

Seguridad en vehículos de dos ruedas

Valentín Sánchez Ramírez

Proyecto Fin de Carrera Ingeniería de Telecomunicaciones

Dpto. de Electronica y Tecnología de Computadores

Escuela Técnica Superior de Ingeniería Informática y Telecomunicaicones

ETSIIT

Universidad de Granada UGR

5 de abril de 2015

Dedicado a

Índice general

Índice de figuras	IV
Índice de cuadros	VI
Resumen	VIII
Abstract	X
1. Introducción	1
1.1. Motivación	1
1.2. Sensores	2
1.3. Objetivos	2
2. Estado del arte	5
2.1. Comparación de módulos inalámbricos	5
2.1.1. Bluetooth RN-42	5
2.1.2. Bluetooth HC-05	6
2.2. Mota Sensora	6
2.3. Plataforma electrónica de código abierto	7
2.4. Dispositivo para la app	8
2.5. Sistemas de seguridad actuales y tendencias futuras	9
3. Desarrollo del Proyecto	12
3.1. Hardware	12
3.1.1. Arduino Uno R3	12
3.1.2. Pololu minimu 9 v.2	13
3.1.3. Bluetooth HC-05	13
3.2. Firmware	14
3.2.1. Configuración Arduino Uno y Pololu Minimu 9 V.2	14
3.2.2. Configuración Arduino Uno y Bluetooth HC-05	16
3.2.3. Schematic conexión módulos y placa Arduino	17
3.3. Software	17
3.3.1. Android	18
3.3.2. Diagrama estados	19
3.3.3. Interfaz App	21
4. Resultados	23
4.1. Resultados Hardware	23
4.2. Resultados Software	25
4.3. Resultados App Android	25
4.3.1. Bluetooth desactivado	25

4.3.2. GPS desactivado	25
4.3.3. No se detecta accidente	25
4.3.4. Posibilidad de accidente	25
4.3.5. Accidente y aviso a emergencias	25
4.4. Prototipo	25
5. Conclusiones	28
Bibliografa	30
Anexo 1	32
Anexo 2	34

Índice de figuras

2.1. Módulo Bluetooth RN-42	5
2.2. Módulo Bluetooth HC-05	6
2.3. Pololu minimu-9 v2	7
2.4. Arduino Uno R3	7
2.5. Comparativa Sistema Operativos en uso	8
2.6. Airbag en casco y cazadora	9
3.1. ATmega328 en Arduino Uno R3	13
3.2. Schematic Pololu Minimu 9 V.2	14
3.3. Schematic Bluetooth HC-05	15
3.4. Conexiones Motasonra y Arduino	16
3.5. Schematic módulos y Arduino conectados	17
3.6. Resultado conexión todos los componentes	18
3.7. Estructura app Android	19
3.8. Diagrama de Estados	20
3.9. Interfaz app Android	22
4.1. Theta teórico 0	23
4.2. Theta teórico 90	24
4.3. Diseño PCB	26
4.4. Prototipo proyecto	27

Índice de cuadros

2.1. Porcentaje mundial de sistemas operativos en Smartphones	8
3.1. Librerías cargadas en Arduino para usar la motasensora	15
3.2. Librerías cargadas en Arduino para usar Bluetooth HC-05	16

Resumen

A la hora de usar un vehículo de dos ruedas lo mas importante es nuestra seguridad, prueba de ello son los sistemas que se están implantando tanto en el propio vehículo como en cascos o chaquetas. En el caso de sufrir un accidente el tiempo juega en nuestra contra, una rápida actuación de los servicios sanitarios puede salvarnos la vida en caso de que lo necesitemos.

Al año se producen mas de 2000 accidentes de moto, el 72 % de esos accidentes precisa atención médica. Alguno vehículos de alta gama ya poseen un sistema de seguridad en caso de sufrir en accidente, sistema que será obligatorio en Espaa a partir del 31 de Marzo de 2018 en coches. Quedando disponible realizar un sistema parecido para motos.

Por ello este proyecto se compone de un sistema electrónico con sensores comunicado mediante Bluetooth con una app móvil encargada de avisar a los servicios sanitarios si hemos sufrido un accidente para que estos puedan actuar en el menor tiempo posible.

Palabras clave: seguridad en vehiculos de dos ruedas, aviso a emergencias, seguridad en motos, sensor, app móvil, bluetooth, smartphone.

Abstract

The most important thing when you are using a two-wheeler is our safety, the proof is the systems that are being implemented in the vehicle itself and on helmets or jackets. In the case of an accident time plays against us, act swiftly of health services can save our lives if we need it.

Per year over 2000 motorcycle accidents happens, 72 % of those accidents needs medical attention. Some luxury cars already have a security system should suffer in an accident and which shall be binding in Spain from March 31, 2018. Being available to perform a similar system for motorcycles.

Therefore, this project consists of an electronic system with sensors connected via Bluetooth to a mobile app which will notify health services if we had an accident, then they can act in the shortest time possible.

Keywords: safety two-wheeled vehicles, emergency notification, motorcycle safety, sensor, mobile app, bluetooth, smartphone.

Capítulo 1

Introducción

En este capítulo pretendemos situar al lector en el contexto del proyecto, en que campo vamos a trabajar, motivación y los objetivos del proyecto para que se pueda entender con mayor facilidad el trabajo realizado y los objetivos a conseguir.

1.1. Motivación

Desde la invención de la motocicleta por Sylvester Howard Roper en 1867, el mundo del motor ha evolucionado desde el motor de cilindros a vapor hasta los motores actuales. Diferentes tipos de motocicletas con distintas cilindradas inundan nuestras calles a día de hoy.

Como podemos ver al salir a la calle las motocicletas no son el único vehículo que nos encontramos, también podemos encontrar coches, camiones y sus variantes en forma, motor, tamaño y número de ruedas. Lo que si podemos observar es que la moticleta es el vehículo menos seguro frente a los mencionados, no solo por dimensiones sino porque en caso de accidente es el cuerpo del motorista quien sufre el impacto y no la carrocería como en los otros vehículos.

Con el paso de los años la seguridad en vehículos con motor ha aumentado progresivamente. Este crecimiento ha sido mayor con la introducción de nuevas tecnologías, tales como sensores, dotan al vehículo de cierta inteligencia.

Hoy en día es difícil encontrar una persona que no posea un smartphone, una herramienta que no solo sirve para llamar, leer la prensa o encontrar una ubicación, sino que con la posibilidad del desarrollo de apps podemos convertir nuestro smartphone en una herramienta que nos ayude en cualquier tarea, circunstancia o incluso que nos salve la vida.

La combinación de un vehículo con sensores y comunicacion Bluetooth con un smartphone es el punto clave para que este proyecto pueda ser llevado a cabo.

Por ello podemos decir que los elementos clave para aumentar la seguridad en un vehículo de dos ruedas en este proyecto serían:

- Sensores.
- Bluetooth.
- Smartphone.

1.2. Sensores

En este proyecto se usará una mota sensora Pololu MinIMU-9 v2 que se compone de tres sensores:

- Acelerómetro: es el encargado de medir aceleraciones, mide la aceleración asociada con el fenómeno de peso experimentado por una masa de prueba que se encuentra en el marco de referencia del dispositivo.
- Giróscopo: es el encargado de medir, mantener o cambiar la orientación en el espacio de algún aparato o vehículo. Está formado por un cuerpo con simetría de rotación que gira alrededor del eje de dicha simetría.
- Magnetómetro: es el encargado de medir en fuerza o dirección la señal magnética de una muestra.

1.3. Objetivos

En este proyecto, se pretende entrar en contacto con el diseño y programación de una placa Arduino UNO, la interpretación de los datos que proporciona el Pololu MinIMU-9 v2, establecer la comunicación Bluetooth entre la placa Arduino UNO y un dispositivo Android y la programación y diseño de la app móvil en Android.

