistema lo van a ocupar atletas con una edad mínima de 17 años, entonces la frecuencia máxima a la que el sistema debe de asegurar llegar es de 197 pulsos en un minuto y una frecuencia mínima de 40 pulsos por minuto, ambas frecuencias que sólo atletas profesionales pueden alcanzar de manera segura.

La frecuencia mínima es de 40 pulsos por minuto, es decir que hay un tiempo de 1.5 segundos entre cada pulso y si se tiene una frecuencia de 41 pulsos por minuto, el tiempo entre cada pulso es de 1.463 segundos. Entonces si el promedio de la muestra de los 5 pulsos está entre estos dos valores se le va a asignar una frecuencia de 40 pulsos por minuto y así hasta llegar a la frecuencia de 209 pulsos por minuto.

La siguiente es una parte de la tabla de valores de frecuencia cardiaca.

```
while True:
    if p == 5:                                1
        p = 0
        tiempo_total = tiempo_total/5
        if 1.4634 < tiempo_total <= 1.5:      2
            valor = 40
        if 1.4286 < tiempo_total <= 1.4634:
            valor = 41
        if 1.3953 < tiempo_total <= 1.4286:
            valor = 42
        if 1.3636 < tiempo_total <= 1.3953:
            valor = 43
            .
            .
            .
        if 0.2899 < tiempo_total <= 0.2913:
```

```

        valor = 206
    if 0.2885 < tiempo_total <= 0.2899:
        valor = 207
    if 0.2871 < tiempo_total <= 0.2885:
        valor = 208
    if 0.2857 < tiempo_total <= 0.2871:
        valor = 209

```

### Código 2.2 Cálculo de la Frecuencia Cardíaca

El programa anterior funciona de la siguiente manera:

- 1 Cuando la variable `p` alcanza un valor igual a 5 se compara el tiempo total en una tabla.
- 2 Si el tiempo total se encuentra entre estos dos valores se le relaciona el valor correspondiente de frecuencia cardíaca, en caso contrario se pasa a la siguiente comparación hasta que se encuentre el valor al que corresponde.

### 2.3.3. Interfaz Gráfica con el Usuario (GUI)

En la interfaz gráfica con el usuario o GUI se debe poder ingresar y guardar los datos de los atletas a través de una base de datos. Estos datos se deben visualizar mientras se realiza el monitoreo, además de poder guardar los datos de la frecuencia cardíaca en un archivo de texto, deberá ser clara y sencilla de manejar para el usuario.

Ingresar los datos de los atletas requiere tener un teclado conectado a la Raspberry Pi y si se tiene en cuenta que el sistema se quiere hacer portátil, esto es una limitante. La solución para no tener que cargar con el teclado mientras se realiza el monitoreo es ingresar los datos de los atletas antes de realizar el entrenamiento, esto requiere tener una GUI para ingresar los datos de los atletas y otra GUI para realizar el monitoreo de los mismos.

Ambas GUIs se desarrollaron usando la librería TkInter para Python, esta librería es de licencia libre. La librería TkInter es la más usada en Python para crear interfaces gráficas pues es sencilla y fácil de aprender.

#### 2.3.3.1. GUI para la base de datos

La GUI para manejar la base de datos debe de registrar los datos de los todos los atletas y guardarlos en una base de datos. Se deben almacenar de tal manera que sea posible abrirlas con la GUI de monitoreo y ver la información del atleta. Los datos que se van a ingresar son los siguientes:

**Nombre** Es el nombre del atleta al que van a pertenecer los datos.

**Edad** Este dato es importante para poder calcular la FCmax.

**FCmin** Es la frecuencia cardiaca en estado de reposo del atleta, para conocerla se va a integrar en la GUI una ventana que va a desplegar la frecuencia cardiaca cada minuto.

**FCmax** Es la frecuencia máxima que el atleta puede alcanzar, se puede calcular mediante la ecuación 2, este proceso de calcular la FCmax está automatizado mediante un botón.

**Intensidad baja** Es el rango de frecuencia cardiaca en las que el atleta se va a encontrar realizando un esfuerzo de intensidad baja. Este valor se calcula usando la ecuación 3.

**Intensidad media** Es el rango de frecuencia cardiaca en las que el atleta se va a encontrar realizando un esfuerzo de intensidad media. Este valor se calcula usando la ecuación 3.

**Intensidad alta** Es el rango de frecuencia cardiaca en las que el atleta se va a encontrar realizando un esfuerzo de intensidad alta. Este valor se calcula usando la ecuación 3.

**Intensidad máxima** Es el rango de frecuencia cardiaca en las que el atleta se va a encontrar realizando un esfuerzo de intensidad máxima. Este valor se calcula usando la ecuación 4.

**Archivo** Es el nombre del archivo en el que se van a guardar los datos.

**Carpeta** Es la carpeta en la que se va a guardar el archivo.

En esta GUI hay tres eventos importantes que se deben manejar:

**Guardar el archivo** Para guardar el archivo, se debe adquirir el texto de las entradas para ingresarlas a una lista, generar el archivo en el directorio especificado y guardar la lista en el archivo de texto generado. Esto se aprecia en el código siguiente:

```

def guardar_lista(lista , ruta):           1
    fichero = file(ruta , "w")
    nl = lista
    pickle.dump(nl , fichero)

def guardar():                             2
    v = str(e1.get())
    w = str(e2.get())
    x = str(e3.get())
    y = str(e4.get())
    ib = str(e7.get())
    im = str(e8.get())
    ia = str(e9.get())
    ima = str(e10.get())
    archivo = e5.get() + '.txt'
    carpeta = e6.get()
    directorio = '/home/pi/' + carpeta      3
    if os.path.exists(directorio):
        directorio = directorio
    else:
        os.mkdir(directorio)
        os.chmod(directorio , 0777)
    guardar_lista((v , w , x , y , ib , im , ia , ima) ,
    directorio + '/' + archivo)            4

```

### Código 2.3 Código para Guardar Archivo

El programa del código 2.3 funciona de la siguiente manera:

- 1** Primero se define una función para guardar el archivo, esta función pide la lista de los datos que se van a guardar y la ruta del archivo en que se va a guardar.
- 2** Después se define una función que va a obtener el texto de las entradas de texto y transformarlos en cadenas para poder manejarlos.
- 3** En esta línea se define el directorio en el que se va a guardar el archivo y se comprueba si este directorio existe o no, si no existe lo crea.
- 4** En esta línea se llama a la función que va a guardar el archivo, función que se definió en la línea 1.

**Cálculo de la FCmax** Mediante un botón se puede indicar que se calcule la FCmax, a través de la ecuación 2.

```
def calcular():
    e = e2.get()
    edad = int(e)
    fm = 208.75 - 0.73 * edad
    fm = int(fm)
    e4.delete(0,END)
    e4.insert(0,fm)
```

Código 2.4 Código para Calcular la FCmax

El programa calcula la FCmax y funciona de la siguiente manera:

- 1 Se define la función que va a calcular la FCmax, para esto la función va a obtener la edad de la entrada de texto correspondiente.
- 2 En esta línea se calcula la FCmax usando la ecuación 2.
- 3 El valor obtenido de la FCmax se imprime en el campo de texto correspondiente.

**Cálculo de la intensidad de esfuerzo** Para esto se va a ocupar la ecuación 3 que se vio anteriormente. En el siguiente programa se puede ver el cálculo del rango para la intensidad baja, el cálculo de las demás intensidades se hace de la misma manera solamente cambiando los porcentajes.

```
def intensidadb():
    fm = e3.get()
    fma = e4.get()
    fm = int(fm)
    fma = int(fma)
    ib1 = (fma - fm) * 0.5 + fm
    ib2 = (fma - fm) * 0.7 + fm
    ib1 = int(ib1)
    ib2 = int(ib2)
    ib1 = str(ib1)
    ib2 = str(ib2)
    ib = ib1 + "\_a_\"+ib2
    e7.delete(0,END)
    e7.insert(0,ib)
```

Código 2.5 Código para Calcular la Intensidad de Esfuerzo

El programa anterior calcula el rango de frecuencias para la intensidad de esfuerzo baja, de igual manera se calculan las demás intensidades:

- 1 Se define la función que va a determinar el rango de frecuencia cardiaca para cada intensidad de esfuerzo baja.
- 2 En esta línea se calcula la frecuencia cardiaca mínima para la intensidad de esfuerzo baja, posteriormente se calcula la frecuencia máxima, ambas usando la ecuación 3.
- 3 El rango de frecuencias para la intensidad de esfuerzo baja se imprime en el cuadro de texto correspondiente.

Además de manejar estos eventos, se creó la estructura de la ventana que va a estar basada en cuadros de texto para ingresar los datos, etiquetas que orientaran al usuario sobre la función de los cuadros de texto y una ventana en donde aparecerá el valor de la FCmin. Para obtener la FCmin se generó una variable que aumente de valor con cada pulso detectado y que después de un minuto imprima este valor en una ventana de la GUI.

```
def adquisicion_de_pulsos(channel):                                1
    time_now = time.time()
    if(time_now - time_stamp) >= 0.014:
        pulsos = pulsos + 1
        time_stamp = time_now

GPIO.add_event_detect(18, GPIO.RISING,                          2
callback=adquisicion_de_pulsos)
```

#### Código 2.6 Código para Adquisición de Pulsos por Minuto

Este programa obtiene el total de pulsos que llegan a la Raspberry Pi en un minuto:

- 1 Esta función registra los pulsos que son detectados en la variable pulsos.
- 2 Esta línea declara el pin que se va a usar para detectar los pulsos y llama a la función adquisicion\_de\_pulsos cada que hay una interrupción.

