



**Universitat Autònoma
de Barcelona**

Gestión de mecanismos Arduino controlados
inalámbricamente por dispositivos Android

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Gestió
realitzat per
Adrián Fernández Medina

i dirigit per
Marta Prim Sabrià

Escola d'Enginyeria

Sabadell, *juny de 2013*

La sotasignat, **Marta Prim Sabrià**,
professora de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present
memòria
ha estat realitzat sota la seva direcció per

Adrián Fernández Medina

I per a que consti firma la present.
Sabadell, juny de **2013**

Signat: **Marta Prim Sabrià**

FULL DE RESUM – PROJECTE FI DE CARRERA DE L'ESCOLA D'ENGINYERIA

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| Títol del projecte: GESTIÓN DE MECANISMOS ARDUINO CONTROLADOS INALÁMBRICAMENTE POR DISPOSITIVOS ANDROID | |
| Autor: ADRIÁN FERNÁNDEZ MEDINA | Data: JUNIO 2013 |
| Tutora: MARTA PRIM | |
| Titulació: ENGINYERIA TÈCNICA EN INFORMÀTICA DE GESTIÓ | |
| Paraules clau (mínim 3) <ul style="list-style-type: none">• Català: seguretat inalàmbrica, seguretat amb Arduino, control d'Arduino amb Android, Alarma per Android.• Castellà: Seguridad inalámbrica, Seguridad con Arduino, control de Arduino con Android, alarma para Android.• Anglès: wireless security, safety with Arduino, Control of Arduino with Android, Alarm for Android. | |
| Resum del projecte (extensió màxima 100 paraules) <ul style="list-style-type: none">• Català: L'objectiu d'aquest projecte és el de realitzar la fabricació d'una alarma la qual es pot controlar des de qualsevol dispositiu Android, ja que la comunicació es realitza a través de la connexió Bluetooth entre una aplicació Android i una placa Arduino que realitza el control del mecanisme Hardware. A través de l'aplicació que s'instal·la al dispositiu mòbil, es pot realitzar tot el control del prototip, com la validació d'usuari, activació i desactivació de l'alarma i l'enllaç amb el mòdul Bluetooth.• Castellà: El objetivo de este proyecto es el de realizar la fabricación de una alarma la cual se puede controlar desde cualquier dispositivo Android, ya que la comunicación se realiza a través de la conexión Bluetooth entre una aplicación Android y una placa Arduino que realiza el control del mecanismo Hardware. A través de la aplicación que se instala en el dispositivo móvil, se puede realizar el control del prototipo, como la validación del usuario, activación y desactivación de la alarma y enlazar con el módulo Bluetooth.• Anglès: The objective of this project is to make an alarm which can be controlled from any Android device, because communication is done via Bluetooth connection between an Android application and an Arduino, that controls the Hardware mechanism. Through the application that is installed on the mobile device, you can control the Arduino Board for the user validation, activation and desactivation of the alarm and pair to the Bluetooth module. | |

Estructura de la memoria:

| | |
|-----------------------------------------------------------------|-----------|
| Capítulo 1: Introducción | 9 |
| 1.1 Introducción | 9 |
| 1.2 Motivaciones personales | 9 |
| 1.3 Estado del arte | 11 |
| 1.4 Estructura de la memoria: | 13 |
| Capítulo 2: Estudio de Viabilidad | 15 |
| 2.1 Introducción al Estudio de Viabilidad | 15 |
| 2.2 Tipología y palabras clave. | 15 |
| 2.3 Descripción. | 15 |
| 2.4 Objetivos del proyecto. | 16 |
| 2.5 Definiciones, acrónimos y abreviaciones. | 16 |
| 2.6 Partes interesadas. | 19 |
| 2.7 Producto y documentación del proyecto. | 19 |
| 2.8 Estudio de la situación actual | 20 |
| 2.8.1 Contexto..... | 20 |
| 2.8.2 Diagnóstico del sistema. | 20 |
| 2.9 Usuarios y/o personal del sistema. | 21 |
| 2.10 Normativas y legislación. | 21 |
| 2.11 Requisitos del sistema | 21 |
| 2.11.1 Requisitos funcionales..... | 21 |
| 2.11.2 Requisitos no funcionales | 22 |
| 2.11.3 Restricciones del sistema..... | 22 |
| 2.11.4 Catalogación y priorización de los requisitos..... | 22 |
| 2.12 Alternativas y selección de la solución | 23 |
| 2.12.1 Contratación Alarma Securitas Direct:..... | 23 |
| 2.12.2 Instalación alarma Prosegur Promobile :..... | 23 |
| 2.12.3 Instalación alarma Arduino gestionada por Android: | 24 |
| 2.12.4 Solución propuesta para las alternativas:..... | 24 |
| 2.13 Conclusiones | 24 |
| Capítulo 3: Plan de Proyecto | 25 |
| 3.1 Introducción. | 25 |
| 3.2 Descripción. | 25 |
| 3.3 Definiciones, acrónimos y abreviaciones. | 25 |
| 3.4 WBS (Work Breakdown Structure) | 26 |
| 3.4.1 Fases y actividades del proyecto..... | 26 |
| 3.4.2 Diagrama WBS. | 27 |
| 3.4.3 Milestones..... | 28 |
| 3.5 Recursos del proyecto. | 28 |
| 3.5.1 Calendario de recursos..... | 28 |
| 3.6 Calendario del proyecto. | 29 |
| 3.6.1 Dependencias. | 29 |
| 3.6.2 Calendario temporal..... | 29 |
| 3.7 Evaluación de riesgos. | 30 |
| 3.7.1 Lista de riesgos. | 30 |
| 3.7.2 Catalogación de riesgos..... | 30 |
| 3.7.3 Plan de contingencia. | 30 |

| | |
|--------------------------------------------------------------------------------------------|-----------|
| 3.8 Presupuesto | 31 |
| 3.8.1 Estimación del coste de personal | 31 |
| 3.8.2 Estimación del coste de los recursos..... | 31 |
| 3.9 Conclusiones del plan de proyecto | 31 |
| Capítulo 4: Fundamentos Teóricos | 33 |
| 4.1 Comunicación inalámbrica : | 33 |
| 4.2 Ondas electromagnéticas : | 34 |
| 4.2.1 Modulación de onda: | 35 |
| 4.2.2 Amplitud modulada y frecuencia modulada : | 36 |
| 4.2.3 Comunicaciones terrestres : | 36 |
| 4.3 Arduino : | 37 |
| 4.3.1 “Hola mundo” Arduino :..... | 41 |
| 4.4 Qué es Android : | 42 |
| 4.4.1 Desarrollar Android :..... | 43 |
| 4.4.2 Hola mundo Android : | 43 |
| Capítulo 5: Alarma Arduino controlada inalámbricamente por dispositivos Android ... | 49 |
| 5.1 Parte mecánica (HARDWARE) | 49 |
| 5.2 Parte no mecánica (SOFTWARE) | 57 |
| Capítulo 6: Test y pruebas | 67 |
| 6.1 Arduino: | 67 |
| 6.2 Android: | 68 |
| Capítulo 7: Conclusiones | 69 |
| 7.1 Objetivos: | 69 |
| 7.2 Objetivos no cumplidos: | 70 |
| 7.3 Posibles mejoras : | 71 |
| 7.4 Conclusiones personales :..... | 71 |
| Bibliografía: | 73 |