Para ello conectaremos el Pololu MinIMU-9 v2 y el dispositivo Bluetooth HC-05 a la placa Arduino UNO, programaremos nuestra placa Arduino para que reciba los datos del sensor y los envíe mediante Bluetooth al dispositivo Android.

Posteriormente debemos programar una app Android en nuestro Smartphone que sea capaz de conectarse al dispositivo HC-05 y leer la información recibida, esta app será la encargada de ejecutar el algoritmo y comprobaciones pertinentes. En caso de haber sufrido un accidente se le notificara al número de emergencias 112 de nuestra ubicación via SMS.

Para llevar a cabo este proyecto, el trabajo a realizar se divide en los siguientes objetivos:

- Instalación de entorno Arduino y familiarizarse con dicha programación. Estudio de la programación Arduino para leer los datos recibidos de la mota sensora y enviarlos via Bluetooth.
- Estudio de la bibliografía acerca de la calibración de los sensores y de los algoritmos de cálculo de Pitch, Roll y Yaw. Se estudiará el funcionamiento de las motas y como deben ser programadas.
- Montaje de la mota sensora y HC-05. Montar la mota sensora y el módulo HC-05 y tenerlos listos para su uso, realización de programas básicos de prueba y verificación de que funcionan correctamente.
- Estudio de programación Android en Eclipse. Con la correspondiente app que ejecute nuestro algoritmo de detección de accidente y en caso necesario enviar la ubicación actual a Emergencias 112 via SMS.
- Pruebas del sistema en diferentes condiciones. Realización de las pertinentes pruebas para comprobar el correcto funcionamiento del sistema desarrollado.

- Desarrollo de PCB's. Desarrollo de una placa PCB para evitar el uso de cables en el proyecto y hacerlo mas compacto.

Capítulo 2

Estado del arte

En este capítulo vamos a realizar un estudio de la situación tecnológica actual en todo lo referente a la comunicación inalámbrica, sensor utilizado, plataforma electrónica de código abierto, dispositivo para el que se desarrollará la app, sistemas de seguridad actuales en vehículos de dos ruedas y las tendencias futuras.

2.1. Comparación de módulos inalámbricos

Buscamos un protocolo inalámbrico de comunicación que nos permita enviar los datos con la mayor exactitud posible, con un consumo de energía razonable y de forma segura. Para ello estudiaremos los módulos Bluetooth RN-42 y HC-05.

2.1.1. Bluetooth RN-42

Este módulo Bluetooth [6] está diseñado para reemplazar los cables de serie. Está completamente encapsulado, el usuario solo ve los caracteres de serie se transmiten hacia atrás y adelante. Este dispositivo es usado para corto alcance, con un consumo de 26 μ A en reposo. Fácil de integrar en sistemas embebidos y de conectar a dispositivos ya existentes

- Soporte para: BCSP, LAN, GAP, SDP, RFCOMM y L2CAP
- Velocidad UART: hasta 3 Mbps
- Alcance: 15-18 m



Figura 2.1: Módulo Bluetooth RN-42

2.1.2. Bluetooth HC-05

Este módulo Bluetooth [4] ofrece una mejor relación precio frente a prestaciones, es un módulo maestro esclavo, no solo recibe conexiones sino que también las genera hacia otros dispositivos Bluetooth. Posee la versión V2.0+EDR, trabajando a una frecuencia de 2,4 GHz en la banda ISM, con una modulación GFSK.

- Soporta comando AT para ser configurado
- Velocidad: hasta 2,1 Mbps y Síncrono 1Mbps/1Mbps
- Alcance: 10 m



Figura 2.2: Módulo Bluetooth HC-05

Estudiados ambos módulos se decidió usar el módulo Bluetooth HC-05 no solo por ser mas económico sino porque es un dispositivo que puede seguir trabajando en un rango de temperaturas de -20°C hasta 75°C y teniendo en cuenta que este sistema estará ubicado en el interior de la moto es posible alcanzar dichas temperaturas. Además las características del módulo HC-05 son suficientes para la realización de este proyecto.

2.2. Mota Sensora

La Mota que se va a usar para la realización de este proyecto es el Pololu minimu-9 v2 [5], la cual es una unidad de medición inercial (IMU), que viene equipada con un L3GD20 giroscopio de 3 ejes y un LSM303DLHC con 3 ejes para el acelerómetro y 3 ejes para el magnetómetro. El módulo incluye un regulador de voltaje y un circuito de desplazamiento que permite el funcionamiento de 2,5 a 5,5 V.

Las especificaciones de la mota son las siguientes:

- Dimensiones: 20 x 13 x 3 mm
- Peso: 0,7 g
- Alimentación: 10 mA
- Giróscopo: lectura de 16 bits por eje
- Acelerómetro: lectura de 12 bits por eje
- Magnetómetro: lectura de 12 bits por eje
- Rango de sensibilidad configurable



Figura 2.3: Pololu minimu-9 v2

2.3. Plataforma electrónica de código abierto

La plataforma electrónica de código abierto que vamos a utilizar es Arduino Uno R3 [1], placa electrónica basada en el ATmega 328. Cuenta con 14 pines digitales de entrada/salida, 6 entradas analógicas, un resonador cerámico de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reinicio. Basta con conectarlo a un ordenador con un cable USB o a una batería para empezar.

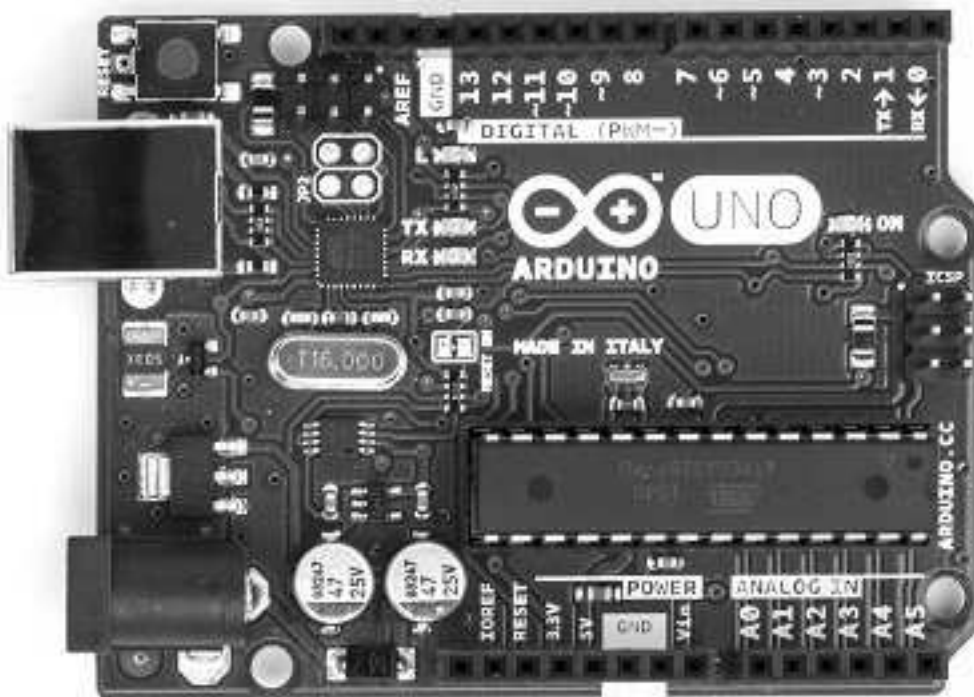


Figura 2.4: Arduino Uno R3

Arduino Uno puede ser alimentado con 6 hasta 20 voltios. Si se alimenta con menos de 7 voltios, el pin encargado de suministrar 5 voltios es muy probable que suministre menos. En caso de suministrar mas de 12 voltios, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango de alimentación recomendado es de 7 a 12 voltios.

Para programar Arduino Uno se puede usar el software de Arduino. Software desde el que podremos cargar nuestros programas en la placa Arduino a través de un cable USB, Arduino Uno cuenta con una memoria de 2KB de SRAM y 1 KB de EEPROM.

