

Proyecto fin de carrera de Ingeniería de Telecomunicaciones

Diseño de un Sistema de Alerta Móvil en Accidentes de Tráfico



Valentín Sánchez Ramírez

Dirigido por:
Miguel Ángel Carvajal Rodríguez
Fernando Martínez Martí

Ingeniería de Telecomunicaciones
Universidad de Granada
Escuela Técnica Superior de Ingeniería Informática y
Telecomunicaciones
Departamento de Electrónica y Tecnología de los Computadores

Junio 2015

Dedicado a

Índice general

Índice de figuras	V
Índice de tablas	VII
Resumen	IX
Abstract	XI
1. Introducción	1
1.1. Motivación	1
1.2. Descripción del proyecto	2
1.3. Objetivos	3
2. Estado del arte	7
2.1. Comparación de módulos inalámbricos	7
2.1.1. Infrarrojos	7
2.1.2. Wi-Fi	8
2.1.3. ZigBee	8
2.1.4. Bluetooth RN-42	9
2.1.5. Bluetooth HC-05	9
2.1.6. Conclusión	9
2.2. Mota Sensora	10
2.3. Plataforma electrónica de código abierto	11
2.4. Dispositivo para la aplicación	12
2.5. Sistemas de seguridad actuales y tendencias futuras	13
2.5.1. MY-ALARMME!	15
2.5.2. MEC — Mobile Emergency Call	15

3. Diseño	18
3.1. Diagrama estados	18
3.2. Hardware	20
3.2.1. Arduino Uno R3	20
3.2.2. Pololu minimu 9 v.2	22
3.2.3. Bluetooth HC-05	23
3.3. Firmware	23
3.3.1. Configuración Arduino Uno y Pololu Minimu 9 V.2	24
3.3.2. Configuración Arduino Uno y Bluetooth HC-05	25
3.4. Aplicación Móvil	26
3.4.1. Android	26
3.4.2. Interfaz Aplicación	27
4. Resultados	31
4.1. Resultados Hardware	31
4.2. Resultados Aplicación Android	32
4.2.1. GPS desactivado	32
4.2.2. No se detecta accidente	33
4.2.3. Posibilidad de accidente por pérdida de señal	34
4.2.4. Accidente y aviso a emergencias	34
4.3. Prototipo	34
5. Conclusiones	42
5.1. Conclusiones	42
5.2. Líneas de trabajo futuras	43
Bibliografía	45
Anexo 1	47
Anexo 2	52
Anexo 3	55

Índice de figuras

2.1. Módulo Bluetooth RN-42	9
2.2. Módulo Bluetooth HC-05	10
2.3. Pololu minimu-9 v2	11
2.4. Arduino Uno R3	12
2.5. Comparativa Sistema Operativos en uso	13
2.6. Airbag en casco y cazadora	14
3.1. Diagrama de Estados	19
3.2. ATmega328 en Arduino Uno R3	21
3.3. Conexiones Motasonra y Arduino	24
3.4. Resultado conexión todos los componentes	26
3.5. Estructura aplicación Android	27
3.6. Interfaz aplicación Android	30
4.1. Theta teórico 0	32
4.2. Theta teórico 90	33
4.3. Interfaz aplicación Android si el GPS no está activado	36
4.4. Interfaz aplicación Android reconociendo ángulo de inclinación	37
4.5. Interfaz aplicación Android con pérdida de señal bluetooth	38
4.6. Interfaz aplicación Android cuando he sufrido un accidente	39
4.7. Diseño PCB	40
4.8. Prototipo proyecto	41
5.1. Esquemático diseño PCB	48
5.2. Diseño PCB en PCBnew	49
5.3. Modelo 3D PCB	50
5.4. SDK Manager	56

Índice de cuadros

2.1. Porcentaje mundial de sistemas operativos en Smartphones	13
5.1. Coste Hardware	52
5.2. Coste Software	52
5.3. Coste total proyecto	53

Resumen

Palabras clave: seguridad en vehiculos de dos ruedas, aviso a emergencias, sensor, aplicación móvil, bluetooth, smartphone, unidad inercial.

Este proyecto fin de carrera propone un sistema de seguridad en vehículos de dos ruedas que permita una rápida intervención de los servicios de emergencias en el caso de sufrir un accidente mientras se circulaba.

Hoy en día este aviso se produce si la persona accidentada llama a emergencias o bien si algún testigo llama desde su teléfono personal.

El objetivo principal es detectar si se ha sufrido un accidente gracias a la información que suministrará un sistema electrónico situado en el interior de la moto y alimentado por la batería de la misma.

La aplicación instalada en el smartphone del usuario será la encargada de recopilar toda la información suministrada por el sistema electrónico, detectar si se ha sufrido un accidente realizando las comprobaciones pertinentes del algoritmo desarrollado y enviar un SMS al número de emergencias indicándoles nuestra ubicación en caso de haber sufrido un accidente.

Abstract

Keywords: safety two-wheeled vehicles, emergency notification, sensor, mobile application, bluetooth, smartphone, inertial unit.

This end career project proposes a security system in two-wheelers that allows a fast intervention of emergency services in the event of an accident while he was driving.

Today this warning occurs if the casualty emergency call any witnesses or if you call from your home phone.

The main objective is to detect if you have had an accident due to information that will provide an electronic system located inside of the motorcycle and powered by the battery of the same.

The application installed on the smartphone driver will be responsible for gathering all the information provided by the electronics sistem that detect when an accident making the necessary checks of the developed algorithm and send a SMS to the emergency number about our location in case of an accident.

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.

Se busca, por tanto, que con la lectura de este capítulo pueda entenderse el resto de esta memoria.

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.

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. Algunos vehículos de alta gama ya poseen un sistema de seguridad en caso de sufrir en accidente, sistema que será obligatorio en España a partir del 31 de Marzo de 2018 en coches. Quedando disponible realizar un sistema parecido para motos, el cual podría llegar a ser obligatorio en un futuro.

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 aplicaciones podemos convertir nuestro smartphone en una herramienta que nos ayude en cualquier tarea, circunstancia o incluso que nos salve la vida.

Si combinamos la tecnología de nuestro smartphone con un sistema electrónico que pueda ser integrado en una moto, ya sea desde el momento de su fabricación o un extra, aumentaríamos la seguridad de todo motorista.

Es cierto que no disminuiríamos el números de accidentes que puedan ocurrir en carretera, pero si disminuiríamos el tiempo de reacción de los servicios sanitarios, aumentando las probabilidades de atender a una persona en el menor tiempo posible.

1.2. Descripción del proyecto

Como se ha indicado, es muy importante el tiempo de reacción de los servicios de emergencias, facilitando ademas nuestra ubicación ayudamos a que nos encuentren con facilidad en caso de estar en algún lugar de difícil acceso, que no hayamos podidos llamar por nuestros medios o que no hubiera testigos de dicho accidente que pudieran llamar a emergencias para informar de la gravedad de lo ocurrido.

Además debemos tener en cuenta que este sistema funciona solo, no necesita de la intervención del motorista una vez el sistema ha sido iniciado en caso de accidente.

Para que este sistema sea efectivo es preciso que el motorista circule con un smartphone encendido, conectado vía bluetooth al sistema electrónico integrado en la motocicleta y con la aplicación iniciada.

Este proyecto pretende ser un sistema completo y autónomo que se componga de un sistema

electrónico, compuesto por un giróscopo, magnetómetro y acelerómetro, capaz de transmitir vía bluetooth a un smartphone todos los parámetros necesarios para que la aplicación desarrollada ejecute el algoritmo implementado.

Por ello podemos decir que los elementos clave para aumentar la seguridad en un vehículo de dos ruedas y reducir el tiempo de intervención de los servicios de emergencias en este proyecto serían:

- Sensores.
- Bluetooth.
- Smartphone.

Además, es necesario que el smartphone posea GPS, para una vez los datos han sido recibidos y almacenados, podamos operar con ellos en nuestro algoritmo, el cual nos comprobará si hemos sufrido un accidente en base a nuestro ángulo de inclinación y la velocidad a la que circulamos. En caso que el algoritmo detecte haber sufrido un accidente enviará un SMS a los servicios de emergencias indicándoles la ubicación recogida por el GPS.

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 aplicación 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 aplicación Android en nuestro Smartphone que sea capaz de conectarse al dispositivo HC-05 y leer la información recibida, esta aplicación 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 las 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. Se dispondrá de una protoboax para las pruebas pertinentes hasta el desarrollo de una placa en PCB para posterior implementación.
- Estudio de programación Android en Eclipse. Con la correspondiente aplicación 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. Pruebas a diferentes velocidades, diferentes ángulos de inclinación del sistema electrónico que irá integrado en la moto. Además de pruebas del sistema en diferentes circunstancias meteorológicas para comprobar el alcance del dispositivo bluetooth.
- Desarrollo de PCB's. Desarrollo de una placa PCB para evitar el uso de cables en el proyecto y hacerlo mas compacto.

Además los objetivos que debemos cumplir de cara a la comercialización de este proyecto serían los siguientes:

- Acabado sistema electrónico: debe ser un sistema de tamaño reducido, de facil ingración a una moto y con una entrada de alimentación desde la batería.
- Comodidad de uso de cara al usuario: el individuo que disponga de dicho proyecto, solo debe preocuparse de emparejar el bluetooth de su smartphone con el bluetooth del sistema electrónico, despues de ello, solo debe encender la aplicación antes de circular con su moto y apagarla una vez haya llegado a su destino.
- Monitorización en tiempo real: se requiere de una comunicación en tiempo real entre smartphone y sistema electrónico, para que se pueda actuar en la mayor brevedad posible.
- Movilidad: la aplicación puede ser instalada en cualquier dispositivo portatil Android que

posea bluetooth, GPS y pueda enviar SMS, por lo que debe poseer una tarjeta SIM que posibilite el uso de la red telefónica.