El siguiente programa muestra la creación de la ventana principal, así como las entradas de texto y de las etiquetas que conforman la estructura de la GUI de Obtención de Datos.

```
v0=Tk()                                                         1
v0.resizable(0,0)
v0.geometry("560x220")
v0.title("Obtencion_de_Datos")
```

```

l1 = Label(v0, text=" Datos _del _Alumno" , width=20)
l1.grid(row=0, column=2)
l2 = Label(v0, text=" Nombre: _" )
l2.grid(row=1, column=0, sticky=W)
e1 = Entry(v0, bd = 5, justify = CENTER)
e1.grid(row=1, column=2)
e1.focus_set()
l3 = Label(v0, text=" Edad: _" )
l3.grid(row=2, column=0, sticky=W)
e2 = Entry(v0, bd = 5, justify = CENTER)
e2.grid(row=2, column=2)
l4 = Label(v0, text=" Frecuencia _Minima: _" )
l4.grid(row=3, column=0, sticky=W)
e3 = Entry(v0, bd = 5, justify = CENTER)
e3.grid(row=3, column=2)
b1 = Button(v0, text=" Frecuencia _Maxima: _" ,
width = 20, command=calcular)
b1.grid(row=4, column=0, sticky=W)
e4 = Entry(v0, bd = 5, justify = CENTER)
e4.grid(row=4, column=2)
b3 = Button(v0, text=" Intensidad _Baja: _" ,
width = 20, command=intensidadb)
b3.grid(row=5, column=0, sticky=W)
e7 = Entry(v0, bd = 5, justify = CENTER)
e7.grid(row=5, column=2)
b4 = Button(v0, text=" Intensidad _Media: _" ,
width = 20, command=intensidadm)
b4.grid(row=6, column=0, sticky=W)
e8 = Entry(v0, bd = 5, justify = CENTER)
e8.grid(row=6, column=2)
b5 = Button(v0, text=" Intensidad _Alta: _" ,
width = 20, command=intensidada)
b5.grid(row=7, column=0, sticky=W)
e9 = Entry(v0, bd = 5, justify = CENTER)
e9.grid(row=7, column=2)
b6 = Button(v0, text=" Intensidad _Maxima: _" ,
width = 20, command=intensidadma)
b6.grid(row=8, column=0, sticky=W)
e10 = Entry(v0, bd = 5, justify = CENTER)
e10.grid(row=8, column=2)
l5 = Label(v0, text=" Archivo: _" )
l5.grid(row=9, column=0, sticky=W)

```

```

e5 = Entry(v0, bd = 5, justify = CENTER)
e5.grid(row=9,column=2)
l6 = Label(v0, text=" Carpeta : ")
l6.grid(row=10,column=0,sticky=W)
e6 = Entry(v0, bd = 5, justify = CENTER)
e6.grid(row=10,column=2)
b2 = Button(v0, text=" Guardar" ,command=guardar)
b2.grid(row = 11, column = 2)
frame1 = Frame(v0)
frame1.grid(row = 0, column = 4, rowspan = 11)
scroll1=Scrollbar(frame1 ,bd=2)
list1=Listbox(frame1 ,bd=5,height = 20,width = 10)
list1.grid(row = 0,column = 4)
colocar_scrollbar(list1 , scroll1)
l11 = Label(v0, text = " Frecuencia Cardiac en Reposo" ,
width = 30)
l11.grid(row = 0, column = 4)

v0.mainloop()

```

3

### Código 2.7 Código para Crear la Ventana Principal

Este programa funciona de la siguiente manera:

- 1 Se crea la ventana principal, se define el tamaño de la ventana y el título.
- 2 En esta parte del código se crean las etiquetas, los cuadros de texto y los botones que se van a utilizar para crear la interfaz, además se les da una posición específica dentro de la ventana principal.
- 3 Esta línea ejecuta en un ciclo infinito la ventana de la GUI.

En la figura 2.16 se puede ver la ventana diseñada para esta GUI.



Figura 2.16: GUI para la Obtención de Datos

### 2.3.3.2. GUI para el monitoreo de la frecuencia cardiaca

En la GUI de monitoreo se pueden abrir cualquiera de los archivos guardados con anterioridad y visualizar los datos del atleta que éste contenga. También se puede visualizar el valor de la frecuencia cardiaca y guardar estos valores en un archivo de texto en el momento que se desee.

En esta GUI hay dos eventos importantes que se deben manejar:

- Abrir los archivos de la base de datos
- Guardar los datos de la frecuencia cardiaca

**Abrir los archivos de la base de datos** La manera más sencilla para abrir los documentos que contienen la información del atleta es mediante una ventana o menú que permita visualizar las carpetas y los documentos que contienen. Una vez ubicados se tienen que abrir de tal manera que se puedan visualizar los datos en la GUI. A continuación se muestra su código:

```
def abrir():
    fileName = askopenfilename()
    try:
        cargar_lista(fileName)
    except:
        print("Error al abrir el archivo")

def cargar_lista(ruta):
    fichero = file(ruta)
```

```

lista_recuperada = pickle.load(fichero)
texto.insert(END, lista_recuperada)

```

### Código 2.8 Código para Abrir los Archivos de la Base de Datos

En este programa se va a recuperar el archivo que se guardó con la anterior GUI y desplegar los datos que contenga en un cuadro de texto, éste funciona de la siguiente manera:

- 1 Esta función crea el menú y la ventana que permite buscar el archivo que se quiere abrir.
- 2 Esta función recupera los datos que se encuentran en el archivo que se eligió abrir y los imprime en el cuadro de texto respectivo.

**Guardar los datos de la frecuencia cardiaca** Para guardar estos datos se debe generar una lista con todos los datos que se hayan desplegado hasta ese momento y generar un archivo de texto nuevo. En éste se va a guardar la lista en la misma ruta del archivo de datos del atleta que se abrió.

```

def guardar_lista(lista, ruta):          1
    fichero = file(ruta, "w")
    nl = lista
    pickle.dump(nl, fichero)

def guardar():                            2
    global localtiempo
    localtime = time.asctime(time.localtime(time.time()))
    global fileName
    datos = list2.get(0, END)
    datos = localtiempo + str(datos) + localtime
    archivo = fileName
    nombre, extension = os.path.splitext(archivo)
    guardar_lista(datos, nombre + '_' + localtime + '_'.txt')

```

### Código 2.9 Código para Guardar los Datos de Entrenamiento

Este programa funciona de manera parecida al utilizado en la GUI para la base de datos:

- 1 Esta función guarda los datos del entrenamiento en el directorio especificado.

**2** Esta función recolecta los datos del entrenamiento y le asigna un nombre único al archivo usando la fecha y hora a la que se guardó.

También se tuvieron que crear las ventanas de texto, las etiquetas y el menú necesarios para hacer funcional la GUI, se debe tener en cuenta que no se va a contar con un teclado al momento de hacer el monitoreo en el entrenamiento por lo que se evitó que el usuario tenga que ingresar texto.

La ventana de esta GUI incluye un menú de opciones, en el que se van a encontrar las siguientes:

**Abrir** Esta opción va a desplegar una ventana, que se puede ver en la figura 2.17, en la que se va a escoger el archivo del atleta que se desee abrir. Una vez que se abra el archivo, los datos van a ser visualizados en un cuadro de texto.

**Guardar** Con esta opción se van a guardar los datos de la frecuencia cardiaca que tuvo el atleta durante el entrenamiento.

**Salir** Esta opción va a cerrar la ventana.

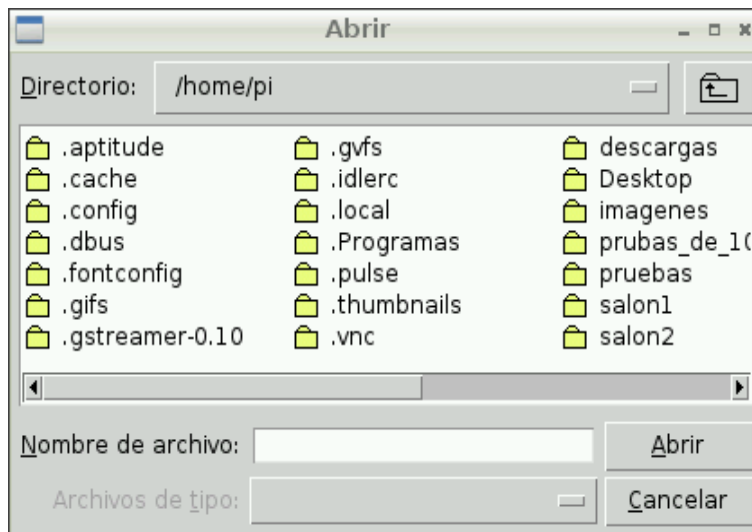


Figura 2.17: Ventana para Abrir Archivos

La parte más importante de esta ventana es la del cuadro de texto en donde se van a visualizar los valores de la frecuencia cardiaca. El programa con el que se crea la ventana y objetos que se necesitan es el siguiente:

```
v0=Tk()  
v0.resizable(0,0)  
v0.geometry("420x190")
```