2.4. Dispositivo para la app

A la hora de realizar la aplicación que procesará los datos recibidos por el Bluetooth y ejecutará el algoritmo de accidente estudiamos cual es la situación del mercado en lo que respecta al sistema operativo de los dispositivos en uso.

Según una noticia publicada por xataka en 2014 [7] podemos ver los datos proporcionados por la consultora IDC, resultados que se muestran a continuación.

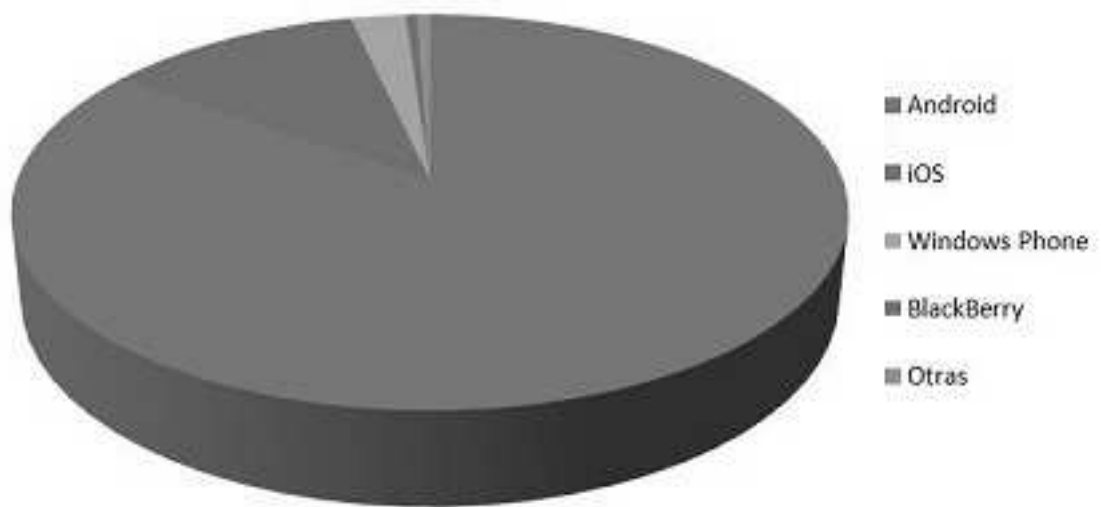


Figura 2.5: Comparativa Sistema Operativos en uso

En la siguiente tabla mostramos los porcentajes de los dispositivos que usan cada uno de los sistemas operativos disponibles en el mercado.

Sistema Operativo	% Uso mundial
Android	84.7
iOS	11.7
Windows Phone	2.5
BlackBerry	0.5
Otros	0.6

Cuadro 2.1: Porcentaje mundial de sistemas operativos en Smartphones

Viendo estos datos se decidió realizar la aplicación para un dispositivo Android.

2.5. Sistemas de seguridad actuales y tendencias futuras

La tecnología comienza a subirse a la moto, la comunicación entre vehículos y carreteras contribuirá a la prevención de accidentes y disminución de la siniestralidad de uno de los colectivos más vulnerables.

La implantación de sistemas de seguridad como el cinturón de seguridad, airbag, ABS, entre otros muchos han sido incorporados de forma tardía en la moto. La década en la que nos encontramos se centrará en la moto, los sistemas inteligentes van a tener un papel especial.

En materia de seguridad activa, la que ayuda a evitar accidentes hay que destacar los sistemas de ayuda en la frenada, además de la implantación de mejoras en materia de iluminación, amortiguación, estabilidad, sistemas de cambios, neumáticos... En prueba se encuentran algunos sistemas novedosos basados en la comunicación entre vehículos y con las infraestructuras para advertir sobre situaciones peligrosas como tráfico, climatología adversa...

En los últimos dos años se ha iniciado una revolución electrónica en la motocicleta, aportando una gran seguridad como es el ABS, control de tracción. En 2017 la UE (Union Europea) trabaja para que las motos salgan con el ABS de fábrica.

En materia de seguridad pasiva nos encontramos elementos como el casco, la cabeza sufre traumatismos en el 50 % de los accidentes. Un casco adecuado puede reducir estas lesiones en un 69 % y la mortalidad en un 42 %. Además de ropa ligera que protege en las caídas o incluso cazadoras, chalecos y cascos con airbag en caso de caída.



Figura 2.6: Airbag en casco y cazadora

Toda esta información ha sido recopilada de un reportaje de la DGT [3].

Tal y como hemos podido leer todos estos sistemas en materia de seguridad activa y pasiva son para prever accidentes o reducir los daños sufridos en caso de accidente. Si el motorista ha sufrido un accidente el tiempo juega en su contra, por lo que se requiere una rápida intervención de los servicios sanitarios.

Un accidente puede ocurrir en un lugar concurrido y cualquier viandante que se encuentre allí puede realizar dicha llamada a emergencias, el inconveniente sería que el accidente ocurra en una vía poco transitada y el conductor no pueda realizar dicha llamada, esperar a que otro conductor circule por esa misma vía puede llevar demasiado tiempo.

Como consecuencia nace la realización de este proyecto, cuya app he llamado MotoSafe. Consiste en que el propio smartphone del conductor pueda avisar a emergencias indicándoles nuestra posición en caso que la app haya detectado que se ha producido un accidente a partir de los datos suministrados por el sistema electrónico que ubicaremos en la moto.

En el próximo capítulo se proporcionarán mas detalles acerca del desarrollo de este proyecto.

Capítulo 3

Desarrollo del Proyecto

En este capítulo se va a explicar de forma más detallada las características y el funcionamiento de todos los elementos de los que se hará uso para llevar a cabo el desarrollo de este proyecto. Además se detallará el desarrollo hardware, firmware y software, explicando con detalle la configuración electrónica y se presentará la solución software implementada.

3.1. Hardware

Los dispositivos hardware que vamos a usar y sus configuraciones son los siguientes.

3.1.1. Arduino Uno R3

Arduino Uno R3 [1], placa electrónica basada en el microcontrolador ATmega 328. Cuenta con 14 pines digitales de entrada/salida, 6 entradas analógicas, un resonador cerámico de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reinicio. Basta con conectarlo a un ordenador con un cable USB o a una batería para empezar.

Arduino Uno puede ser alimentado con 6 hasta 20 voltios. Si se alimenta con menos de 7 voltios, el pin encargado de suministrar 5 voltios es muy probable que suministre menos. En caso de suministrar mas de 12 voltios, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango de alimentación recomendado es de 7 a 12 voltios.

Para programar Arduino Uno se puede usar el software de Arduino. Software desde el que podremos cargar nuestros programas en la placa Arduino a través de un cable USB, Arduino Uno cuenta con una memoria de 2KB de SRAM y 1 KB de EEPROM.

ATmega328 [2] es un microcontrolador creado por Atmel y que pertenece a la serie megaAVR. ATmega328 es un circuito integrado de alto rendimiento que está basado en un microcontrolador RISC, combinando 32 KB ISP flash de memoria con la capacidad de leer y escribir, 1 KB de memoria EEPROM, 2 KB de SRAM, 23 líneas de E/S de propósito general, 32 registros de proceso general, tres temporizadores contadores con modo de comparación, interrupciones internas y externas, programador de modo USART, una interfaz serial orientada a byte de 2 cables, SPI puerto serial, 6 canales 10 bit conversor A/D y cinco modos de ahorro de energía seleccionables por software. Opera entre 1,8 y 5,5 voltios. Este dispositivo alcanza una respuesta de 1 MIPS, balanceando consumo de energía y velocidad de proceso.