- Disponibilidad, robustez y eficiencia: el sistema debe ser tolerante y recuperable a fallos. Por un ejemplo un fallo puede ser la pérdida de comunicación bluetooth momentanea, por lo que se reconoce como un fallo puntual y no como un accidente. La transmisión de datos debe diseñarse con una estructura lo mas eficiente posible, con una buena sincronización. Además se debe desarrollar una interfaz de la aplicación intuitiva.
- Interpretabilidad de los datos: no solo la monitorización en tiempo real es necesaria para el correcto funcionamiento del algoritmo implementado. Una buena y rápida interpretabilidad de los datos es fundamental. En esta versión de prueba los datos se mostrarán en diferentes TextView, posibilitando así la interpretación de los datos y poder ver el comportamiento del algoritmo en función de los datos recibidos.

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 existente y nuestra decisión, sensor utilizado, plataforma electrónica de código abierto utilizada, dispositivo para el que se desarrollará la aplicación, sistemas de seguridad actuales en vehículos de dos ruedas y las tendencias futuras.

Se acordó trabajar con la plataforma hardware Arduino, debido a las innumerables ventajas con las que cuenta, plataforma absolutamente desconocida, por lo que necesitaría una amplia labor previa de búsqueda de información, lectura de documentos, descarga de nuevos entornos de desarrollo y familiarización con dicha plataforma electrónica.

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 Infrarrojos, módulo Wi-fi, ZigBee y los módulos Bluetooth RN-42 y HC-05.

2.1.1. Infrarrojos

De forma general podemos definir la comunicación vía infrarrojos como un haz enfocado de luz en el espectro de frecuencia infrarrojo, donde se modula la información y se envía de un transmisor a un receptor a una distancia relativamente corta.

Sus características son:

- Bajo coste.
- Bajos requerimientos de potencia.
- Alta inmunidad al ruido.
- Conectividad direccional.

Su principal desventaja frente a otros módulos de comunicación es su modo punto a punto, que requiere que transmisor y receptor estén alineados, lo más cerca posible y con una línea de visión directa.

2.1.2. Wi-Fi

Es la tecnología usada en una red o conexión inalámbrica para la comunicación de datos entre equipos situados dentro de una misma área, interior o exterior, de cobertura.

Sus ventajas son:

- Movilidad desde cualquier sitio dentro de su cobertura.
- Fácil instalación.
- Flexibilidad, permite el acceso a una red en entornos de difícil cableado.
- Permite incorporar redes sin la necesidad de cables.

2.1.3. ZigBee

ZigBee es una alianza, sin ánimo de lucro, de 25 empresas, la mayoría de ellas fabricantes de semiconductores, con el objetivo de auspiciar el desarrollo e implantación de una tecnología inalámbrica de bajo coste. Los miembros de tal alianza justifican el desarrollo de este estándar para cubrir el vacío que se produce por debajo del Bluetooth.

Sus propiedades más destacadas son:

- Velocidad entre 20 kB por segundo y 250 kB por segundo, con un rango entre 10 y 75 metros de alcance.
- Bajo consumo.

2.1.4. Bluetooth RN-42

Este módulo Bluetooth [8] 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

2.1.5. Bluetooth HC-05

Este módulo Bluetooth [5] 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

2.1.6. Conclusión

Estudiados estos 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



Figura 2.2: Módulo Bluetooth HC-05

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 [7], 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

Arduino es una plataforma de electrónica que realiza prototipado de circuitos, de software libre, basada en hardware y software flexible y fácil de usar. De manera más concreta, en lo que respecta, es la combinación de un microcontrolador y placa base asociada con un entorno de desarrollo que simplifica la programación de este controlador, la cual es en un lenguaje muy parecido a C y que por tanto permite la comunicación con un gran número de dispositivos de forma cómoda y asequible.

Arduino no es la única plataforma basada en microcontroladores, pero si es una plataforma libre, económica y con una gran comunidad creciendo detrás de ella. Además posee una filosofía multidisciplinar, combinado con la facilidad de tomar información del entorno a través de sus diferentes entradas analógicas y digitales.

Existen diferentes placas oficiales de Arduino, entre ellas Arduino Mega, Yún, Leonardo, Uno, Esplora, Due, Micro, Robot, etcétera.

La plataforma electrónica de código abierto que vamos a utilizar es Arduino Uno R3 [2], 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.

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.

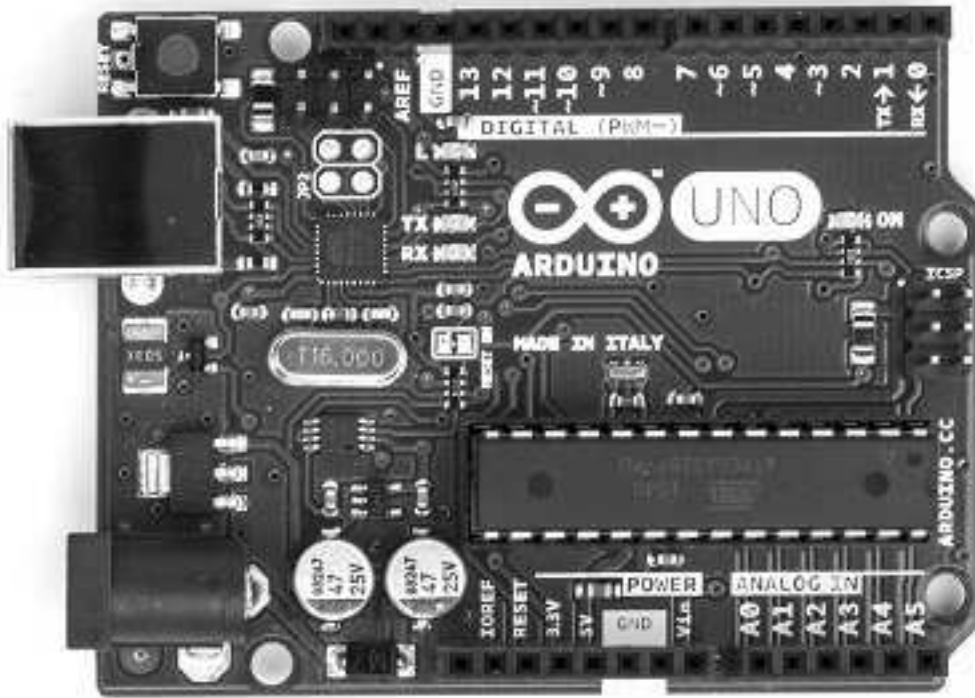


Figura 2.4: Arduino Uno R3

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.

Una de las ventajas que ofrece Arduino es:

- Su bajo coste.
- La facilidad para leer los datos suministrados por sensores.
- La facilidad para construir el sistema electrónico que deseamos.

2.4. Dispositivo para la aplicación

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 [9] podemos ver los datos proporcionados por la consultora IDC, resultados que se muestran a continuación.

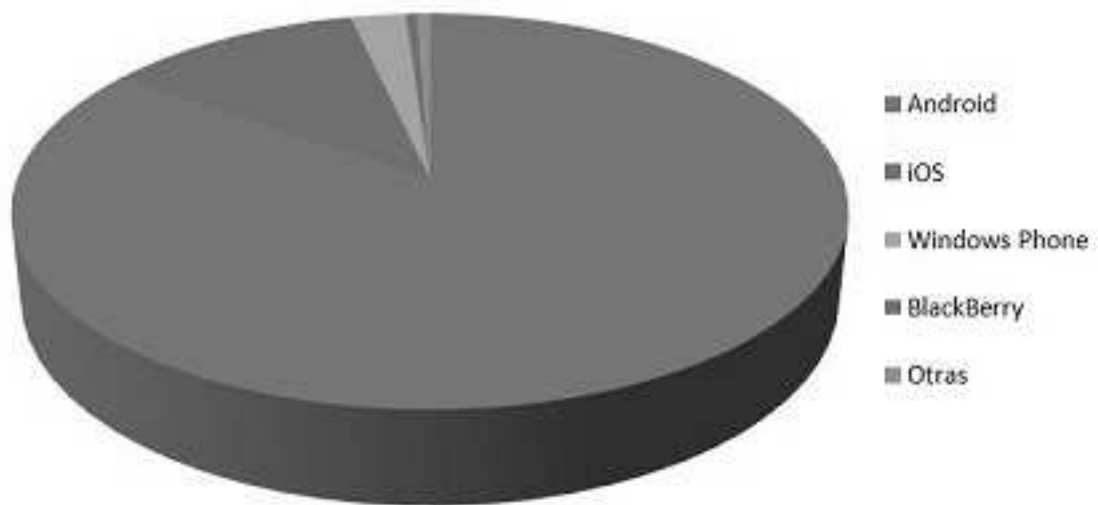


Figura 2.5: Comparativa Sistema Operativos en uso

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 [4].

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 aplicación he llamado MotoSafe. Consiste en que el propio smartphone del conductor pueda avisar a emergencias indicándoles nuestra posición en caso que la aplicación 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.

A continuación detallaré algunos sistemas actuales de alerta móvil en caso de accidentes de tráfico.

2.5.1. MY-ALARMME!

MyAlarmMe! [6] es un sistema de alarma y emergencia para moteros que funciona a través del teléfono móvil.

Ofrece un servicio personalizado que permite al usuario avisar, en tiempo real y de forma manual, a todas las personas que el desee con el envío de un SMS que ha sufrido un accidente.

Los sms se enviarán de forma automática al activar tu alarma, el mensaje enviado será prefijado por el usuario incluyendo el nombre de dicho usuario y las coordenadas de su ubicación.

El funcionamiento es simple, una vez la aplicación ha sido configurada, en caso de sufrir un accidente el usuario solo tiene que abrir la aplicación y pulsar sobre el icono de exclamación, en ese momento se enviarán los sms a los contactos o números previamente indicados.