1

```

v0.title("MONITOR_RITMO_CARDIACO")

menubar = Menu(v0)
v0.config(menu=menubar)
filemenu = Menu(menubar, tearoff=0)
filemenu.add_command(label = "Abrir", command = abrir)
filemenu.add_command(label = "Guardar", command = guardar)
filemenu.add_separator()
filemenu.add_command(label = "Salir", command = v0.quit)
menubar.add_cascade(label = "Archivo", menu = filemenu)

l1 = Label(v0, text="Datos del Alumno", width=20)
l1.grid(row = 0, column = 0)
frame1 = Frame(v0)
frame1.grid(row = 1, column = 0)
texto = Listbox(frame1, height = 1, width = 30)
texto.grid(row = 2, column = 0)
l2 = Label(v0, text = "Frecuencia Cardiaca", width=20)
l2.grid(row = 0, column = 1)
frame2 = Frame(v0)
frame2.grid(row = 1, column = 1)
scroll1 = Scrollbar(frame2)
list2 = Listbox(frame2, height = 10, width = 10)
list2.grid(row = 2, column = 1)
colocar_scrollbar(list2, scroll1)

v0.mainloop()

```

Código 2.10 Código para la Creación de la Ventana Principal

Este programa crea la ventana principal de la GUI para el monitoreo de la frecuencia cardiaca.

- 1** En esta línea del programa se crea la ventana principal y se le asigna título y tamaño.
- 2** En esta línea se crea el menú con las opciones abrir, guardar y salir.
- 3** En esta sección se crean las etiquetas y los cuadros de texto necesarios para desplegar la información.
- 4** En esta línea se ejecuta en un ciclo infinito la creación de la ventana de la GUI.

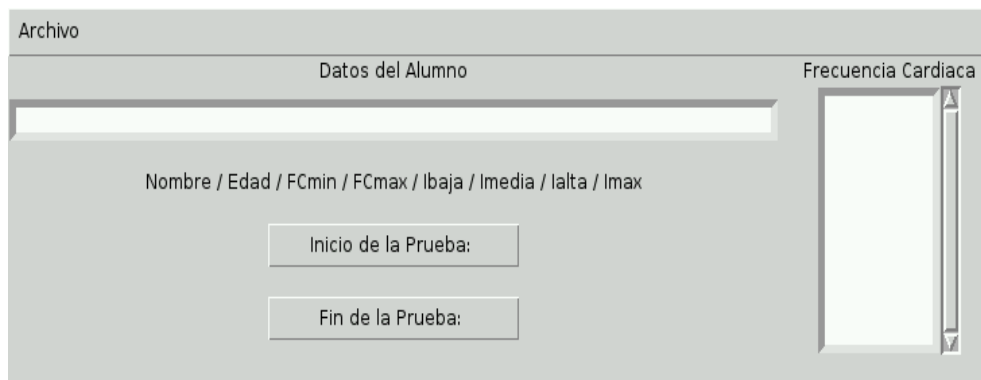


Figura 2.18: GUI para el Monitoreo de la Frecuencia Cardiaca

Además se agregaron dos botones con etiquetas de inicio y final de prueba para que el profesor pueda saber en qué momentos se realizaron pruebas durante el entrenamiento.

En la figura 2.18 se puede ver la ventana de la GUI para el monitoreo de la frecuencia cardiaca.

Para concluir la GUI se obtuvo la frecuencia cardiaca, usando el programa que se presentó anteriormente. Esto permitió que en la GUI se pudiera imprimir en un cuadro de texto la frecuencia cardiaca.

## 2.4. Construcción final

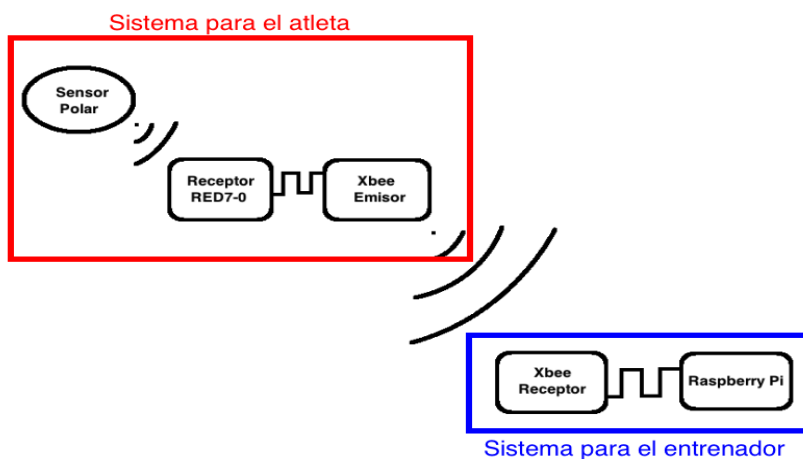


Figura 2.19: Diagrama del Sistema

En la figura 2.19 se puede ver el diagrama del sistema de Monitoreo de Ritmo Cardíaco a Larga Distancia donde se compone de dos sistemas, el sistema que va a tener consigo el atleta y el sistema que va a tener el entrenador.

Para incorporar estos sistemas se tuvieron que diseñar los PCB's que permitieron realizar las conexiones entre los diferentes componentes de los sistemas, para esto se utilizó el programa de licencia libre Fritzing.

También se tuvieron que diseñar los *case* o gabinetes que permitieron tener de manera embebida a los dos sistemas.

### 2.4.1. Sistema para el atleta

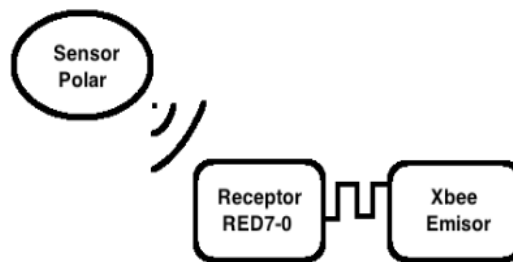


Figura 2.20: Diagrama del Sistema para el Atleta

El sistema que se diseñó para ser usado por el atleta, está integrado por la etapa de la medición del ritmo cardíaco y la parte de la etapa de la transmisión de los datos a larga distancia, es decir, el sensor de banda elástica Polar, el receptor RED7-0 y el emisor XBee.

Ya que este sistema es portátil se incluyó la batería que lo va a alimentar y su regulador de voltaje para mantener el voltaje a 3.3V. Además se tuvo que diseñar el *case* de tal manera que se pudiera recargar la batería sin tener que desconectarla del sistema.

En la figura 2.21 se puede ver el diagrama de la tarjetala y en la figura 2.22 se puede ver el PCB del circuito ya hecho.

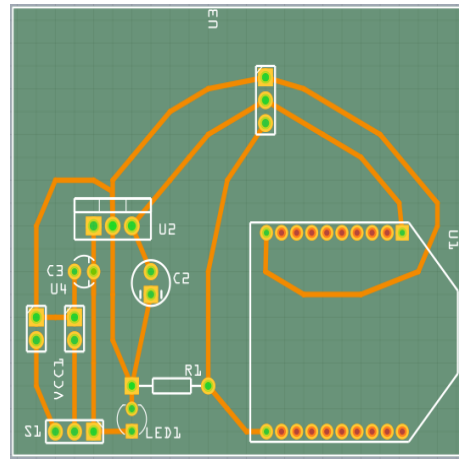


Figura 2.21: Diagrama de la Tarjeta

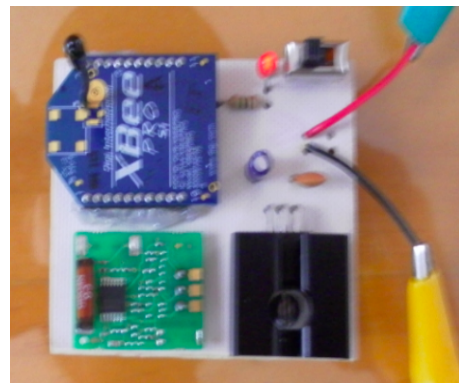


Figura 2.22: PCB del circuito

También se necesitó construir un *case* para incluir a la tarjeta y a la batería. Esto se diseño usando un programa CAD.