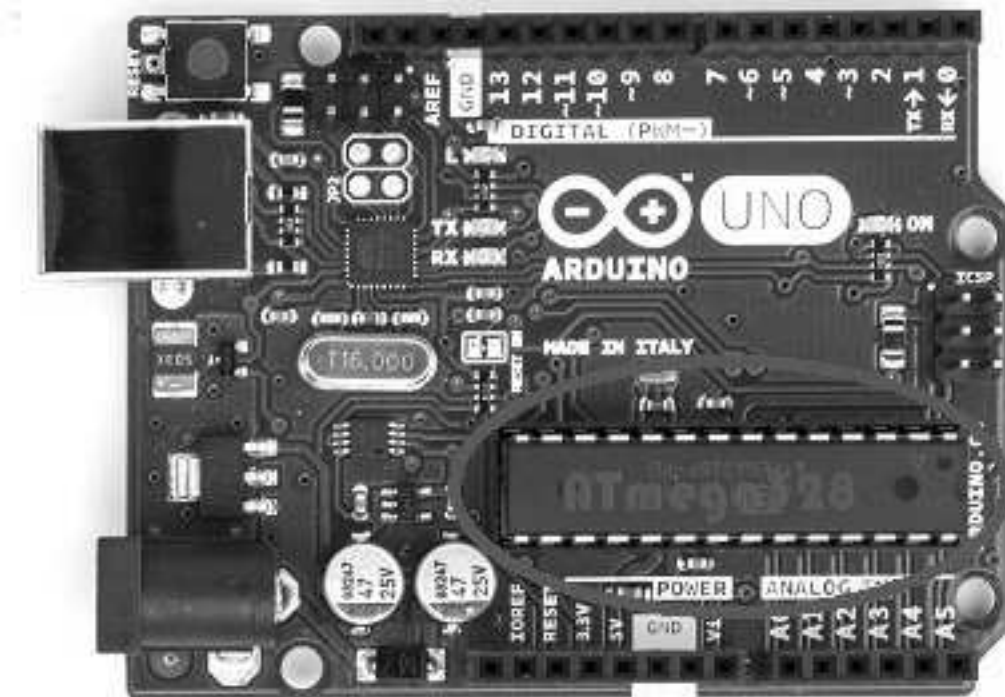


Figura 3.1: ATmega328 en Arduino Uno R3

3.1.2. Pololu minimu 9 v.2

Pololu minimu-9 v2 [5], la cual es una unidad de medición inercial (IMU), que viene equipada con un L3GD20 giroscopio de 3 ejes y un LSM303DLHC con 3 ejes para el acelerómetro y 3 ejes para el magnetómetro. El módulo incluye un regulador de voltaje y un circuito de desplazamiento que permite el funcionamiento de 2,5 a 5,5 V.

Cuyo Schematic se puede ver la figura 3.2.

3.1.3. Bluetooth HC-05

Este módulo Bluetooth [4] ofrece una mejor relación precio frente a prestaciones, es un módulo maestro esclavo, no solo recibe conexiones sino que también las genera hacia otros dispositivos Bluetooth. Posee la versión V2.0+EDR, trabajando a una frecuencia de 2,4 GHz en la banda ISM, con una modulación GFSK.

- Soporta comando AT para ser configurado
- Velocidad: hasta 2,1 Mbps y Síncrono 1Mbps/1Mbps
- Alcance: 10 metros

El alcance teórico es de 10 metros, pero el alcance práctico es de 8,5 metros aproximadamente, medidos en un espacio abierto sin inclemencias meteorológicas.

El Schematic del dispositivo HC-05 podemos verlo en la figura 3.3.

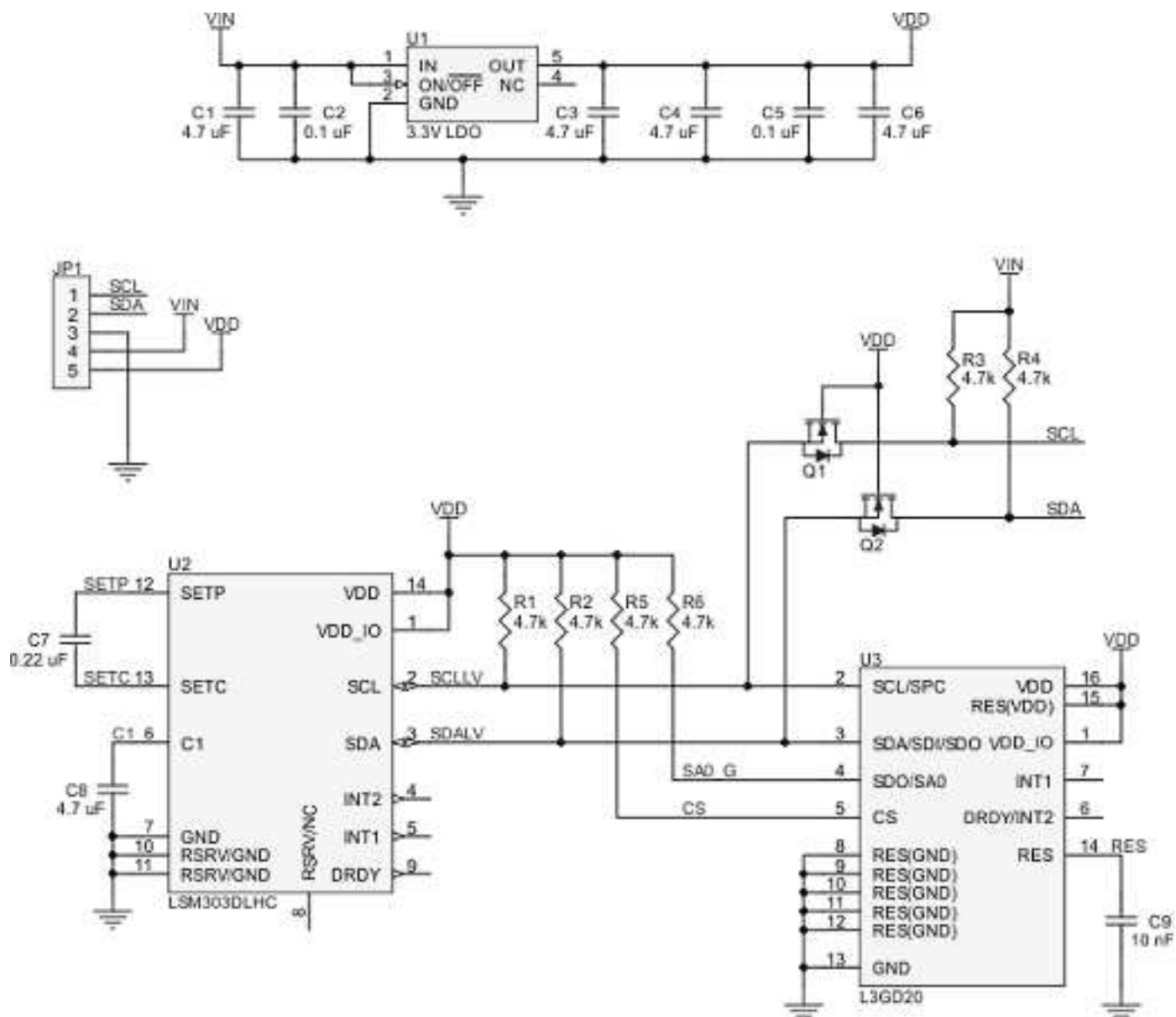


Figura 3.2: Schematic Pololu Minimu 9 V.2

3.2. Firmware

En los que respecta al firmware vamos a detallar como hemos configurado cada uno de los dispositivos hardware y conectado entre sí. Mostrando al final un Schematic con todos los componentes conectados y su implementación sobre una protoboax.

3.2.1. Configuración Arduino Uno y Pololu Minimu 9 V.2

El uso de este sensor en la placa Arduino requería añadir dos librerías en el software que posteriormente cargaríamos en la placa arduino.

Las dos librerías usadas son:

respectivamente de la placa Arduino, además de VCC y GND entre la motasensora y la placa.