2.5.2. MEC — Mobile Emergency Call

MEC – Mobile Emergency Call [1] es una aplicación que envía mensaje de alarma automáticamente si detecta que hemos tenido un accidente de tráfico. Los mensajes de alerta que envía son por medio de email, SMS y llamada de voz con locución incorporada.

Este sistema se puede usar en coches, motos y bicicletas, teniendo en cuenta las particularidades dinámicas y de conducción de cada vehículo. Si el algoritmo detecta un accidente se inicia una cuenta atrás para enviar los mensajes de alerta, cuenta atrás que puede ser cancelada si el usuario así lo indica.

En esos mensajes de alerta se incluyen la posición GPS del accidente, además de los datos de usuario previamente insertados, como son el modelo de vehículo, alergias conocidas, etcétera. El usuario puede escoger a quien desea enviar dichos mensajes de alerta.

Capítulo 3

Diseño

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. Diagrama estados

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

A la hora de empezar a usar la aplicación 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 aplicación y comenzar a usarla.

Al abrir la aplicación 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 aplicación.

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 aplicación 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 aplicación se cerrará automáticamente.

Posteriormente y de forma automática se creará un Handler que será el encargado de conectar la aplicación 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 aplicación no ejecutará el algoritmo implementado.

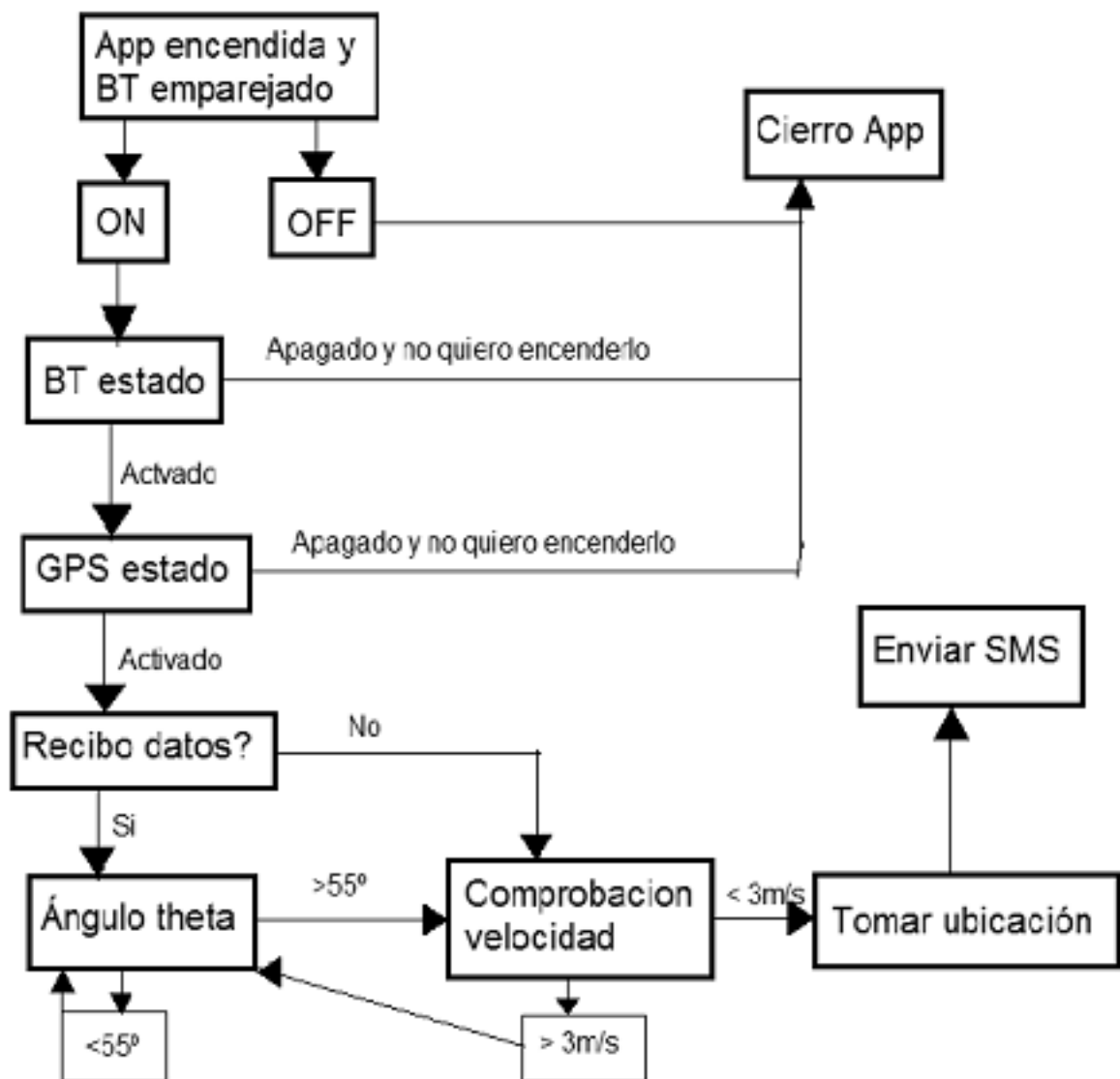


Figura 3.1: Diagrama de Estados

Una vez todo está activado, la aplicación 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 interpretados, 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.

3.2. Hardware

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

3.2.1. Arduino Uno R3

Arduino Uno R3 [2], 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.

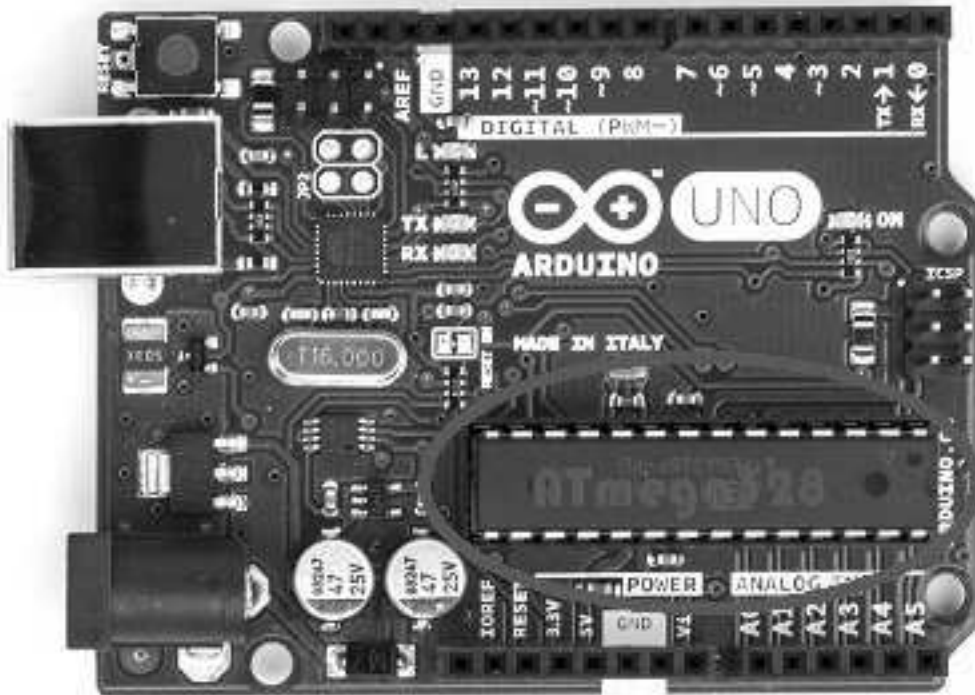


Figura 3.2: ATmega328 en Arduino Uno R3

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 [3] es un microcontrolador creado por Atmel y que pertenece a la serie megaAVR. Cabe destacar:

- Es un circuito integrado de alto rendimiento que está basado en un microcontrolador RISC.
- Combina 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.
- 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.
- Alcanza una respuesta de 1 MIPS.
- Consumo balanceando de energía y velocidad de proceso

3.2.2. Pololu minimu 9 v.2

Pololu minimu-9 v2 [7], 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.

A continuación procederé a explicar con detalle cada uno de los integrados pertenecientes a dicho Pololu.

El L3GD20 es un sensor de tres ejes para medir velocidad angular a baja potencia.

Incluye un elemento de detección y una interfaz IC capaz de proporcionar la velocidad angular medida con el exterior a través de una interfaz digital (I2C / SPI).

El sensor está fabricado usando un proceso de micro-mecanizado dedicado al desarrollado por STMicroelectronics para producir sensores inerciales y actuadores de silicio.

La interfaz IC se fabrica utilizando un proceso CMOS que permite un alto nivel de integración para diseñar un circuito dedicado que se recorta para adaptarse mejor a las características del elemento de detección. El L3GD20 tiene una escala de 250 - 2.000 dps y es capaz de medir las tasas con un ancho de banda seleccionable por el usuario.

El L3GD20 está disponible en un paquete de plástico y puede operar dentro de un rango de temperatura de -40 a 85 grados centígrados.

El LSM303DLHC es un sistema empaquetado con un sensor lineal digital en 3 ejes de aceleración y un sensor magnético digital en 3 ejes.

El LSM303DLHC tiene escalas lineales llenos de aceleración de 2, 4, 8 y 16 g y un campo magnético a gran escala de 1,3, 1,9, 2,5, 4,0, 4,7, 5,6 y 8,1 gauss.

El LSM303DLHC incluye una interfaz de bus serie I2C que soporta el modo estándar y rápido a 100 kHz y 400 kHz. El sistema puede ser configurado para generar señales de interrupción por eventos, así como por la posición del propio dispositivo. Los umbrales y tiempos de generadores de interrupción son programables por el usuario final. Los bloques magnético y acelerómetro se pueden activar o poner en modo de apagado por separado.