Índice de ilustraciones:

| | |
|-----------------------------------------------------------------|----|
| <i>Ilustración 1: Diagrama de Gant</i> | 29 |
| <i>Ilustración 2: aspecto onda electromagnética</i> | 34 |
| <i>Ilustración 3: espectro electromagnético</i> | 35 |
| <i>Ilustración 4: esquema placa Arduino Uno</i> | 38 |
| <i>Ilustración 5: Sketch Blink en Arduino</i> | 41 |
| <i>Ilustración 6: aplicación App Inventor</i> | 43 |
| <i>Ilustración 7: árbol de carpetas Eclipse</i> | 44 |
| <i>Ilustración 8: clase MainActivity de Eclipse</i> | 44 |
| <i>Ilustración 9: carpeta Values de Eclipse</i> | 45 |
| <i>Ilustración 10: carpeta layout Eclipse</i> | 45 |
| <i>Ilustración 11: main.xml de la carpeta layout Eclipse</i> | 46 |
| <i>Ilustración 12: AndroidManifest.xml de Eclipse</i> | 46 |
| <i>Ilustración 13: emulador Android en Eclipse</i> | 47 |
| <i>Ilustración 14: kit inventor Arduino</i> | 50 |
| <i>Ilustración 15: aplicación Fritzing</i> | 51 |
| <i>Ilustración 16: altavoz con interruptor de lengüeta</i> | 52 |
| <i>Ilustración 17: shield Bluetooth para Arduino</i> | 53 |
| <i>Ilustración 18: proyecto en ejecución</i> | 53 |
| <i>Ilustración 19: aplicación solicitando permiso Bluetooth</i> | 54 |
| <i>Ilustración 20: aplicación S2 Bluetooth</i> | 55 |
| <i>Ilustración 21: menú aplicación S2 Bluetooth</i> | 55 |
| <i>Ilustración 22: S2 Bluetooth conectando a un dispositivo</i> | 56 |
| <i>Ilustración 23: S2 Bluetooth conectando a iTeed</i> | 56 |
| <i>Ilustración 24: S2 Bluetooth conectado a iTeed</i> | 56 |
| <i>Ilustración 25: mostrando mensaje de alarma conectada</i> | 57 |
| <i>Ilustración 26: validación App del proyecto</i> | 62 |
| <i>Ilustración 27: error de validación app proyecto</i> | 63 |
| <i>Ilustración 28: solicitud de permiso App proyecto</i> | 63 |
| <i>Ilustración 29: menú App proyecto</i> | 64 |
| <i>Ilustración 30: mensaje para enlazar</i> | 64 |
| <i>Ilustración 31: mensaje de alarma desconectada</i> | 64 |
| <i>Ilustración 32: mensaje alarma conectada</i> | 65 |

Agradecimientos :

Me gustaría dedicar las primeras líneas de esta memoria a agradecer el apoyo de la gente que me ha ayudado a tirar para adelante este proyecto, gente que ha aguantado mis nervios y me ha entendido, personas que desinteresadamente han contribuido con su granito de arena para avanzar y tirar adelante los propósitos iniciales de este proyecto tan importante para mi, y a esas otras personas como mis padres que me han ayudado no sólo ahora, sino durante toda la carrera a no tirar la toalla en los momentos más difíciles, en momentos donde me he sentido bloqueado y han sabido desatascarme y han ayudado a no rendirme. También me gustaría dedicar unas palabras para agradecer a Marta Prim, mi tutora, por facilitarme mucho el proceso de realización del proyecto, ya que ha hecho que la distancia no sea un inconveniente, y me ha ayudado a salir de algún apuro con su colaboración en algún que otro momento, y por supuesto a mi jefe, Óscar Fábregas, que no me ha puesto ninguna pega a la hora de escaparme y cogerme tantos vuelos como hiciera falta para poder hacer tutorías, entregas, o lo que hiciera falta para que todo saliera bien.

A todos vosotros, muchísimas gracias.

Capítulo 1: Introducción

Durante las siguientes líneas se explicará el motivo de la decisión de realizar este proyecto, motivaciones e inquietudes que me ayudaron a decantarme por un proyecto híbrido, combinando parte mecánica hardware con otra parte fundamental para la comunicación, como es la parte software. Y porque va relacionado sobre un tema como puede ser la seguridad combinada con la comunicación inalámbrica.