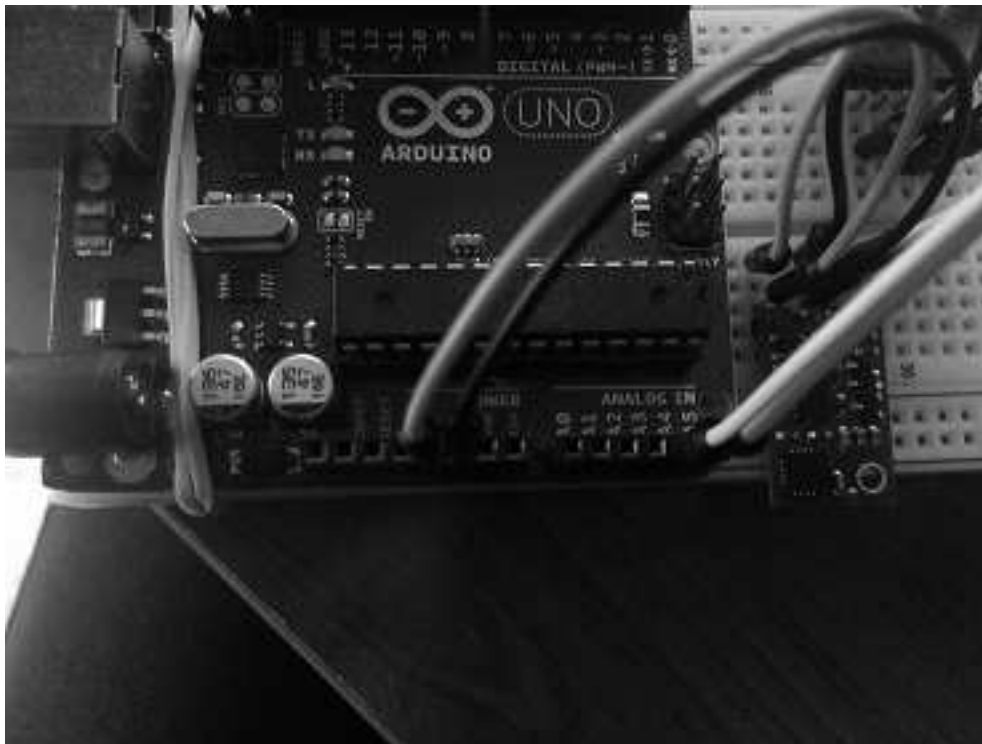


Figura 3.4: Conexiones Motasonra y Arduino

En lo que respecta a la configuración de la moto sensora, debemos fijarle una posición inicial. Que será con respecto a la cual calculemos los ángulos de inclinación con respecto a la vertical de la moto. Al encender este sistema electrónico se produce un calibrado de los angulos iniciales.

3.2.2. Configuración Arduino Uno y Bluetooth HC-05

El uso del dispositivo bluetooth HC-05 en la placa Arduino requería añadir dos librerías en el software que posteriormente cargaríamos en la placa arduino.

Las librerías usadas son:

#include	SoftwareSerial.h
#include	wire.h

Cuadro 3.2: Librerías cargadas en Arduino para usar Bluetooth HC-05

En nuestro código Arduino debemos dejar indicado que el pin RXD es el 10 y el pin TDX es el 11 con el siguiente comando:

```
SoftwareSerial BT = SoftwareSerial(10, 11); //10 RX, 11 TX.
```

Además debemos configurar el Arduino para que el pin 10 sea de entrada y el pin 11 de salida, indicando en el PinMode si es InPut o OutPut respectivamente. Ambos componentes se deben comunicar a 9600 baudios, velocidad por defecto de funcionamiento del bluetooth HC-05, una configuración a una velocidad diferente provoca que no haya comunicación entre el dispositivo bluetooth y la placa Arduino.

En lo que respecta a las conexiones, la patilla RXD y TXD del módulo bluetooth se conectarán a las patillas 11 y 12 respectivamente de la placa Arduino, además de VCC y GND entre el módulo bluetooth y la placa.

En los que respecta al método loop() de nuestro código Arduino lo que haremos será comprobar que el módulo bluetooth HC-05 se encuentra conectado a un dispositivo móvil, si la respuesta es negativa permaneceremos a la espera de conexión, en caso de ser afirmativa la respuesta enviaremos por dicho canal de comunicación los datos medidos en la motasensora.

Dichos datos serán enviados cada 500 milisegundos, periodo de envío que he establecido para no saturar al receptor.

3.2.3. Schematic conexión módulos y placa Arduino

En el Schematic que se mostrará a continuación podemos ver de forma fácil y rápida todas las conexiones que se han implementado.

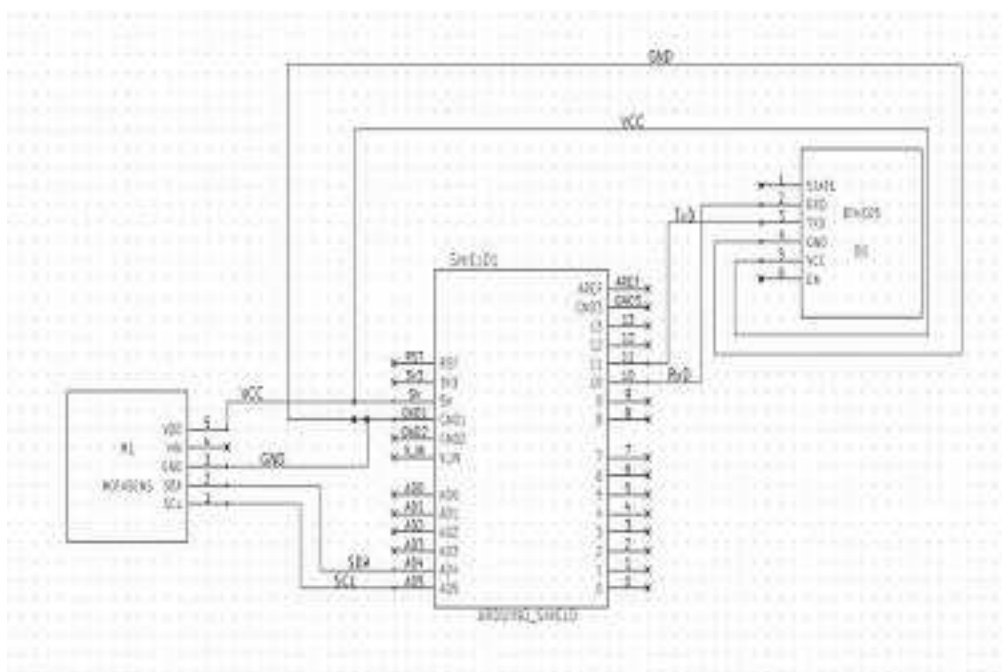


Figura 3.5: Schematic módulos y Arduino conectados

El módulo del centro representa a la placa Arduino, mientras que el módulo a la izquierda de la imagen representa la morasensora y el módulo de la derecha al dispositivo bluetooth HC-05.

Una vez esto ha sido implemetado y conectado en la placa Arduino con la ayuda de una protoboax, obtenemos el siguiente resultado mostrado en la figura 3.6.

3.3. Software

En esta sección procederé a explicar en detalle el desarrollo de la app Android, así como el diagrama de estados del algoritmo implementado y la interfaz eventual para comprobar el funcionamiento correcto de la app.