El LSM303DLHC está disponible en un paquete de plástico y puede operar dentro de un rango de temperatura de -40 a 85 grados centígrados.

3.2.3. Bluetooth HC-05

Este módulo Bluetooth [5] 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.

3.3. 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.

3.3.1. 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:

- include L3G.h
- include LSM303.h

Además hemos usado la librería `#include SoftwareSerial.h`, que permite la comunicación entre los pines de la motasensora y de la placa Arduino.

Una vez cargadas dichas librerías inicializamos todas las variables a usar en el código además de los pertinentes métodos para tomar los valores medidos de cada sensor en los 3 ejes. En lo que respecta al código Arduino siempre tendremos dos métodos a usar, como son `setup()` el encargado de configurar la placa e inicializar los métodos y el método `loop()`, el cual ejecutará las acciones que le ordenemos de forma reiterativa.

En lo que respecta a las conexiones, conectamos la patilla SDA y SCL con la patilla A4 y A5 respectivamente de la placa Arduino, además de VCC y GND entre la motasensora y la placa.

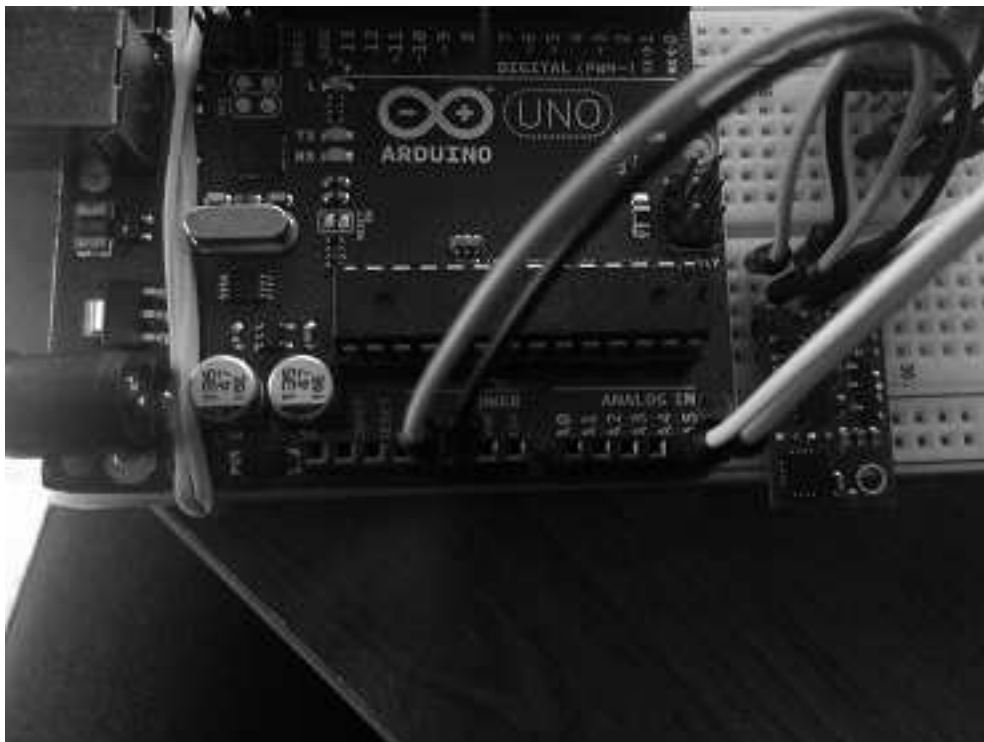


Figura 3.3: Conexiones Motasonra y Arduino

En lo que respecta a la configuración de la mota 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.

3.3.2. 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

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.

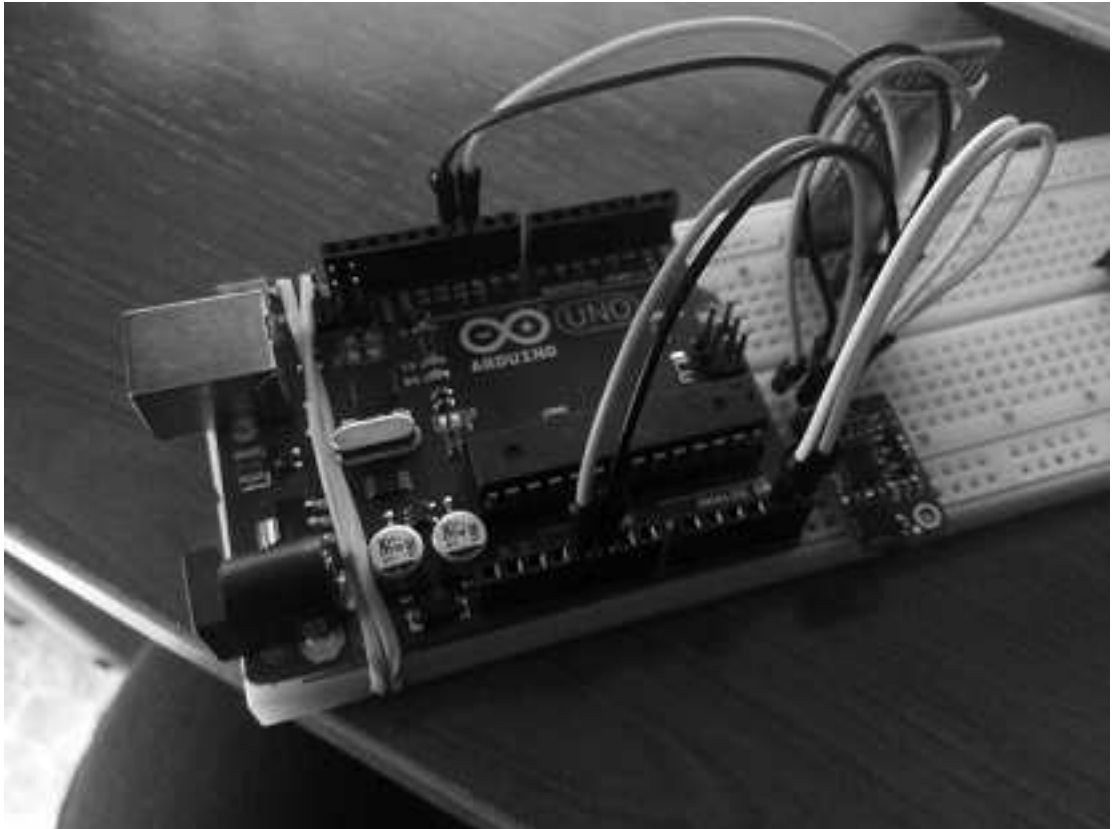


Figura 3.4: Resultado conexión todos los componentes

3.4. Aplicación Móvil

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

3.4.1. Android

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

Aquí 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.

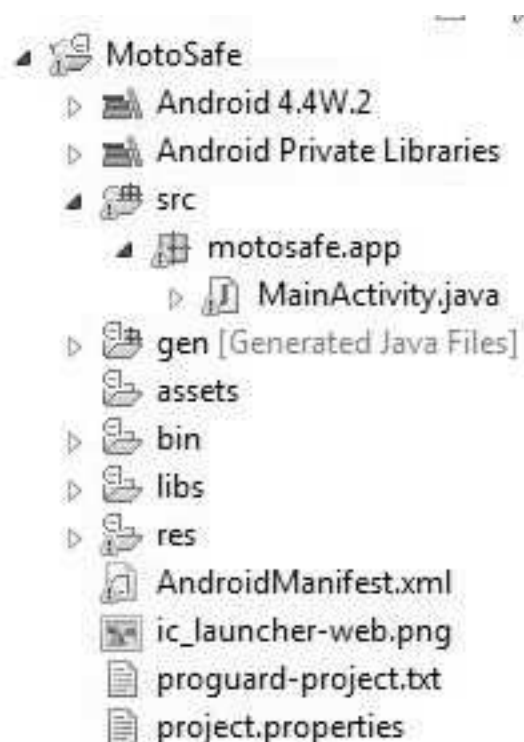


Figura 3.5: Estructura aplicación Android

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 enviársela vía SMS a emergencias.

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

3.4.2. Interfaz Aplicación

La interfaz eventual de nuestra aplicación 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 aplicación.

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 3.6: Interfaz aplicación 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 con la realización de pruebas de evaluación y análisis de dichos resultados.

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 [5] 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.

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.

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.

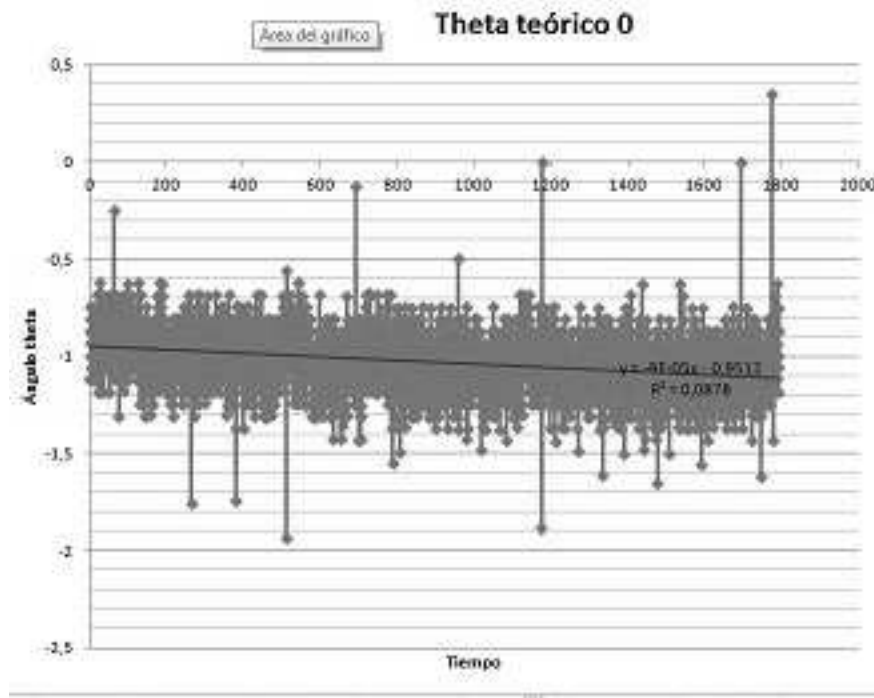


Figura 4.1: Theta teórico 0

4.2. Resultados Aplicación Android

En lo que respecta a los resultados de la aplicación Android vamos a detallar algunas de las situaciones ya explicadas en el capítulo desarrollo sobre como se comporta nuestra aplicación en diferentes circunstancias, mostrando una captura de pantalla de la interfaz en ese instante.