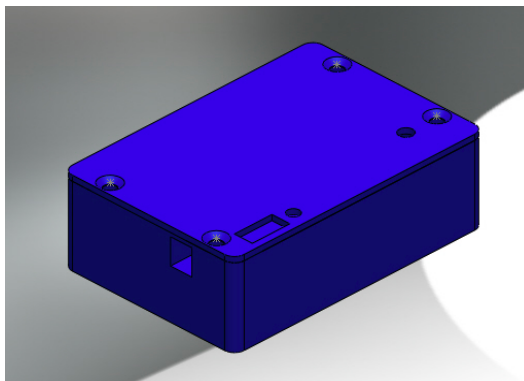


Figura 2.23: *Case* para el Sistema del Atleta

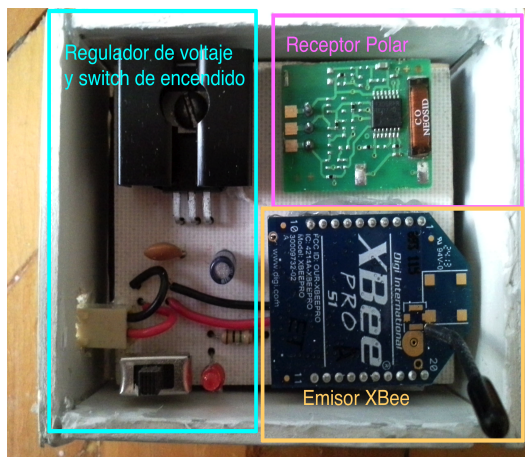


Figura 2.24: Sistema para la Medición del Ritmo Cardíaco

#### 2.4.2. Sistema para el entrenador



Figura 2.25: Diagrama del Sistema para el Entrenador

El diagrama del sistema de monitoreo para el entrenador, se puede ver en la figura 2.25, y se compone del receptor XBee y de la Raspberry Pi. La comunicación

de estos dispositivos se va a dar a través del pin 11 del XBee y del pin 6 de la Raspberry Pi, esta conexión se puede hacer de manera directa pues ambos funcionan a 3.3V y la Raspberry Pi tiene resistencias de pull up-down internas.

En la figura 2.26 se puede ver el diagrama de la tarjeta y en la figura 2.27 se puede ver el sistema conectado.

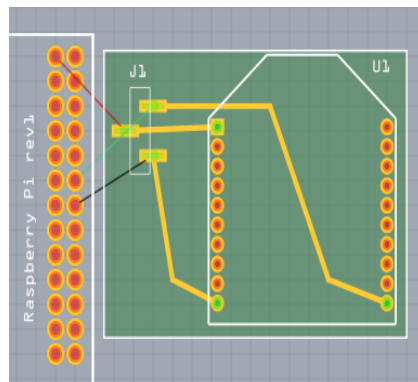


Figura 2.26: Diagrama de la Tarjeta

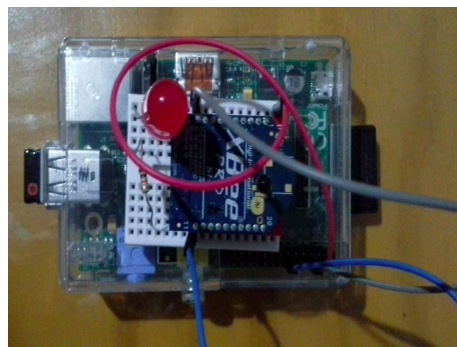


Figura 2.27: Circuito del Sistema para el Entrenador



# Capítulo 3

## Pruebas y Resultados

Para comprobar el funcionamiento de todo el sistema, se realizaron pruebas de cada una de las etapas que lo conforman, las cuales se presentan a continuación.

### 3.1. Pruebas con el sensor Polar

Lo primero que se evaluó es la precisión del sensor Polar y del receptor RED7-0, para esto se compararon la señal que entrega el receptor y la señal que entrega el sensor de tacto Pulse Sensor [30], que funciona con la tarjeta de desarrollo Arduino.



Figura 3.1: Sensor Pulse Sensor

La prueba consistió en tener conectado el sensor Pulse Sensor y el sensor Polar a un sujeto que se monitoreo en estado de reposo y con un osciloscopio comparar las señales que entregan los dos sensores.

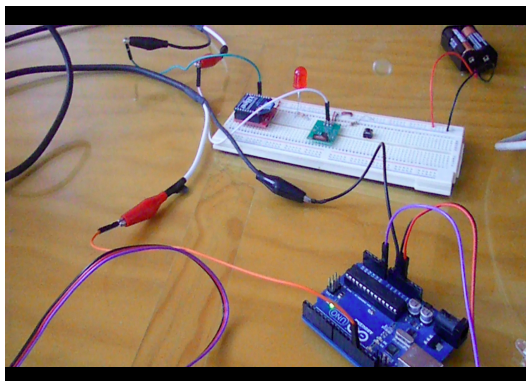


Figura 3.2: Conexión de los Sensores al Osciloscopio

### 3.2. Resultados de las pruebas con el sensor Polar

En la figura 3.3 se puede ver la comparación de las salidas en el osciloscopio. La señal de arriba es la del sensor Pulse Sensor y la señal de abajo es la del sensor Polar.

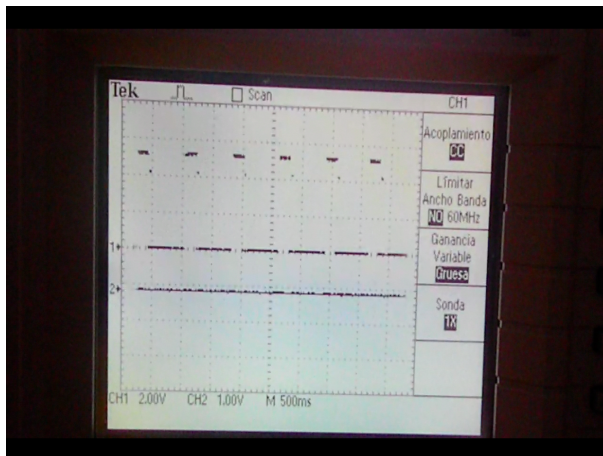


Figura 3.3: Comparación de las Señales de los Sensores

Se puede ver que la señal del sensor Polar aparece ligeramente después que la señal del sensor Pulse Sensor y que dura menos tiempo activa.

Después de realizar la prueba durante 10 minutos, no se tuvo ninguna pérdida de información por parte del sensor Polar en comparación al sensor Pulse Sensor, lo que indica que el sensor sí es capaz de captar los pulsos del corazón y entregar la información de manera adecuada.

### 3.3. Módulos XBee

#### 3.3.1. Pruebas de transmisión de datos

Después de verificar que el sensor funciona bien y con precisión, se debe comprobar que no se pierda información en la transmisión de los módulos XBee. Para comprobar esto se compararon la señal de entrada del XBee emisor y la señal de salida del XBee receptor. Se conectó al osciloscopio la salida del receptor RED7-0 conectada al pin 11 del XBee y la salida del pin 11 del módulo receptor XBee, de esta manera se puede comprobar que los datos que entren al XBee emisor salgan del XBee receptor.

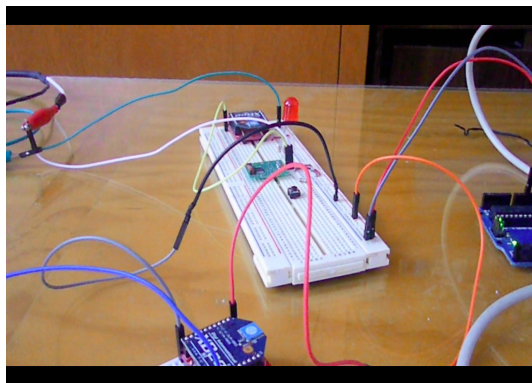


Figura 3.4: Conexión de los Módulos XBee al Osciloscopio

#### 3.3.2. Resultados de transmisión de datos

Después de realizar la prueba durante 10 minutos, se encontró que se pierden algunos datos, o en este caso que se pierden algunos pulsos. En los diez minutos se pueden perder en total alrededor de 10 pulsos de manera aleatoria, durante un minuto se puede no perder ningún pulso pero en el siguiente se pueden perder dos o tres pulsos, es decir la pérdida de la información no tiene un patrón constante.

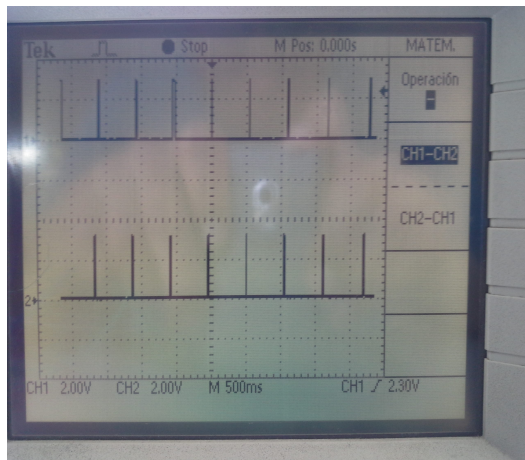


Figura 3.5: Pulsos Enviados y Recibidos de los Módulos XBee



Figura 3.6: Pérdida de Pulso en el Receptor XBee

La razón de esta pérdida se considera que es la interferencia que puede haber de señales como, wi-fi, bluetooth, señal de celular, o algún otro tipo de señal inalámbrica que trabaje en el rango de 2.4 GHz.

### 3.3.3. Pruebas de distancia de transmisión

Para comprobar la distancia de la transmisión de datos se colocó el módulo XBee receptor en el centro de un campo de futbol soccer y mientras un sujeto lo monitoreaba, otro sujeto con el módulo XBee emisor y el sensor Polar, se alejó en línea recta hasta que la transmisión se vio interrumpida, la distancia máxima que se alcanzó fue de 14 mts, una distancia muy por debajo de los 100 mts que el

dispositivo promete. Después se comprobó que la distancia fuera constante en toda la circunferencia, y se encontró que en algunas partes la distancia de transmisión disminuía hasta los 10 mts.