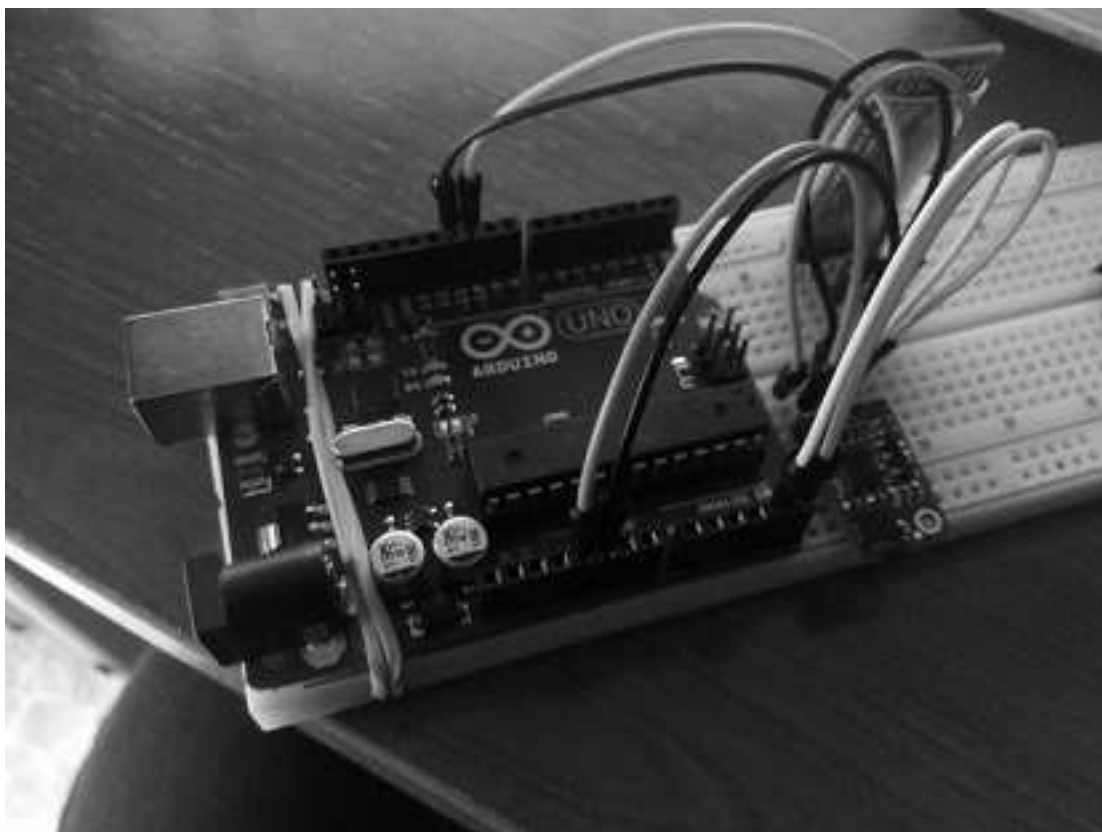


Figura 3.6: Resultado conexión todos los componentes

3.3.1. Android

Para el desarrollo de ésta app hemos descargado el programa Eclipse ADT de Android Developer. Programa que he usado para el desarrollo íntegro de la app MotoSafe. Se debe prestar atención a partir de que versión de Android deseamos implementar la app, escogí la versión Android 4.4 ya que es la que actualmente posee mi smartphone, para una posterior comercialización debo hacer ésta app disponible a partir de la version Android 2.3 ya que la mayoría de los smartphones poseen ésta versión.

En su implementación se ha usado una única clase, que es la clase main, por defecto invocada al ejecutarse el programa y que contiene el bucle que realiza todas las acciones necesarias para que el programa funcione correctamente, tal y como se muestra en la figura 3.7.

Aqui debo destacar que en el fichero main no solo contengo la clase main, sino también la clase RecibirComando, que es la encargada de enviar por el Handler toda la información recibida del bluetooth.

En resumen, la clase main será la encargada de gestionar todos los datos recibidos por el Handler, comprobar el estado del GPS, Bluetooth y si ambos dispositivos se encuentran conectados o se pierde la conexión. Además será la clase encargada de ejecutar el algoritmo y en caso necesario registrar la ubicación para posteriormente enviarsela vía SMS a emergencias.

En los próximos apartados se explicará con mas detalle el funcionamiento de la app y se mostrará el diagrama de estados.

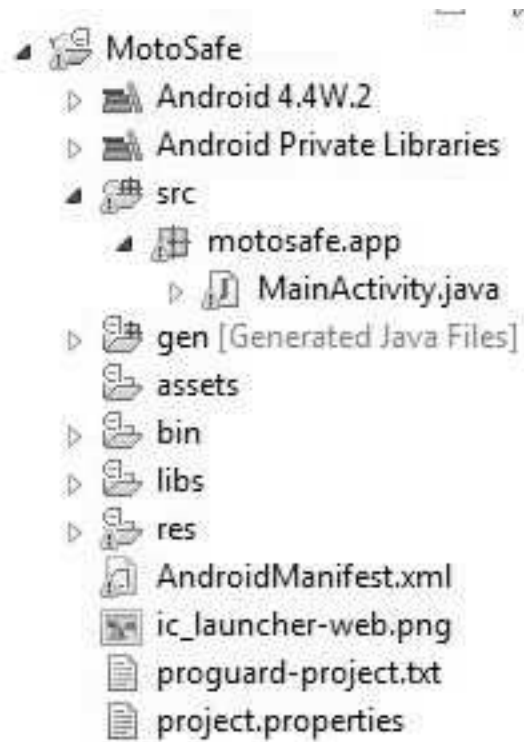


Figura 3.7: Estructura app Android

3.3.2. Diagrama estados

En este subapartado se va a explicar el funcionamiento de la app además de un diagrama de estados representativo.

A la hora de empezar a usar la app MotoSafe debemos entrar en la configuración del bluetooth de nuestro smartphone y emparejarlo con el dispositivo HC-05, la clave por defecto para su emparejamiento es "1234". Para poder realizar dicha emparejamiento debemos tener en cuenta que el módulo bluetooth se debe encontrar encendido, para ello la motocicleta se debe encontrar arrancada o con el contacto encendido. Una vez hemos realizado dicho proceso ya podemos abrir esta app y comenzar a usarla.

Al abrir la app MotoSafe encontraremos la interfaz que mostramos en la figura 3.9. Si pulsamos sobre el botón OFF la aplicación se cerrará y no continuará ejecutándose en segundo plano, es decir, mata este proceso. Si por el contrario pulsamos el botón ON, se iniciarán una serie de procesos que harán funcionar correctamente la app.

Lo primero que hará será comprobar el estado del bluetooth de nuestro smartphone, en el caso de tenerlo desactivado un alert nos informa de esto y nos ofrece de posibilidad de activarlo, en caso de negarnos, la app se cerrará automáticamente. El siguiente paso es comprobar el estado del GPS, en el caso de tenerlo desactivado un alert nos informa de esto y nos ofrece de posibilidad de activarlo, en caso de negarnos, la app se cerrará automáticamente.

Posteriormente y de forma automática se creará un Handler que será el encargado de conectar la app con el bluetooth y poder interpretar todos los datos que recibe. En el caso de no estar emparejado el dispositivo no recibiremos datos y por tanto la app no ejecutará el algoritmo implementado.