Veremos los siguientes casos:

- Si el GPS está desactivado.
- Si no se detecta accidente.
- Posibilidad de accidente por pérdida de señal.
- En caso de sufrir un accidente, como se comporta la aplicación.

4.2.1. GPS desactivado

En el caso de encontrar el sistema GPS de nuestro smartphone desactivado, nos aparecerá un alert, ofreciendonos la posibilidad de activarlo en ese instante. Ver figura 4.3.

Si pulsamos sobre "No" la aplicación se cerrará automáticamente, en caso de pulsar sobre "Menu de opciones GPS" accedemos a la sección de ajustes del smartphone para poder encenderlo

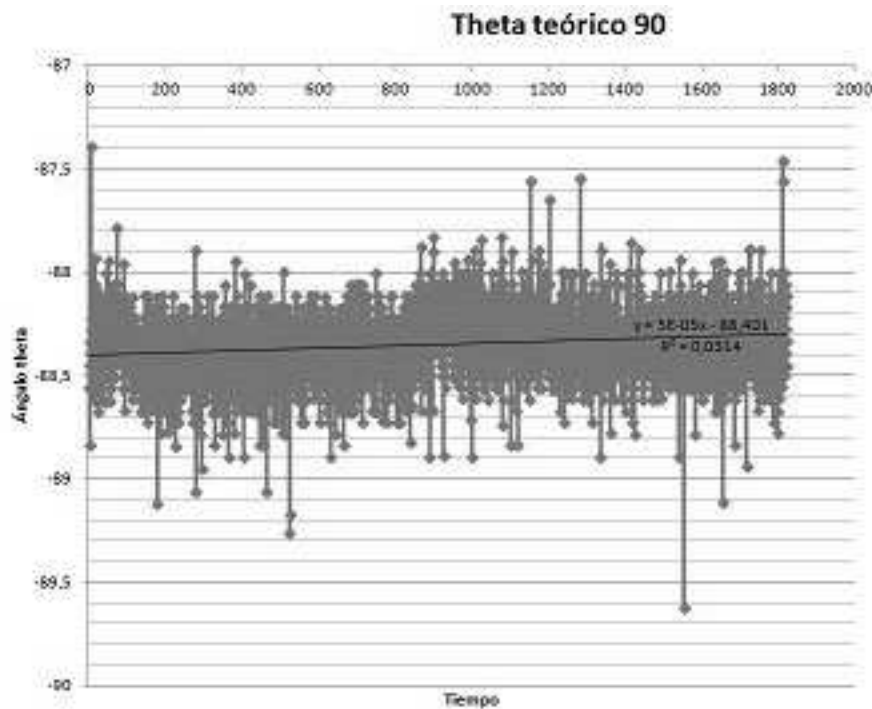


Figura 4.2: Theta teórico 90

y continuar usando la aplicación.

Hemos de tener en cuenta que si no activamos el GPS, ya sea desde esta opción o previamente no podremos usar la aplicación MotoSafe.

4.2.2. No se detecta accidente

Una vez nuestra aplicación comienza a funcionar, mide el ángulo de inclinación de la motocicleta. Ver figura 4.4.

Tal y como se puede observar en la figura 4.4, en el primer TextView, el ángulo de inclinación medido es 0,1217 grados, lo cual indica que permanecemos sobre la vertical y nuestra inclinación es prácticamente nula.

Además el segundo TextView reconoce que no se ha sufrido una caída y nos muestra dicho mensaje para poder ver el funcionamiento de la aplicación.

4.2.3. Posibilidad de accidente por pérdida de señal

Muchos son los motivos por los que se puede perder la comunicación entre el smartphone y el dispositivo que integraremos en la moto, lo cual debemos tener en cuenta para evitar el envío de falsos mensajes de accidente al número de emergencias. Ver figura 4.5.

Como se puede apreciar en ésta figura, la aplicación reconoce la pérdida de comunicación, mostrando el mensaje "perdida de BT", además según el algoritmo que he implementado reconoce como que aún no se ha caído como se puede apreciar en el segundo TextView, pero tal y como se pudo ver en el diagrama de estados del capítulo desarrollo, el algoritmo pasa a comprobar la velocidad a la que circulamos, muestra de ello es el mensaje "mido velocidad" que se encuentra en el tercer TextView.

4.2.4. Accidente y aviso a emergencias

Una vez hemos detectado dicho que ha ocurrido un accidente, ya sea por la pérdida de comunicación o un ángulo de inclinación superior a 55 grados y una velocidad menor a 3 metros por segundo, podemos ver la siguiente interfaz. Ver figura 4.6.

En el cuarto TextView se muestra el SMS que ha sido enviado al número de emergencias.

4.3. Prototipo

En un diseño previo como se pudo ver en algunas figuras anteriores me apoyaba sobre una protoboax y cables para la conexión entre los diferentes dispositivos. Lo cual se decidió evitar con el diseño de una placa PCB, la cual se muestra en la siguiente figura. Ver figura 4.7.

La descripción de la realización de la PCB se encuentra en el anexo 1.

Tras la fabricación de la PCB procedí a su montaje, para ello perforé la placa en los puntos de interés y soldé con estaño los diferentes conectores macho y hembra, obteniendo el siguiente prototipo de proyecto.

He de destacar que en la imagen mostrada el módulo bluetooth HC-05 se encuentra en posición vertical, perpendicular a la placa, en realidad su posicionamiento debe ser en paralelo a la placa. ver figura 4.8.