La distancia máxima que se planteo alcanzar fueron 100 mts por lo que los 14 mts se sienten insuficientes para el alcance que se planteo que el sistema tendría. Los módulos que se usaron fueron los XBee Serie 1 con antena de chip y como se vio en la comparación de las antenas, los XBee Pro prometen un mayor alcance por lo que se decidió cambiar los módulos XBee Serie 1 con antena de chip por unos XBee Pro Serie 1 con antena de cable, para de esta manera aumentar el alcance de transmisión.

### 3.3.4. Resultados de la distancia de transmisión con la serie Pro

Ya con los XBee Pro se realizó la misma prueba en el campo de futbol soccer, se colocó el XBee receptor a la mitad del campo y el XBee emisor se alejó en línea recta utilizando una cinta métrica hasta que la transmisión se viera interrumpida, en esta ocasión se alcanzó una distancia de transmisión en línea recta de 50 mts, y de manera radial no se pudo comprobar pues no se tenía el espacio suficiente para hacerlo, pero la comparación de las antenas indica que el radio de la antena de cable es la más constante de todos los tipos de antenas.

El campo en el que se realizó la prueba, es el campo de futbol de la Facultad de Cultura Física (FCF) de la BUAP, en la figura 3.7 se pueden ver sus medidas y en la figura 3.8 se puede ver la realización de la prueba.

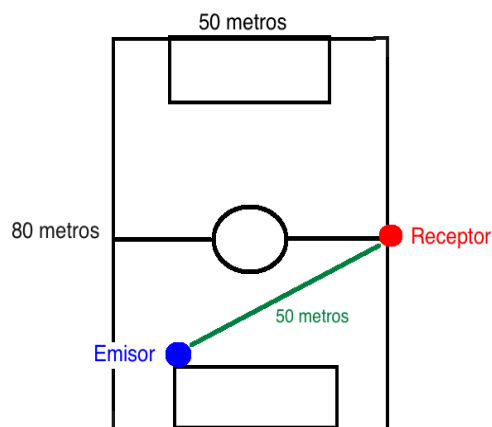


Figura 3.7: Campo de Futbol de la FCF



Figura 3.8: Prueba de la Distancia de Transmisión

### 3.4. Pruebas para el cálculo de la frecuencia cardiaca

Para comprobar que el programa que calcula la frecuencia cardiaca funciona de manera adecuada al momento del entrenamiento se contó con la ayuda del Dr. Jacinto Carvente Rodríguez, profesor de atletismo de la FCF de la BUAP. El profesor tiene experiencia con el uso de pulsómetros y él nos pudo indicar si el sistema que se desarrolló era funcional para realizar entrenamiento.

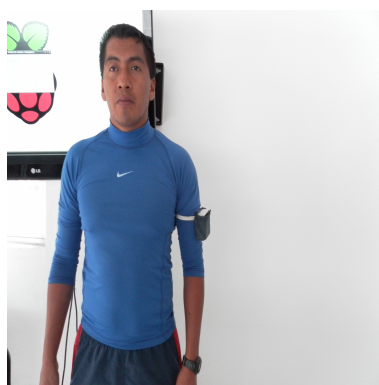


Figura 3.9: Prueba del Cálculo de la Frecuencia Cardiaca

La prueba se efectuó en el campo de futbol de la FCF, el profesor se colocó el sistema para el atleta y realizó una serie de ejercicios para aumentar su frecuencia cardiaca, entre cada ejercicio se tomaba un tiempo de descanso para que ésta se normalizara. El objetivo era que el sistema captará estos cambios en la frecuencia cardiaca y mostrara valores de acuerdo a la intensidad que el profesor aplicaba a cada ejercicio.

### 3.5. Resultados del cálculo de la frecuencia cardiaca

Después de realizar la prueba se vio que al estar en el límite de la distancia de transmisión ó en los movimientos bruscos que se realizan al hacer ejercicio se pueden llegar a perder algunos datos o pulsos en la transmisión de los XBee y esto afecta de gran manera el cálculo de la frecuencia cardiaca y los valores que se despliegan pueden variar mucho. En la figura 3.10 se puede ver la variación de los valores de frecuencia cardiaca.

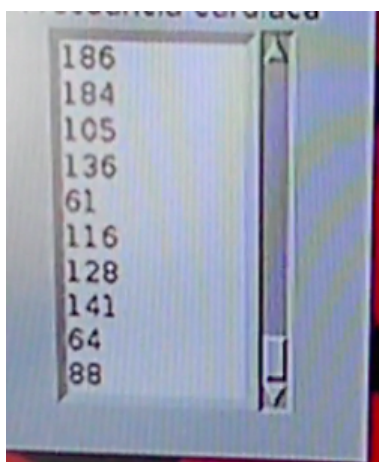


Figura 3.10: Variación de la Frecuencia Cardiaca por Pérdida de Datos

El programa que se utilizó para realizar la prueba mide el tiempo que hay entre los pulsos y cuando ya se recibieron 5, obtiene el promedio de estos valores y compara el resultado en la tabla que se puede ver en el código 2.2 para asignarle un valor de frecuencia cardiaca. Dado que se suman los tiempos en caso de pérdida de algún dato el tiempo total aumenta demasiado, por lo que se debe desarrollar otro programa para calcular la frecuencia cardiaca que evite que la pérdida de pulsos afecte su valor.

Una vez que se obtuvieron los valores se decidió comparar tres programas:

- a) El programa original que se utilizó para realizar la primer prueba.
- b) El programa que obtiene la mediana de los tiempos y asigna el valor de frecuencia cardiaca correspondiente. En el apéndice B se puede ver el programa para calcular la mediana de los tiempos obtenidos.
- c) El programa que obtiene el promedio de los tiempos.

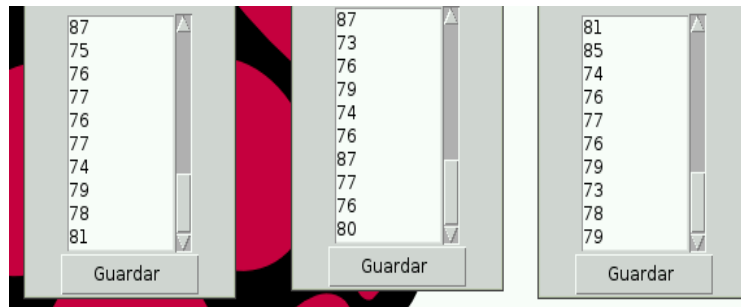


Figura 3.11: izquierda) a, centro) b, derecha) c

Posteriormente se compararon los tres programas al mismo tiempo y se guardaron los datos para graficarlos y compararlos. En la figura 3.11 se pueden ver los 3 programas ejecutándose al mismo tiempo y en las figuras 3.12, 3.13 y 3.14 se pueden ver las gráficas de los mismos.

Al comparar las gráficas se puede apreciar que la gráfica de la mediana tiene menos variación, es decir que la pérdida de pulsos en el envío a larga distancia no afecta demasiado el valor de la frecuencia cardiaca, por lo que se decidió usar este programa en el sistema final.