Una vez todo está activado, la app está lista para funcionar, para ello lo primero que hará será leer los datos recibidos por el Handler e interpretarlos. Nuestro algoritmo recibirá los datos ya in-

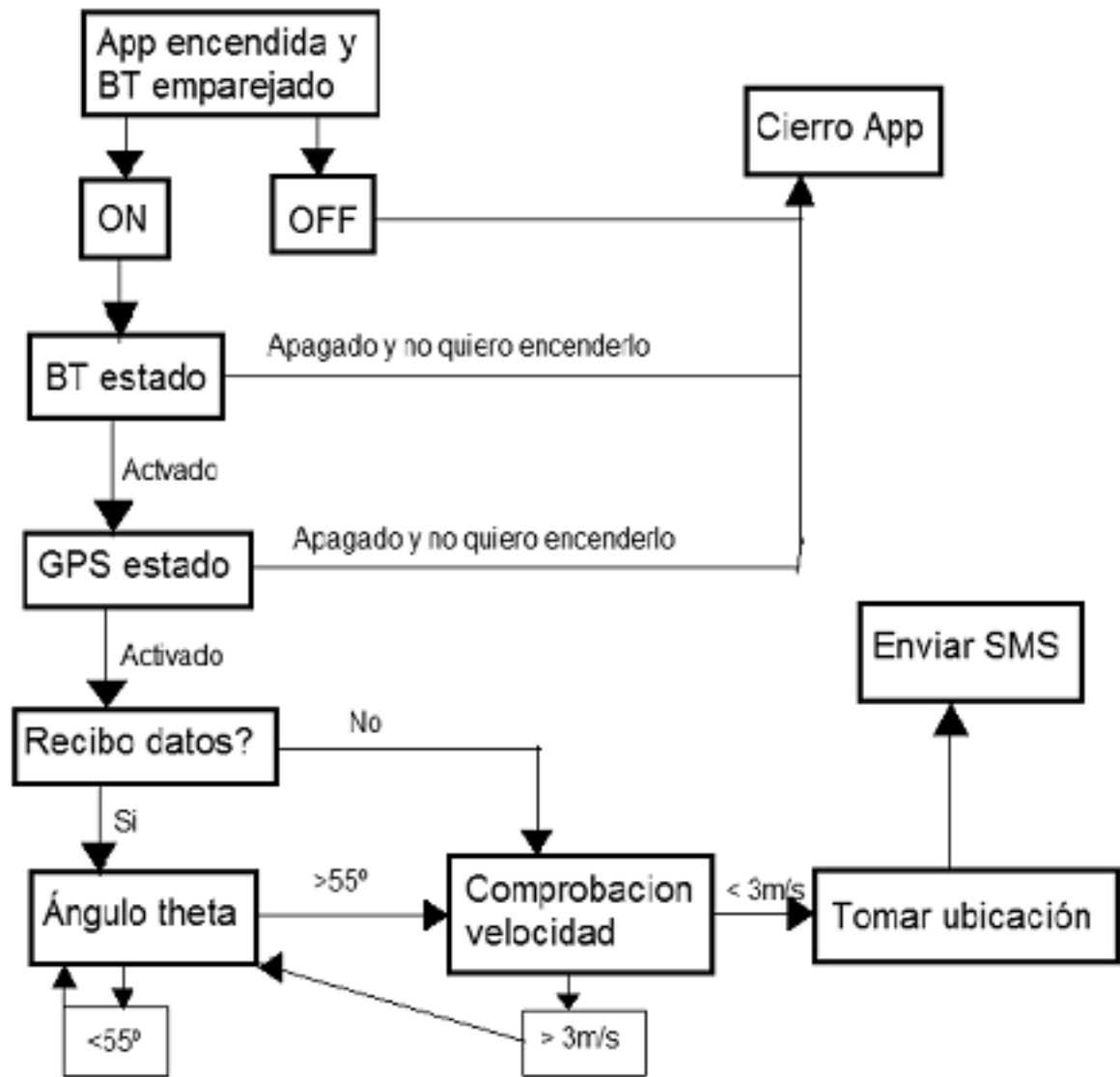


Figura 3.8: Diagrama de Estados

terpretados, con ellos procederá a calcular el ángulo de inclinación de la motocicleta. Si este ángulo es inferior a 55 grados volvemos a hacer los cálculos con los siguientes datos recibidos. Este proceso se realiza de forma periódica cada 550 milisegundos. En el caso de que éste ángulo sea superior a 55 grados procedemos a comprobar la velocidad de la motocicleta.

Para comprobar la velocidad de la motocicleta se usará el GPS del smartphone, por lo que en este momento se procederá al uso del GPS, para ahorrar batería no lo tendremos encendido durante todo el transcurso del desplazamiento. Si la velocidad medida es mayor a 3 metros por segundo, volvemos al principio del algoritmo y volvemos a calcular el ángulo de inclinación de la motocicleta, en caso de que la velocidad sea menor a 3 metros por segundo procedemos a tomar a la ubicación según las coordenadas de latitud, longitud y precisión.

Estos datos obtenidos por el GPS serán los que enviemos via SMS al número de emergencias, indicándoles la posición actual en latitud, longitud y la precisión con que mide el GPS del smartphone.

En el caso de perder la señal y no recibir datos, el sistema entrará en un proceso de espera temporal, si no se recupera la recepción de datos transcurriendo este tiempo se procederá a calcular la velocidad y seguir con el diagrama de estados según la velocidad. Si durante el transcurso de

ese tiempo de espera se pulsa el botón OFF la aplicación se cerrará completamente.

3.3.3. Interfaz App

La interfaz eventual de nuestra app es la mostrada en la figura 3.9. En la cual se pueden apreciar dos botones, uno de encendido ON y otro de apagado OFF. Además disponemos de 4 TextView, los cuales usaremos para comprobar el estado actual de la app.

El primer TextView muestra el ángulo theta de inclinación, siendo su valor entre -90 y 90 grados con respecto a la vertical, dependiendo si estamos inclinados hacia la derecha o hacia la izquierda. El segundo TextView mostrará dos mensajes, uno de ellos "no me he caído" y el otro "es posible que me haya caído", si el mensaje mostrado es el segundo el tercer TextView mostrará "comprobando velocidad". Basándose en el algoritmo implementado, si se ha sufrido un accidente el cuarto y último TextView mostrará el siguiente mensaje "He sufrido un accidente y mi ubicación es: latitud 'x' longitud 'y' precisión 'z' " éste será el mensaje que se enviará vía SMS a emergencias, mostrando las coordenadas calculadas con el GPS. En caso de no haber sufrido accidente no se mostrará nada en este último TextView.



Figura 3.9: Interfaz app Android

Capítulo 4

Resultados

En este capítulo vamos a analizar los datos obtenidos de este proyecto, además de proporcionar una comparación entre los datos teóricos ofrecidos por el fabricante y los datos obtenidos tras su implementación.

4.1. Resultados Hardware

Dentro de Hardware debemos distinguir los dos componentes mas importantes que poseemos como son el dispositivo bluetooth HC-05 y la mota sensora.

En lo que respecta al dispositivo bluetooth HC-05 [4] como pude estudiar su alcance era de 10 metros, pero en un entorno abierto he podido comprobar que su alcance se reduce a unos 8 metros aproximadamente, a pesar de evitar el apantallamiento con el plano masa diseñado en la PCB.

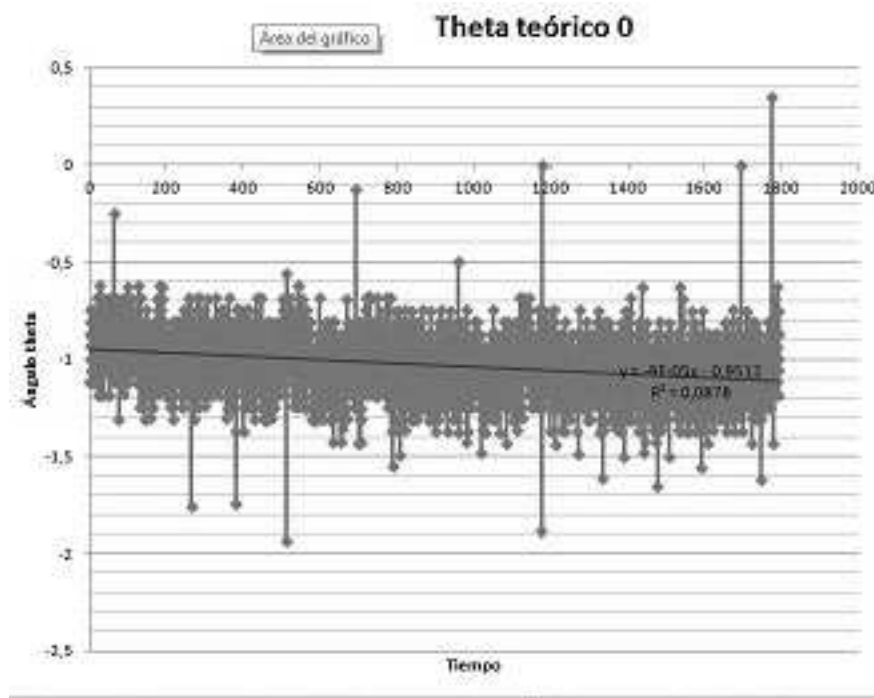


Figura 4.1: Theta teórico 0

Comprobamos el funcionamiento de la mota sensora, para ellos partimos de dos angulos iniciales y comprobamos la tendencia de las medidas del sensor transcurridos 30 minutos y sin variar la posición del dispositivo.