Figura 4.3: Interfaz aplicación Android si el GPS no está activado



Figura 4.4: Interfaz aplicación Android reconociendo ángulo de inclinación



Figura 4.5: Interfaz aplicación Android con pérdida de señal bluetooth

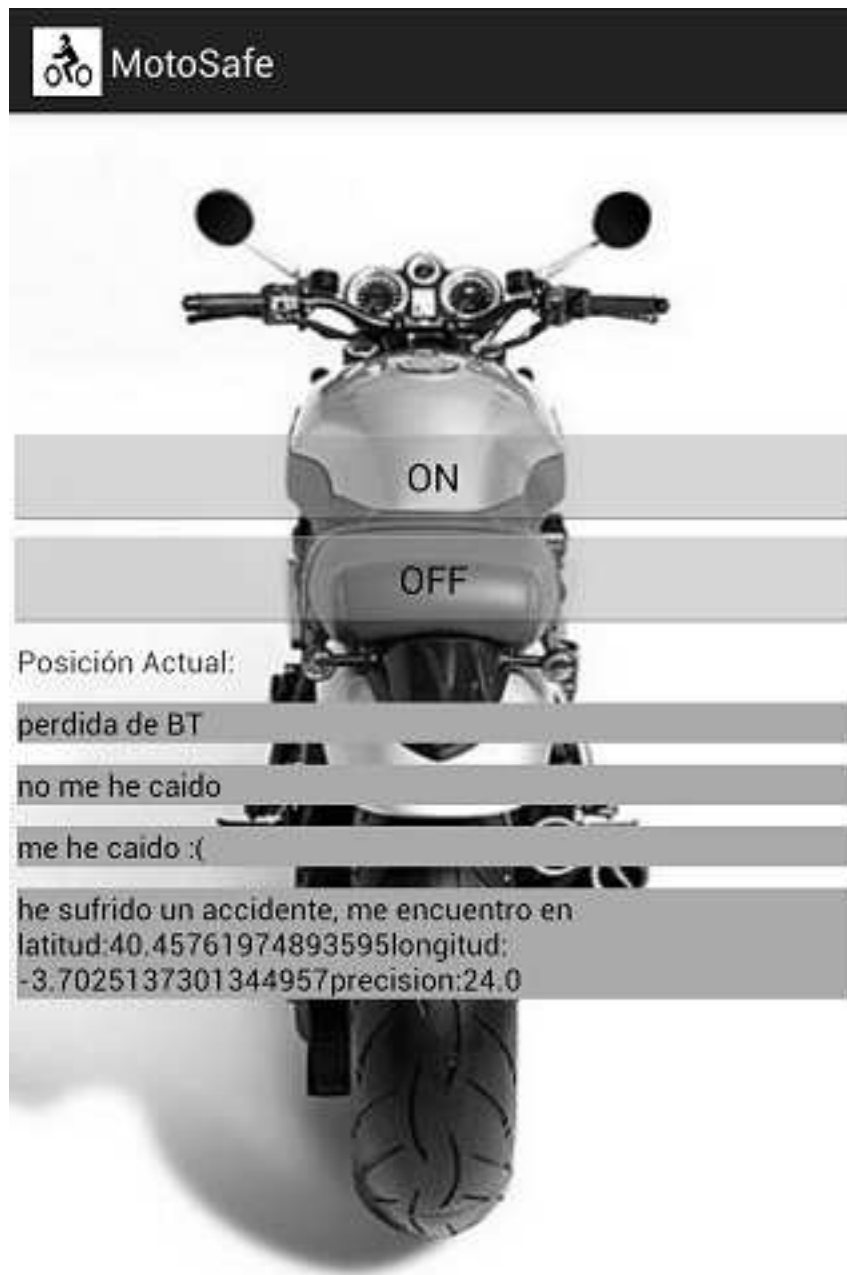


Figura 4.6: Interfaz aplicación Android cuando he sufrido un accidente

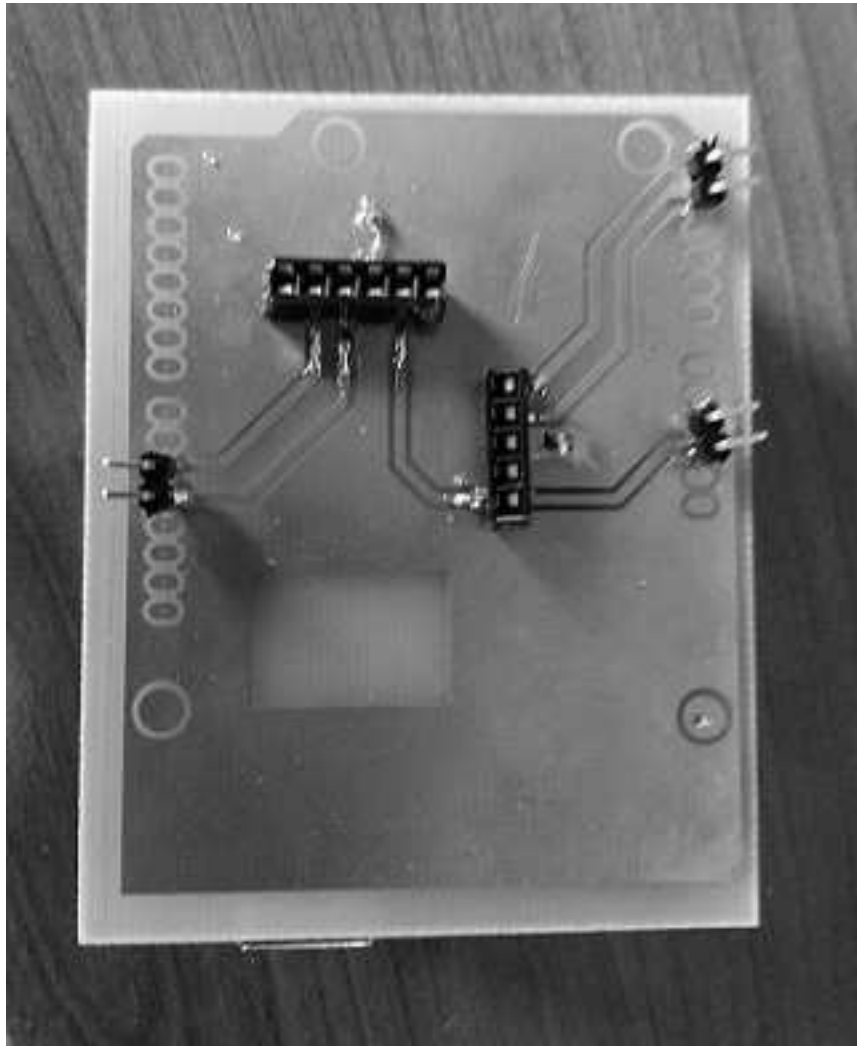


Figura 4.7: Diseño PCB



Figura 4.8: Prototipo proyecto

Capítulo 5

Conclusiones

En este capítulo se lleva a cabo una reflexión sobre el trabajo realizado, se proponen posibles mejoras que podrían realizarse en un futuro en función de los avances tecnológicos y de cara a las necesidades que surjan si procedemos a comercializar este producto.

5.1. Conclusiones

En el presente proyecto se ofrece una monitorización a distancia del sistema electrónico que vamos a instalar en la moto. El sistema se compone de dos dispositivos, uno de ellos será el sistema electrónico que se implantará en la moto compuesto principalmente por la placa Arduino Uno y un smartphone con la aplicación MotoSafe que poseerá la persona encargada de conducir la motocicleta. A través del bluetooth ambos sistemas podrán interactuar, de forma que el smartphone sea el encargado de recibir la información transmitida por la moto sensora del sistema electrónico implantado en la moto.

Este sistema ha sido diseñado en función a unos requisitos de movilidad, independencia, tiempo real y tolerancia a errores, entre otros, el diseño de una PCB eficiente y útil, la cual nos ayudará a evitar el uso de cables en nuestro sistema electrónico. Además el smartphone será el encargado de monitorizar el sensor a través de la información recibida y aplicar el algoritmo para detectar si ha habido un accidente.

Cualquier persona con un smartphone podrá poseer la aplicación, solo necesitará la implantación del sistema electrónico en su moto y podrá usar este sistema de seguridad en cualquier momento. Diariamente todas las personas llevan un smartphone encima, lo cual hace que este sistema de

seguridad sea cómodo de llevar al estar ya acostumbrados.

Además hay que destacar la utilización tanto de hardware como de software libre, Android y Arduino, que proyectan un sistema extensible por terceras personas, facilitando así las líneas de trabajo futuras y permitiendo la potencial mejora mediante la colaboración de todos.

5.2. Líneas de trabajo futuras

El diseño de este proyecto deja abierta la puerta a múltiples líneas de desarrollo futuras, ya sean de forma interna o a través de terceros. Algunas de estas líneas se detallan a continuación:

- Aunque el sistema electrónico a implantar en la moto ocupa un espacio reducido, una línea de mejora importante será la posibilidad de reducir el tamaño de dicho sistema. Para ello se seleccionarían los componentes que usamos de la placa Arduino y desarrollar por nosotros mismos una placa con dichos componentes, con la posibilidad de reducir gastos y hacer un sistema mas robusto y estable.
- Una línea de mejora en el desarrollo software sería la mejora del algoritmo de detección de accidente, posibilitando la variación del ángulo de inclinación máximo en función de la cilindrada, peso y dimensiones de la propia moto.
- Por último, se plantea la idea de mejorar el protocolo de comunicación mediante la integración de un sistema de recuperación o corrección de errores. El más simple a implementar sería el bit de paridad, pero existen otros mucho mas eficientes, aunque más complejos, para la implementación de un protocolo adecuado para el manejo de información del sistema.

Bibliografía

- [1] APP CIRCUS. <https://appcircus.com/apps/mec-mobile-emergency-call>, 2010.
- [2] ARDUINO. <http://www.arduino.cc/en/Main/arduinoBoardUno>, 2012.
- [3] ATMEL. <http://www.atmel.com/images/doc8161.pdf>.
- [4] DGT. <http://www.dgt.es/revista/num213/reportaje-tecnologia-moto.html>, 2012.
- [5] ELECTRONICA 60 NORTE. <http://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf>.
- [6] MUTUA MOTERA. <http://mutuamotera.blogspot.com.es/2012/04/nuevo-sistema-de-alerta-de>
2012.
- [7] POLOLU ROBOTIC AND ELECTRONICS. <https://www.pololu.com/product/1268>.
- [8] ROVING NETWORKS. <https://www.sparkfun.com/datasheets/Wireless/Bluetooth/rn-42-ds.pdf>.
12 de Junio de 2010.
- [9] 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

Para el diseño de la placa PCB se usó la herramienta software de código abierto KiCad, plataforma para la creación y diseño de esquemáticos de circuitos electrónicos y sus correspondientes archivos de fabricación de placas de circuito impreso. El hecho de ser una plataforma de código abierto lo hace perfecto para proyectos orientados a la creación de hardware electrónico.

Kicad no tiene limitación del tamaño de la placa y puede manejar facilmente hasta 16 capas de cobre y hasta 12 capas técnicas. Puede generar todos los ficheros necesarios para la fabricación de placas de circuito impreso, tales como: Gerber files, component locantion files...

Podemos descargar KiCad desde <http://www.kicad-pcb.org>, donde se nos descargará un ejecutable que instalará en la ruta deseada las herramientas de KiCad.

Una vez ha sido instalado KiCad, lo primero que debemos realizar es diseñar el Esquemático de nuestro proyecto. Para ello detallaré a continuación los pasos que seguí y el resultado que obtuve.

Los pasos seguidos fueron los siguientes:

- Lo primero de todo es crear un nuevo proyecto y guardarlo con el nombre que deseemos.
- Una vez ha sido creado el proyecto debemos pulsar sobre EESchema para comenzar a crear el esquemático del proyecto.
- Algunos componentes se encuentran en las librerías ya instalados, otros como el de la placa Arduino lo pude descargar de Internet y posteriormente asociándolo a su footprint.
- Creados todos los componentes procedí a conectarlos con cables, incluyendo las fuentes de alimentación, las tierras y definir sin conexión los pines que no utilizaré.
- Una vez el esquemático ha sido terminado hay que proceder a enlazar todos los componente del esquemático con Cvpcb.