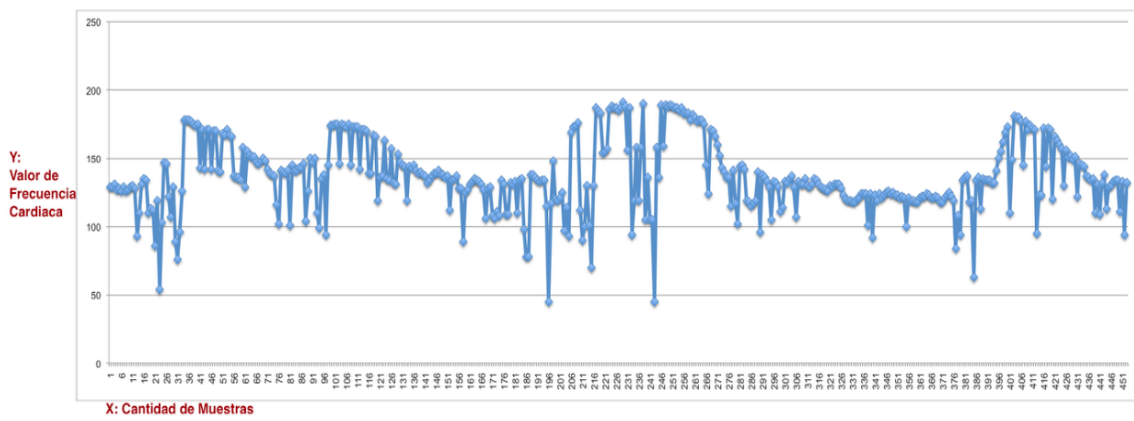


Figura 3.12: Gráfica de los Datos Obtenidos con el Programa Original

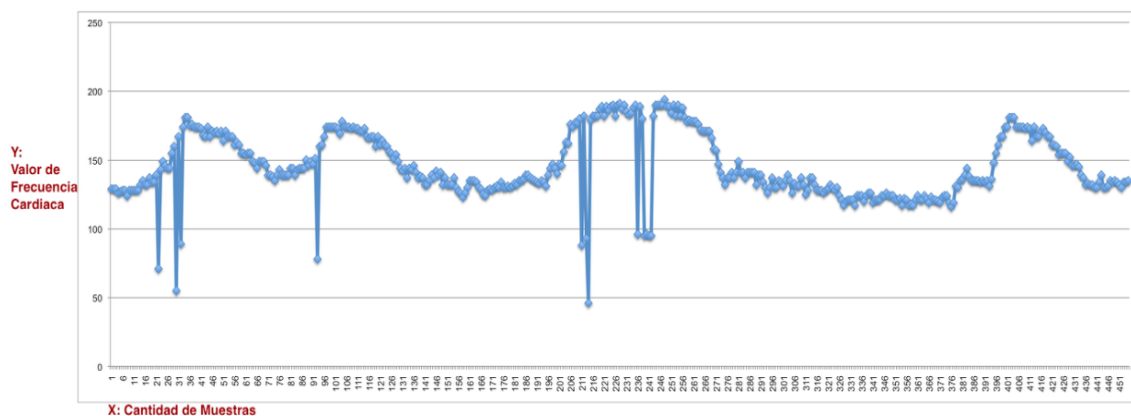


Figura 3.13: Gráfica de los Datos Obtenidos con el Programa de la Mediana

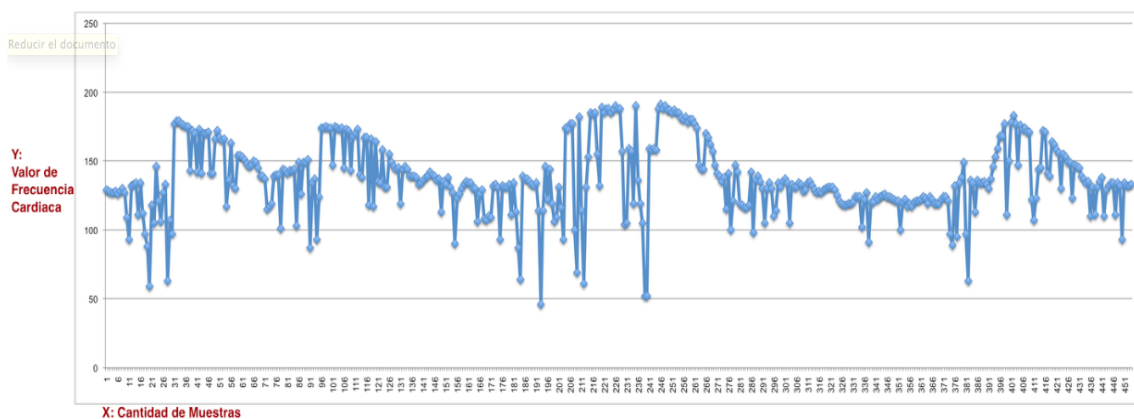


Figura 3.14: Gráfica de los Datos Obtenidos con el Programa del Promedio

### 3.6. Pruebas finales del sistema

La prueba final consistió en verificar la distancia máxima de transmisión de datos en donde el sistema es estable y en verificar que el sistema de monitoreo capte las variaciones de la frecuencia cardiaca y no se vea afectado por la pérdida de pulsos en el envío.

Para comprobar el alcance de la transmisión del sistema completo, se realizó la misma prueba que se hizo para comprobar la distancia de transmisión de los módulos XBee. En el sistema del entrenador se usó una pantalla plana para la visualización del monitoreo. El sistema de monitoreo se ubicó a la mitad del campo de futbol y con la cinta métrica se comprobó la distancia.

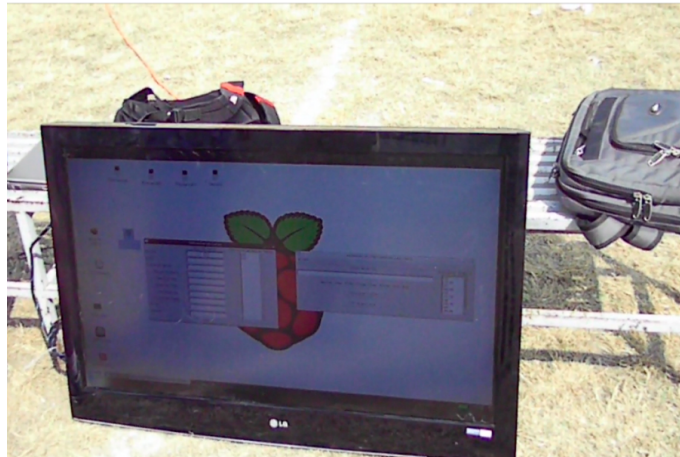


Figura 3.15: Sistema de Monitoreo



Figura 3.16: Sistema para el Atleta

En base a la distancia máxima de alcance previamente determinada, un estudiante de la FCF procedió a realizar las siguientes pruebas con el sistema:

- Trote
- Sprint de corto alcance
- Sprint de medio alcance
- Sprint de largo alcance



Figura 3.17: Profesor de Atletismo y Alumno



Figura 3.18: Alumno Usando el Sistema de Medición de Ritmo Cardíaco

Estas pruebas sirvieron para variar su frecuencia cardiaca. En la figura 3.17 se puede ver al profesor de atletismo y al alumno que ayudó a realizar las pruebas. A la vez que se realizaron las actividades, se comprobó que el sistema captara los cambios de frecuencia cardiaca y que estuvieran de acuerdo a la intensidad del entrenamiento, además de que el valor de frecuencia cardiaca no se viera afectado por la pérdida de algunos pulsos en el envío.

Ambas pruebas, la de distancia de alcance del sistema y la de la lectura de la frecuencia cardiaca se realizaron bajo las siguientes condiciones:

- Se ocupó la ayuda de un alumno de la FCF.
- Se realizaron bajo la supervisión del Dr. Jacinto Carvente Rodríguez.
- Se realizaron en un horario de 1 pm a 3 pm y en un espacio abierto.



Figura 3.19: Alumno Realizando Pruebas de Variación de Frecuencia Cardiaca

### **3.7. Resultados finales del sistema**

La distancia que alcanzó el sistema no llega a la distancia máxima de 100 mts que se planteó que se planteó al principio, en base a las especificaciones de la hoja de datos. Teniendo en consideración las pruebas realizadas, se encontró que la distancia máxima estable del sistema fue de 50 mts. Esta distancia es funcional y permite que el sistema se pueda utilizar en el entrenamiento de atletas. Además los valores de la frecuencia cardiaca que el sistema muestra son bastante confiables y la ventaja de tener una base de datos de los atletas y poder guardar los datos

de la frecuencia cardiaca de sus sesiones de entrenamiento es de gran ayuda para el entrenador.

Además de estas pruebas se realizaron tres pruebas más, dos con el profesor de atletismo como sujeto a monitorear y una más con otro alumno de la FCF, estas pruebas se realizaron alrededor de las 3 pm, horario similar a las pruebas anteriores. En la tabla 3.1 se pueden ver las pruebas.

<b>Núm. de Prueba</b>	<b>Sujeto a Monitorear</b>	<b>Horario</b>	<b>Pruebas Realizadas</b>	<b>Resultados</b>
1	Alumno FCF	1 pm a 2 pm	Distancia máxima	50 mts
2	Alumno FCF	3 pm a 5 pm	Trote Sprint Corto Sprint Medio Sprint Largo	Las mediciones de la FC concordaron con lo esperado
3	Profesor de Atletismo FCF	3 pm a 4 pm	Distancia máxima Trote Sprint Corto	Se midieron los cambios de la FC de acuerdo a los ejercicios realizados, a una distancia de 50 mts
4	Profesor de Atletismo FCF	2 pm a 4 pm	Distancia máxima Sprint corto Sprint medio Sprint largo	Se midieron los cambios de la FC de acuerdo a los ejercicios realizados, a una distancia de 55 mts
5	Segundo alumno FCF	3 pm a 5 pm	Distancia máxima Sprint corto Sprint medio Sprint largo	Se midieron los cambios de la FC de acuerdo a los ejercicios realizados, a una distancia de 50 mts

Tabla 3.1: Pruebas Realizadas

### 3.8. Comparación del sistema con otros pulsómetros

Para verificar la validez de los resultados del sistema se compararon con los datos indicados por otro pulsómetro. Para esto se usó el sensor de banda pectoral del sistema propuesto y un sensor de tacto.

La primera comparación se realizó con el sensor de tacto ubicado en una banda

sin fin de la marca FreeMotion[9] que usa un sensor de la marca Polar. Un alumno del profesor de atletismo se colocó el sistema desarrollado y se subió a la banda para realizar un pequeño entrenamiento que consistió de trote ligero aumentando la intensidad hasta un trote intenso. Mientras realizaba el trote colocaba sus manos en el sensor para que el aparato desplegará su frecuencia cardiaca.



Figura 3.20: Comparación 1. Uso del sistema junto con una banda Sin Fin. El alumno ( playera verde) trae en el brazo el sistema desarrollado.

Se compararon los valores en momentos específicos, es decir, el profesor me indicaba el valor de frecuencia que se veía en el aparato y yo indicaba el valor que se veía en el sistema de monitoreo. Los valores mencionados eran bastante cercanos y la diferencia máxima entre estos valores llegó a ser de tres unidades en una sola ocasión, lo que no representa una variación notable de la frecuencia como para que el análisis del entrenamiento se vea afectado.