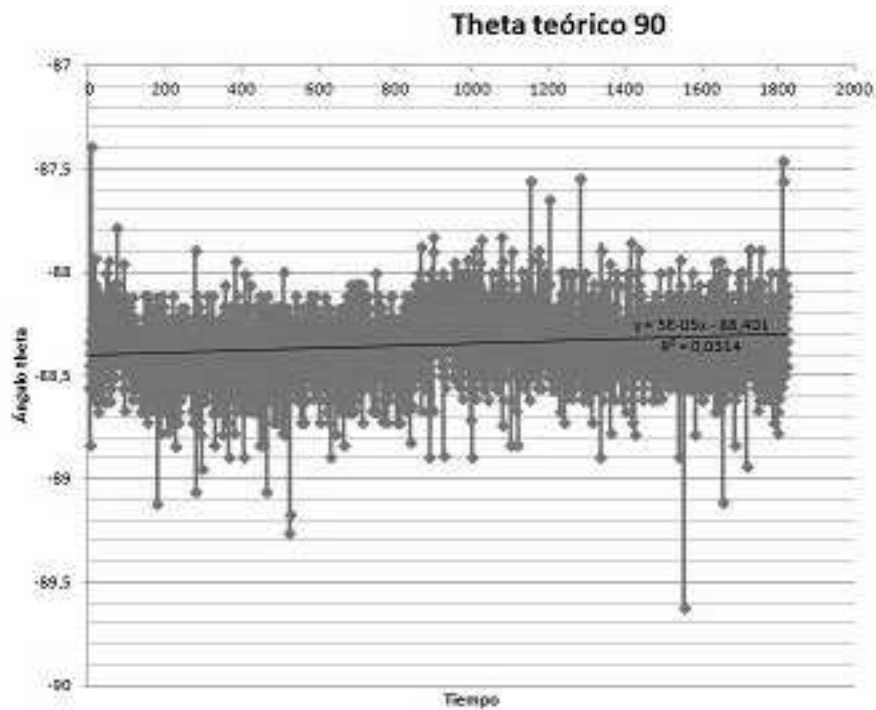


Figura 4.2: Theta teórico 90

Tal y como podemos observar en las fuguras 4.1 y 4.2 transcurridos 30 minutos el sensor tiende a variar 0.1 grados con respecto al ángulo inicial de partida. He de destacar que los ángulos teóricos y prácticos difieren debido al desnivel involuntario que posée la mesa donde realicé las pruebas.

4.2. Resultados Software

4.3. Resultados App Android

4.3.1. Bluetooth desactivado

4.3.2. GPS desactivado

4.3.3. No se detecta accidente

4.3.4. Posibilidad de accidente

4.3.5. Accidente y aviso a emergencias

4.4. Prototipo

En un diseo previo como se pudo ver en algunas figuras anteriores me apoyaba sobre una protoboax y cables para la conexi3n entre los diferentes dispositivos. Lo cual se decidi3 evitar con el diseo de una placa PCB, la cual se muestra en la siguiente figura.

La descripci3n de la realizaci3n de la PCB se encuentra en el anexo 1.

Tras la fabricaci3n de la PCB procedi3 a su montaje, obteniendo el siguiente prototipo de proyecto.

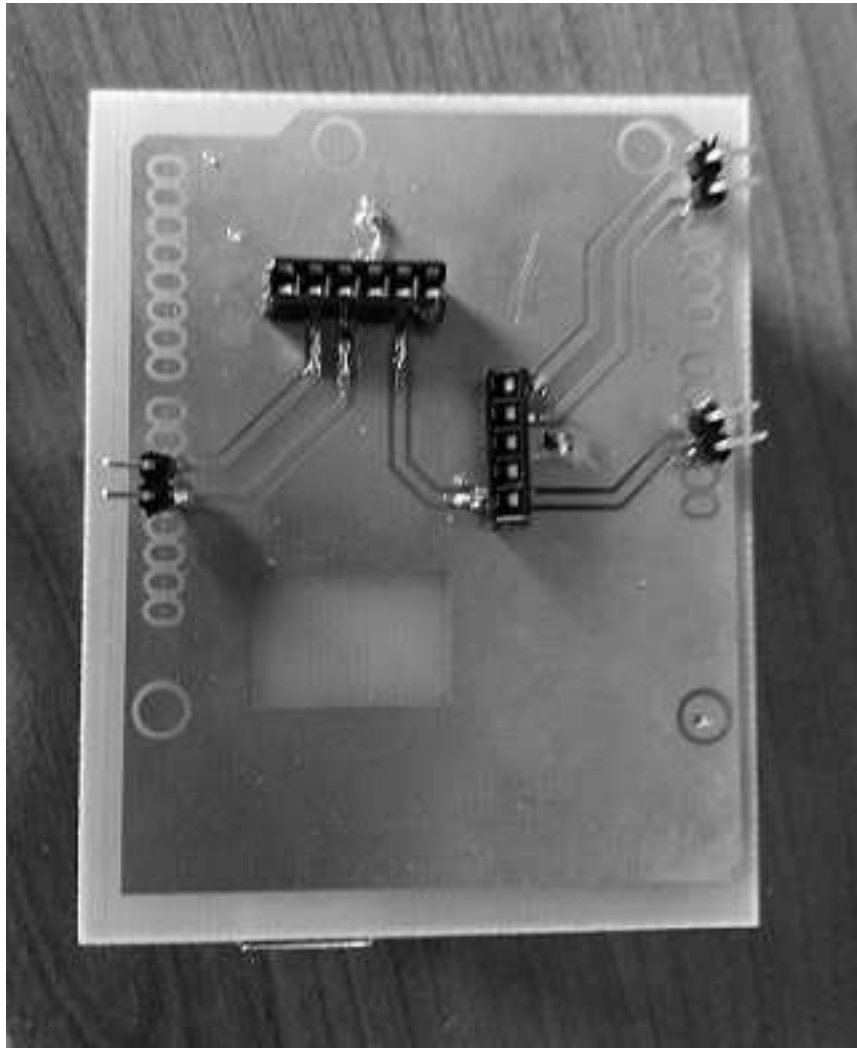


Figura 4.3: Diseño PCB



Figura 4.4: Prototipo proyecto

Capítulo 5

Conclusiones

Conclusiones

Bibliografía

- [1] ARDUINO. <http://www.arduino.cc/en/Main/arduinoBoardUno>, 2012.
- [2] ATMEL. <http://www.atmel.com/images/doc8161.pdf>.
- [3] DGT. <http://www.dgt.es/revista/num213/reportaje-tecnologia-moto.html>, 2012.
- [4] ELECTRONICA 60 NORTE. <http://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf>.
- [5] POLOLU ROBOTIC AND ELECTRONICS. <https://www.pololu.com/product/1268>.
- [6] ROVING NETWORKS. <https://www.sparkfun.com/datasheets/Wireless/Bluetooth/rn-42-ds.pdf>.
12 de Junio de 2010.
- [7] XATAKA. <http://www.xataka.com/moviles/idc-android-e-ios-ya-estan-en-el-96-4-de-los-sm>.
15 de Agosto de 2014.

Anexo 1 PCB

Anexo 2 Presupuesto