El esquemático desarrollado es el siguiente:

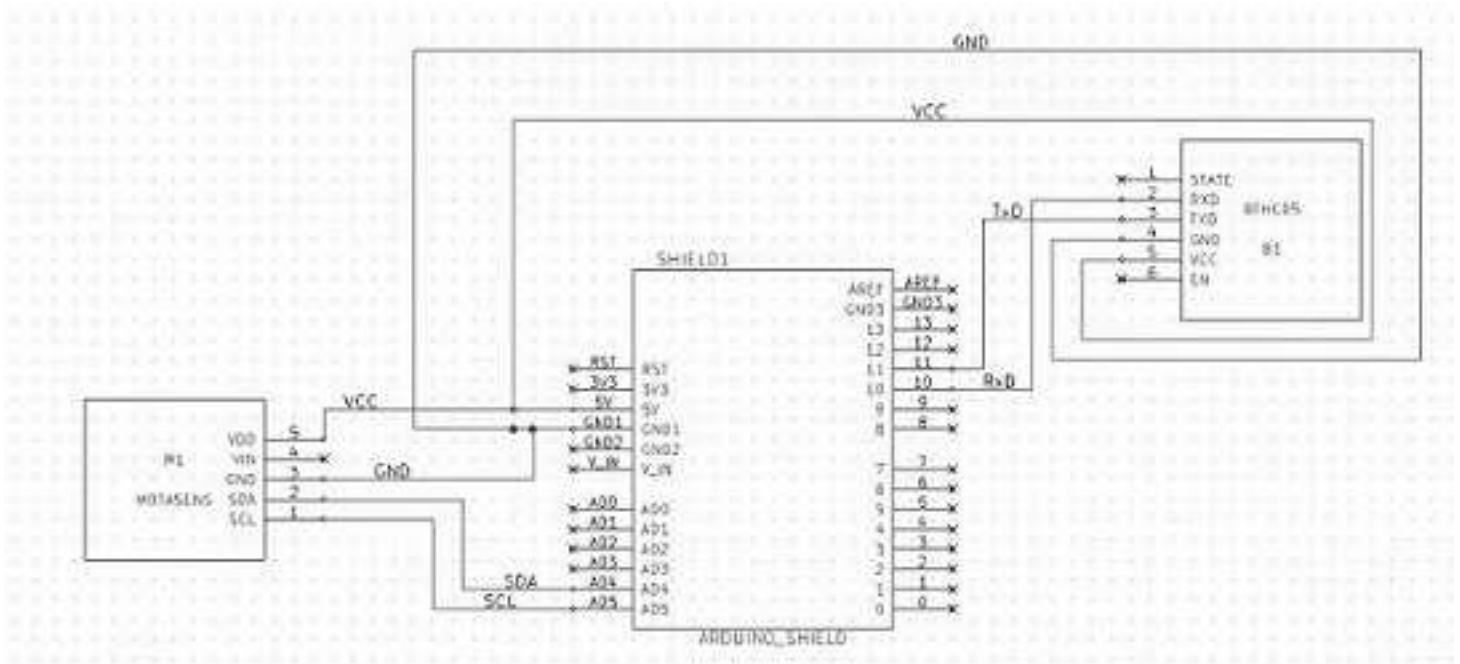


Figura 5.1: Esquemático diseño PCB

Ahora podemos usar la herramienta PCBnew, en la cual diseñaremos el boceto de nuestra PCB final.

Los pasos a seguir son los siguientes:

- En primer lugar debemos fijar el tamaño de la página a A4 y titularla con el nombre de nuestra PCB.
- En segundo lugar debemos fijar el tamaño el clearance, ancho de las vías, separación mínima de las vías, etc.
- Accedemos al NetList, para examinarlo y leer el Netlist actual para cargar todos los footprints asociados.
- Una vez han sido cargados, debemos colocarlos de una forma ordenada, teniendo en cuenta donde vamos a ubicar cada componente y que no haya posibles cruces de pistas ya que solo vamos a imprimir la placa a una sola capa.
- Con "Edge Cuts" fijaremos el tamaño de la placa, además escogeremos la capa F.Cu con la cual creamos las vías que conectarán todos los pines.
- El resto de la placa no usada se rellenará con la capa "F.Cu" con la finalidad de ser el

plano de masa que irá unido a los pines tierra. Debemos dejar un espacio sin rellenar, que será donde estará la antena del dispositivo bluetooth, para evitar que se apantalle y se reduzca de forma considerable el alcance de dicha antena.

- Una vez ha sido creado y unido todo debemos realizar un chequeo de la PCB diseñada, lo que en nuestra herramienta se reconoce como "Comenzar DRC". Realizado dicho chequeo debemos comprobar que la "Lista No Conectados" se encuentra vacía.

Realizado todo esto podemos ver el resultado de la placa que hemos diseñado con PCBnew en la siguiente imagen.

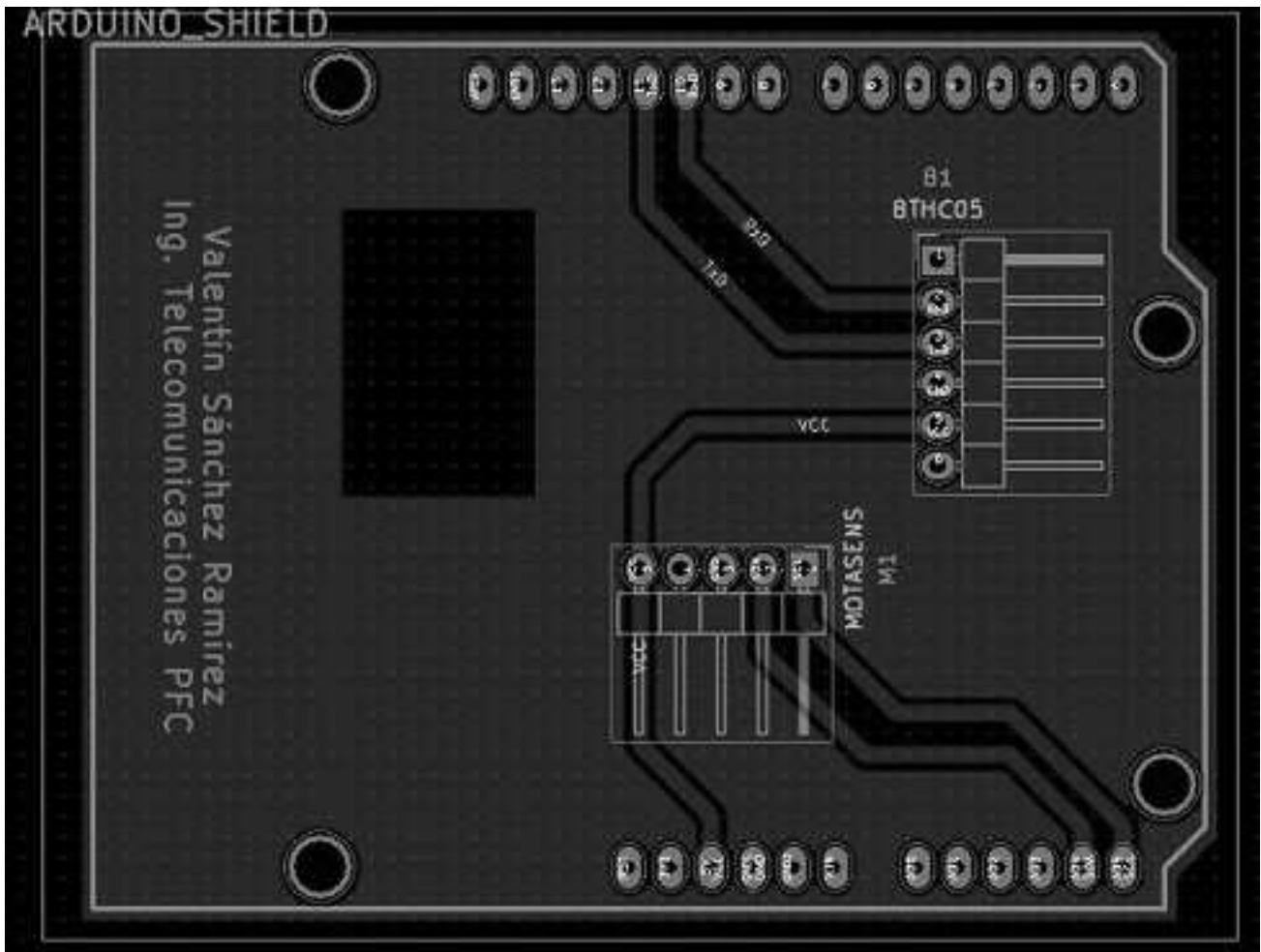


Figura 5.2: Diseño PCB en PCBnew

Además si lo deseamos podemos ver el resultado en 3D, accediendo a la pestaña Ver y pinchando sobre Visualización 3D. Obteniendo el siguiente resultado.