La segunda comparación se hizo con un pulsómetro de sensor de banda pectoral de la marca Polar, conformado por un sensor T31-coded que envía los datos de manera codificada y un receptor FT2 que se muestran en las figuras 3.21 y 3.22.



Figura 3.21: Pulsómetro Polar

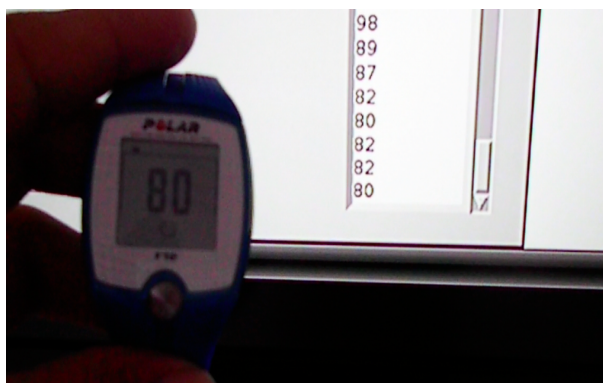


Figura 3.22: Comparación de los Datos

Para esta prueba el profesor de atletismo se colocó el sensor T31-coded y el sensor T34 que usa el sistema desarrollado. Una vez que ambos sistemas de monitoreo captaron los datos, el profesor realizó una serie de ejercicios para variar su frecuencia cardiaca y se tomó un video de los datos que el pulsómetro FT2 mostraba junto a los datos que el sistema desarrollado mostraba para de esta manera compararlos. Se pudo observar que la diferencia entre los datos llegó a ser de varias unidades, pero también se pudo observar que había momentos en los que el receptor FT2 tardaba en actualizar el valor de la frecuencia cardiaca o dejaba de captar los datos enviados por el sensor y éstas eran las razones de tal diferencia. En la figura 3.23 se puede ver la comparación de los datos obtenidos, los datos de color azul son los del receptor FT2 y los datos de color rojo son los del sistema.

La tercera comparación se hizo con un pulsómetro de la marca Timex, durante la realización de esta prueba se notó que algunos valores desplegados por el sistema de monitoreo variaban bastante de acuerdo a lo que se podía esperar. Por lo cual

se revisaron los sistemas de recepción y se encontró que el receptor RED7-0 en algunas ocasiones llegaba a captar la información enviada por el sensor Timex lo que interfirió con el sistema. Después de ver que había interferencia se revisó hasta que distancia ésta seguía ocurriendo. A más de medio metro entre el sensor Timex y el Receptor RED7-0 la interferencia dejaba de ocurrir. Debido a la interferencia la comparación con este pulsómetro ya no se pudo realizar pues el profesor no podía tener puestos ambos sensores.

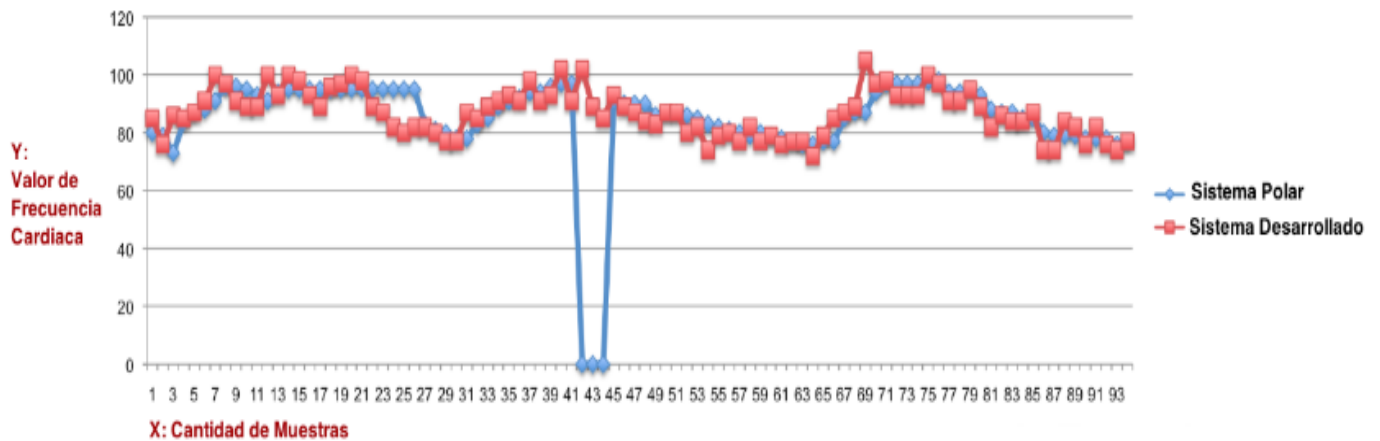


Figura 3.23: Gráfica de Comparación de los Datos

# Capítulo 4

## Conclusiones y Trabajo Futuro

### 4.1. Conclusiones

El sistema desarrollado monitorea la frecuencia cardiaca de un atleta en diferentes pruebas y a una distancia máxima de 50 mts. Este sistema es de gran apoyo para mejorar el entrenamiento de atletas de alto rendimiento. Además este pulsómetro es un avance en el área de los pulsómetros especializados, ya que aporta un enfoque diferente integrando nuevas tecnologías para su desarrollo lo que abre un nuevo camino para el desarrollo de sistemas de medición de la frecuencia cardiaca. Los sensores de la marca Polar cuentan con gran calidad en su fabricación y en la precisión en sus mediciones, por lo que se recomienda su uso.

La computadora Raspberry Pi es de fácil de uso y de bajo costo, lo que permite su uso en sistemas embebidos en una amplia gama de aplicaciones que no necesiten gran capacidad de procesamiento. Ésta tarjeta se convirtió en un elemento importante al permitir que el sistema fuera portátil debido a sus dimensiones reducidas.

Se tuvo que aprender el lenguaje de programación Python que nos permitió explotar toda la capacidad de la Raspberry Pi, además cuenta con una gran cantidad de librerías, algunas específicas para la Raspberry Pi que te permiten realizar programas complejos de manera rápida. Ya que las librerías de Python para crear interfaces gráficas están hechas para ser usadas en computadoras normales la integración de éstas con los pines GPIO de la Raspberry Pi fue compleja.

Los módulos XBee permiten tener comunicación inalámbrica entre dos dispositivos electrónicos sin necesidad de usar un sistema de procesamiento adicional. Cabe mencionar que una vez que se integraron al sistema, se encontró que no alcanzan la distancia que viene especificado en sus hojas de datos y que son susceptibles a interferencia.

## 4.2. Trabajo futuro

Una de las cosas que se puede mejorar es reducir el tamaño del sistema para el atleta y mejorar el *case* de éste para hacerlo más ergonómico y fácil de usar.

El sistema desarrollado solamente permite monitorear a un atleta, pero el trabajo desarrollado tiene la posibilidad de que sea capaz de monitorear a más de un atleta, tentativamente podrían ser hasta 8 atletas.

Para que el sistema sea portátil se deben agregar la pantalla *touch* y un teclado USB que permitan interactuar con las GUI del sistema. Por el momento no se cuenta con la pantalla ni con el teclado pero se podría adquirir en un futuro cercano y de esta manera completar el sistema y hacerlo completamente portátil y muy reducido en tamaño.

Otra área en la que también puede mejorar es en la interfaz gráfica. Si se quiere poder monitorear a más de un atleta se deberá modificar la interfaz. Finalmente otro aspecto que también podría mejorarse es en el desarrollo de de la interfaz en el sentido de que incluya gráficas y no tan solo datos numéricos.

# Bibliografía

- [1] Y. A. KESANIEMI, E. J. DANFORTH, M. D. JENSEN, P. G. KOPELMAN, P. LEFEBVRE y B. A. REEDER, “Dose response issues concerning physical activity and health: an evidence based symposium.”, *Medicine and Science in Sport and Exercise*, vol. 33, pp. 351-358, Junio 2001.
- [2] P. RODRIGUEZ y F. CANÑADA (2014, Junio 25). *Efecto de un programa de entrenamiento de biofeedback sobre el mantenimiento de una tasa cardiaca constante durante el pedaleo y la carrera a baja intensidad*. [Artículo publicado en revista digital] Disponible en: <http://www.efdeportes.com/efd102/biofeed.htm>
- [3] J. C. BOUZAS MARINS, “Empleo de Ecuaciones para Predecir la Frecuencia Cardiaca Máxima en Carrera para Jóvenes Deportistas.”, *Archivos de Medicina del Deporte*, vol. 24, no. 118, pp. 112-120, Febrero 2007.
- [4] J. L. ALGARRA y A. GORROTXATEGUI, *Entrenar con pulsómetro*. Bilbao: Dorleta, 2001.
- [5] J. HOROWITZ, *Entrenamiento inteligente para maratón*. Ediciones Tutor, 2013.
- [6] S. EDWARDS, *Corazón inteligente. Entrenamiento con monitor de Ritmo Cardíaco*. Bilbao: Dorleta, 1999.
- [7] J. M. IRIGOYEN, *Cardiología y Deporte*. Gymnos, 1999
- [8] J. WILMORE y D. COSTILL, *Fisiología del Esfuerzo y del Deporte*. 5ª ed, 2004.
- [9] N. CRAIG, *Preparación física con pulsometro*. Bilbao: Dorleta, 2001.
- [10] D. FINK, *Atletas de Hierro. Preparación para el Ironman*. Paidotribo, 2007.
- [11] W. SCHNEIDER, *Fitness Movilidad, Fuerza* Scriba, 1993.
- [12] J. CAMPOS GRANELL y V. R. CERVERA, *Teoría y Planificación del Entrenamiento Deportivo*. Paidotribo, 2003.
- [13] D. A. BROOKS, *El Libro Personal del Trainer*, Paidotribo, 2007.