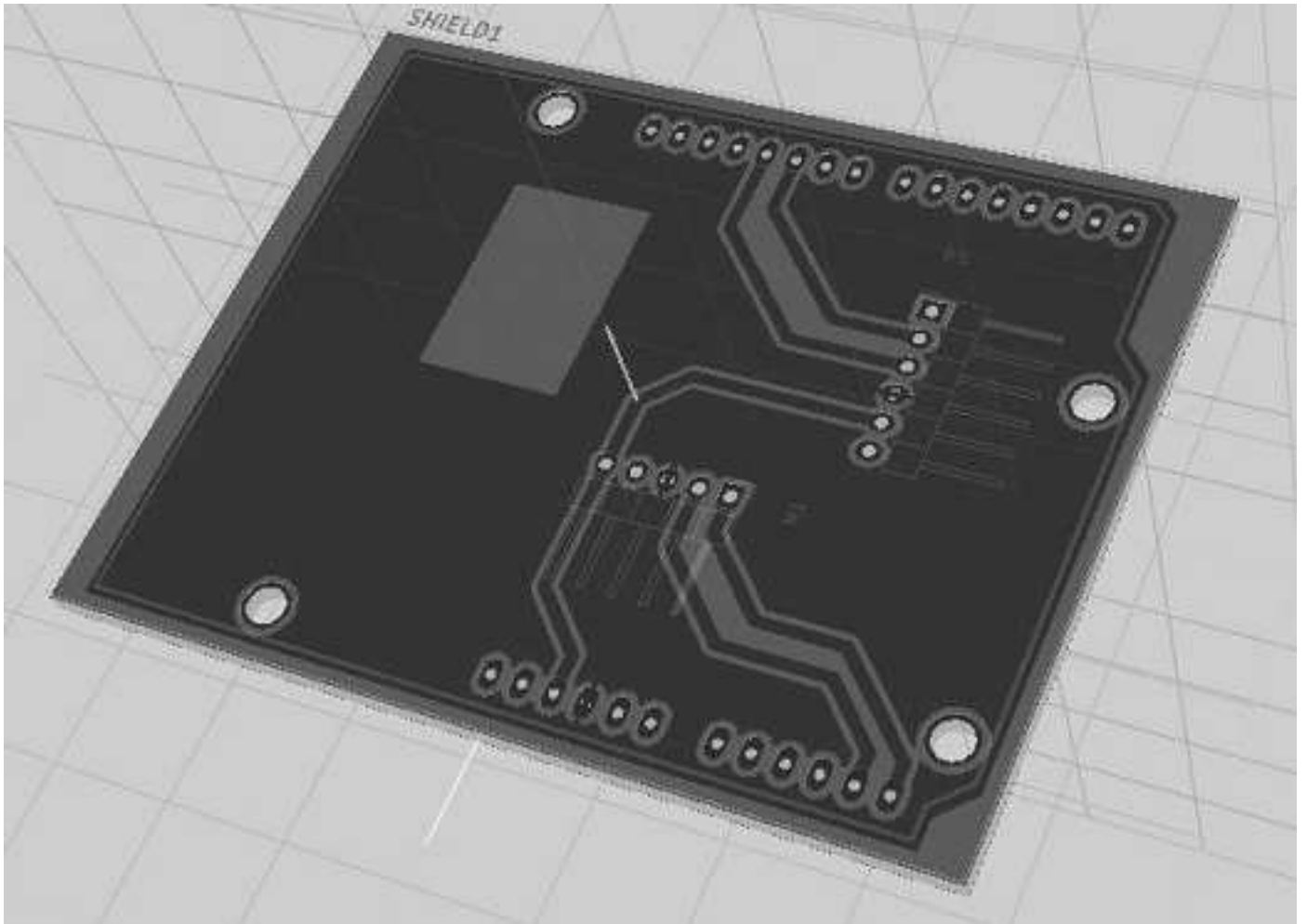


Figura 5.3: Modelo 3D PCB

Además se nos ofrece la posibilidad de generar los ficheros Gerbers. Si queremos generarlos debemos seguir los siguientes pasos.

- Desde PCBnew accedemos a Archivo y despues Plot.
- Elegimos "Gerber" en el desplagable de "Plot Format" y elegimos el directorio donde deseamos guardar los archivos.
- Elegimos las capas que deseamos dibujar para una placa de una sola capa.
- Hacemos click sobre "Trazar" para generar los Gerbers y sobre "Generate Drill File" para generar el archivo de drills.
- Podemos acceder a ver los archivos generados pulsando sobre el icono del GerbView.

Anexo 2 Presupuesto

Se va a elaborar un presupuesto en base a los componentes electrónicos utilizados y horas totales de trabajo empleadas para la realización de dicho proyecto.

En un primer lugar se va a calcular el coste de todos los componentes hardware utilizados para la realización de este proyecto.

Componentes hardware	Coste
Arduino Uno R3	27,38 €
Bluetooth HC-05	7,90 €
Pololu Minimu9 v2.0	19,95 €
PCB	9,90 €
Smartphone gama media	199,90 €
Total	265,03 €

Cuadro 5.1: Coste Hardware

En lo que respecta al desarrollo software vamos a estimar el presupuesto en función de las horas de programación invertidas, teniendo en cuenta que un ingeniero estima un precio de 20 €/la hora y en la programación Arduino se ha invertido 15 horas y en la programación Android se han invertido 60 horas.

Desarrollo software	Coste
Programación Arduino	300 €
Programación Android	1200 €
Total	1500 €

Cuadro 5.2: Coste Software

El presupuesto final del proyecto con las horas de instalación del sistema electrónico, se ha estimado unas 4 horas, lo que conlleva soldar diferentes componentes, instalación en la moto y conectar la alimentación a la batería de dicha moto.

Montaje e instalación	120 €
Coste Hardware	265,03 €
Coste Software	1500 €
Total	1885,03 €

Cuadro 5.3: Coste total proyecto

Por lo que se estima un coste total del proyecto de 1885,03 euros. Sujeto a cambios debido a la posible variación de los costes en el mercado.

Anexo 3 Instalación Eclipse ADT

En este anexo voy a explicar como se debe instalar Eclipse ADT con SDK de Android en la versión r20, versión que he usado para el desarrollo de la aplicación móvil MotoSafe.

Los tres pasos fundamentales a seguir para tener el entorno de desarrollo instalado en nuestro ordenador son los siguientes:

- Descargar Android SDK.
- Descargar las últimas herramientas del SDK y plataformas desde el Manager del SDK.
- Instalar el plugin ADT para Eclipse.

Para descargar la última versión disponible solo tenemos que acceder a la página web de Android Developer para descargar Eclipse ADT y seleccionar la version que queramos para nuestro ordenador.

Realmente Eclipse no se instala, sino que se descarga, se descomprime y ya lo tenemos listo para hacerlo funcionar. Podemos descargarlo a través del siguiente enlace <http://www.eclipse.org/downloads/>.

Una vez realizado este proceso debemos proceder a añadir plataformas y paquetes al Android SDK. Para ello nos iremos al directorio de Android SDK, y en la carpeta tools ejecutamos el archivo android.

Cuando haya terminado de buscar, se nos darán algunas opciones ya marcadas para instalar. Por defecto, se instalarán los drivers de USB para poder probar las aplicaciones en algún dispositivo físico con Android que tengamos y la última versión de Android hasta la fecha. Opción no muy aconsejable ya que suele ser la versión que mas problemas nos presentará.

A partir de aquí podremos escoger la versión API que deseamos instalar y con la cual trabajaremos para desarrollar nuestra aplicación. Debemos aceptar todo lo que contenga el paquete para que pueda ser instalado.

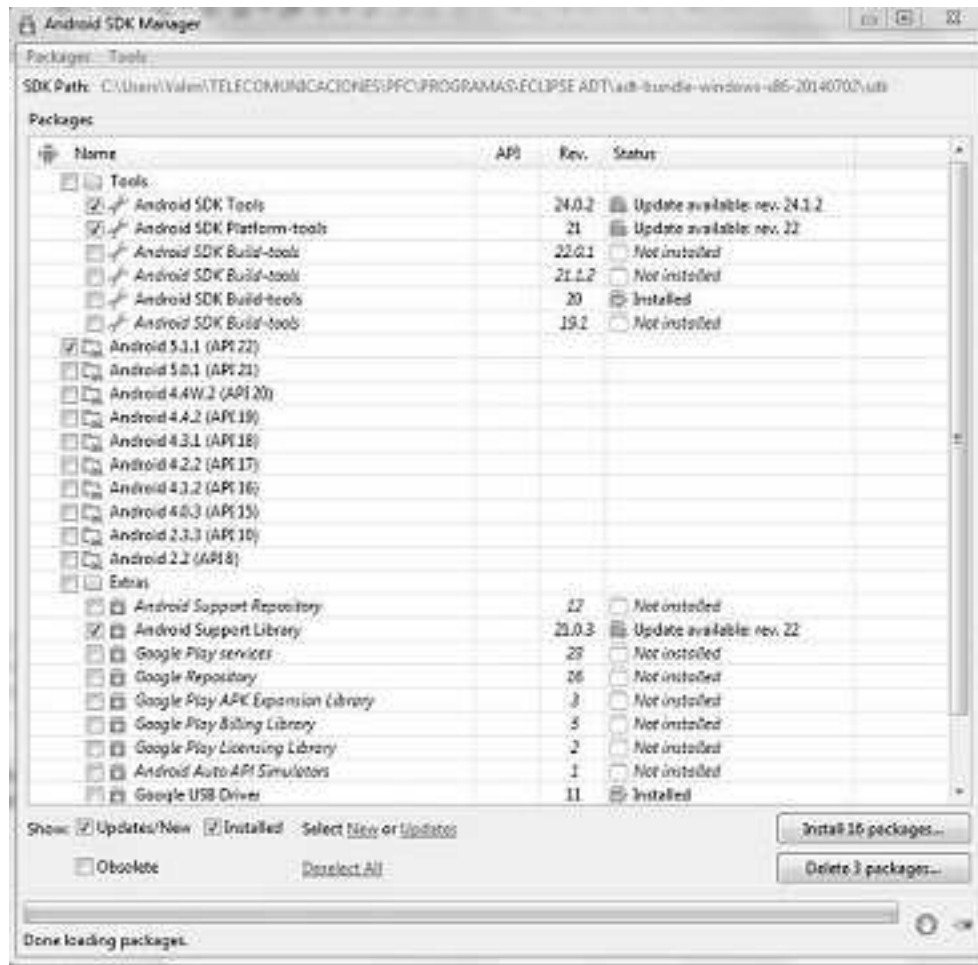


Figura 5.4: SDK Manager

Hay que destacar que SDK de Android y Eclipse son dos programas individuales, nada vinculados. SDK de Android es la ventana que he mostrado en la imagen anterior y que como puede verse es un asistente de descargas. Mientras que Eclipse es la interfaz sobre la cual programaremos. Debemos vincularlos para poder programar en Android.

Ahora trabajaremos con Eclipse, hay que vincular el SDK a nuestro entorno de desarrollo. Esto se realiza mediante la instalación de un plugin para Eclipse denominado ADT (Android Development Tools). Tiene lógica que el plugin sean las herramientas de desarrollo de Android, ya que Eclipse como entorno de trabajo requiere de estas herramientas que no tiene para programar en Android.

Este modo es un muy entretenido, por lo que recomiendo descargarse Eclipse ADT desde la página de Android Developers y evitar un trabajo extra.

He de destacar que en este caso instalé la API de Android 4.4, ya que el dispositivo móvil en el que iba a probar la aplicación desarrollada dispone de dicha versión de Kernel de Android.

Con ello evitaremos diferentes problemas de versión, además al no instalar una de las últimas versiones de API de Android evitamos que surjan múltiples fallos a la hora de compilación o de versiones de Kernel.