- [14] M. RICHARDSON y S. WALLACE, *Getting Started with Raspberry Pi*, UK: O'Reilly. 2013.
- [15] A. OYARCE, *XBee Guía del Usuario*. Ingeniería MCI LTDA, 2010.
- [16] A. DINH y T. WANG, "Bandage-Size Non-ECG Heart Rate Monitor Using ZigBee Wireless Link" in *International Conference on Bioinformatics and Biomedical Technology (ICBBT)*, Chendgu, Abril 2010, pp. 160-163.
- [17] N. S. A. ZULKIFILI, F. K. CHE HARAUN y N. S. AZAHAR, "XBee Wireless Sensor Networks for Heart Rate Monitoring in Sport Training" in *International Conference on Biomedical Engineering (ICoBE)*, Penang, Febrero 2012, pp. 441-444.
- [18] *XBee and XBee Pro OEM RF Module Antenna Considerations*, [Aplicación Note], Digi International Inc, 2012.
- [19] R. FALAUDI, *Building Wireless Sensors*, O'Reily, 2010.
- [20] J. E. GRAYSON, *Python and TkInter Programming*, Manning, 2000.
- [21] S. FERG (2014, Junio 25), *Thinking in TkInter*, [Archivo online] Disponible en: [http://www.ferg.org/thinking\\_in\\_tkinter.html](http://www.ferg.org/thinking_in_tkinter.html)
- [22] Raspberry Pi Foundation. (2013, Julio), [www.raspberrypi.org](http://www.raspberrypi.org)
- [23] Polar Electro OY. (2014, Junio 25), [www.polar.com](http://www.polar.com)
- [24] V. VIÑALS (2014, Junio 25), *Pulsómetros*, [Archivo online] Disponible en: [elmussol.files.wordpress.com/2010/06/pulsometros.pdf](http://elmussol.files.wordpress.com/2010/06/pulsometros.pdf)
- [25] (2014, Junio 25), *Pulsómetros*, [Texto online] Disponible en: [www.frecuencia-cardiaca.com/pulsometros.php](http://www.frecuencia-cardiaca.com/pulsometros.php)
- [26] Python Software Foundation. (2014, Junio 25), <http://www.python.org>
- [27] The Python Wiki (2014, Junio 25), *TkInter*, [Base de datos online] Disponible en: <https://wiki.python.org/moin/TkInter>
- [28] The Parallax Learn Sistem (2014, Junio 25), *Polar Heart Rate Receiver*, [Tutorial online] Disponible en: <http://learn.parallax.com/KickStart/28048>
- [29] ROGER MEIER (2014, Junio 25), *Roger Meier's Freeware*, Disponible en: <http://freeware.the-meiers.org>
- [30] Pulse Sensor. (2014, Agosto 19), <http://pulsesensor.myshopify.com>
- [31] WIKIPEDIA, MONITOR HOLTER (2014, Junio 28), [http://es.wikipedia.org/wiki/Monitor\\_Holter](http://es.wikipedia.org/wiki/Monitor_Holter)

- [32] WIKIPEDIA, ELECTROCARDIOGRAMA (2014, Junio 28), <http://es.wikipedia.org/wiki/Electrocardiograma>
- [33] FreeMotion Treadmill Basic FMTL82509 (2014, Agosto 20), *Free Motion*, [Tienda online] Disponible en: <http://www.freemotionfitness.com/fitness/en/Commercial-Fitness/treadmills/FreeMotion-Treadmill-Basic>



# Apéndice A

## Características del Sistema

	Min	Max	Unidades
<b>Dimensiones</b>			
Alto	71	73	milímetros
Largo	58	60	milímetros
Ancho	28	30	milímetros
<b>Distancia de transmisión</b>			
Radial (espacio abierto)	30	50	metros
Radial (espacio cerrado)	20	30	metros
<b>Distancia para evitar interferencia con otros pulsómetros</b>			
Polar sin codificar	1		metros
Polar codificado	Sin interferencia		
Otros sensores	1.5		metros
<b>Duración de la batería</b>			
Larga distancia	8	10	horas
Corta distancia	12	17	horas

Tabla A.1: Características del Sistema para el Atleta

	Min	Max	Unidades
<b>Dimensiones</b>			
Alto	66	69	milímetros
Largo	114	117	milímetros
Ancho	68	71	milímetros
<b>Duración de la batería</b>			
Larga distancia	5	6	horas
Corta distancia	6	7	horas
<b>Almacenamiento de datos</b>			
Por sesión (1 hora)		5,000	
Datos totales		104,000	

Tabla A.2: Características del Sistema para el Entrenador

## Apéndice B

# Programa para calcular la mediana de los tiempos obtenidos

```
def my_callback(channel):
    p = p + 1
    tiempo_actual = time.time()
    tiempo_entre_pulsos = tiempo_actual - tiempo_anterior
    tiempo_anterior = tiempo_actual
    if p == 1:
        tiempo_total1 = tiempo_total + tiempo_entre_pulsos
    if p == 2:
        tiempo_total2 = tiempo_total + tiempo_entre_pulsos
    if p == 3:
        tiempo_total3 = tiempo_total + tiempo_entre_pulsos
    if p == 4:
        tiempo_total4 = tiempo_total + tiempo_entre_pulsos
    if p == 5:
        tiempo_total5 = tiempo_total + tiempo_entre_pulsos

    lista_valores = [tiempo_total1, tiempo_total2, tiempo_total3, tiempo_total4, tiempo_total5]
    lista_valores.sort(reverse=True)
    mediana = lista_valores[len(lista_valores)/2]
    tiempo_total = mediana
```



## Apéndice C

# Especificaciones de la Raspberry Pi

**Product Name**                    **Raspberry Pi Model B**

---

**Product Description**        The Raspberry Pi is a small, powerful and lightweight ARM based computer which can do many of the things a desktop PC can do. The powerful graphics capabilities and HDMI video output make it ideal for multimedia applications such as media centres and narrowcasting solutions. The Raspberry Pi is based on a Broadcom BCM2835 chip. It does not feature a built-in hard disk or solid-state drive, instead relying on an SD card for booting and long-term storage.

**RS Part Number**                **756-8308**

---

**Specifications**

**Chip**                                    Broadcom BCM2835 SoC (a)

**Core architecture**                ARM11

**CPU**                                    700 MHz Low Power ARM1176JZFS Applications Processor

**GPU**                                    Dual Core VideoCore IV® Multimedia Co-Processor  
Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode  
Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure

**Memory**                                512MB SDRAM

**Operating System**                Boots from SD card, running a version of the Linux operating system

**Dimensions**                        85.6 x 53.98 x 17mm

**Power**                                    Micro USB socket 5V, 1.2A (l)

**Connectors:**

**Ethernet**

10/100 BaseT Ethernet socket (b)

**Video Output**

HDMI (rev 1.3 & 1.4) (c);  
Composite RCA (PAL and NTSC) (d)

**Audio Output**

3.5mm jack (e), HDMI

**USB 2.0**

Dual USB Connector (f)

**GPIO Connector**

26-pin 2.54 mm (100 mil) expansion header: 2x13 strip. Providing 8 GPIO pins plus access to I<sup>2</sup>C, SPI and UART as well as +3.3 V, +5 V and GND supply lines (g)

**Camera Connector**

15-pin MIPI Camera Serial Interface (CSI-2) (h)

**JTAG**

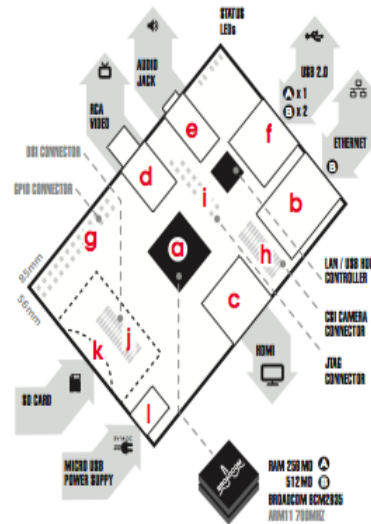
Not populated (i)

**Display Connector**

Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane (j)

**Memory Card Slot**

SDIO (k)





# Apéndice D

## Especificaciones de los módulos XBee

**Table 1-01. Specifications of the XBee®/XBee-PRO® RF Modules**

Specification	XBee	XBee-PRO
<b>Performance</b>		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) international variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for international variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
<b>Power Requirements</b>		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 $\mu$ A	< 10 $\mu$ A
<b>General</b>		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (Industrial)	-40 to 85° C (Industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
<b>Networking &amp; Security</b>		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
<b>Agency Approvals</b>		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	R201WW07215214	R201WW08215111 (Max. 10 dBm transmit power output)*
Australia	C-Tick	C-Tick