1.1 Introducción

El funcionamiento es muy sencillo, se trata de activar la alarma casera mediante un código enviado con una aplicación desarrollada en Android por el dispositivo móvil, de forma inalámbrica y utilizando Bluetooth. La alarma sonará siempre y cuando el campo magnético del interruptor de lengüeta active el “buzzer”. Para desactivar la alarma el mecanismo será el mismo, enviar el código para que el programa que utiliza el hardware no envíe corriente para activar el altavoz y así producir el sonido.

Este es el tema principal por el cual se guiará el proyecto, a medida que se vaya desarrollando, se intentará introducir más novedades y funcionalidades para hacer el proyecto sea más atractivo y a su vez más eficiente.

El título de proyecto indica el objetivo fundamental del mismo, el cual no es otro que el de poder realizar la activación o desactivación de una alarma casera mediante un teléfono móvil. Todo esto se realizará mediante hardware y software libre, ya que es uno de los motivos por los cuales decidí escoger este tema.

1.2 Motivaciones personales

Hace no más de un año me empecé a interesar por la energía libre, por la filosofía con la que soñaba Nicola Tesla y con la curiosidad de saber el funcionamiento de las cosas, especialmente dispositivos eléctricos.

Toda esta inquietud me llevó a desmontar los aparatos domésticos para intentar entender su funcionamiento. Al poco tiempo descubrí que se pueden realizar dispositivos muy complejos o de una gran utilidad con complementos electrónicos muy básicos, siempre y cuando se entienda la filosofía o los principios que utiliza.

Así que decidí buscar la manera para poder empezar a crear algún artilugio eléctrico, con unos conocimientos bastante precarios como eran los míos.

Descubrí la plataforma Arduino, que como comentaba anteriormente es un proyecto de Hardware libre, es decir, ellos distribuyen una serie de módulos o placas y el usuario los puedes modificar o combinar según las especificaciones técnicas que ofrece el fabricante. Arduino también incluye una serie de librerías que ayudan a hacer la integración con diferentes módulos de otras marcas, o como es nuestro caso con un dispositivo móvil controlado por Android.

La placa Arduino incluye una serie de componentes micro electrónicos, como diodos, diodos LEDS, resistencias, memoria y un micro controlador que te permite realizar circuitos sin necesidad de tener muchos conocimientos, y controlar esos dispositivos a través de unos pines analógicos y digitales que incluidos en la placa.

A raíz de aquí, comprobé que se podían adquirir diferentes módulos adaptables a la placa Arduino para su posterior uso, como pueden ser, sensores o adaptadores extra.

En este proyecto, se hace uso de una Shield (armadura) adaptable a la placa Arduino, de tal manera que pueda realizar una comunicación Bluetooth entre dos dispositivos, a lo largo de la memoria, se explica como se ha utilizado.

Una de las metas que se propuso Nicola Tesla, fue la de traspasar energía o información de manera inalámbrica, al final lo consiguió, y es una de las cosas o descubrimientos que más me ha impactado y a la vez motivado. Como Arduino ofrece esa oportunidad a través de sus módulos extensibles, se utilizará un módulo Bluetooth para recibir del dispositivo móvil el código de activación. Por tanto ya unimos mis objetivos de realizar un proyecto electrónico con parte Hardware, parte Software y transmisión inalámbrica de datos.

Así que para darle uno uso real, se ha pensado en la seguridad y en la comodidad del usuario a la hora de activar o desactivar la alarma de una manera inalámbrica, y a través de una aplicación totalmente realizada de cero, el cual me ha parecido un proyecto muy interesante, útil, divertido y muy motivador.

Espero realizar para un futuro varios proyectos relacionados con las funcionalidades que ofrece Arduino, ya que creo que se le puede sacar mucho partido para el día a día y para diferentes temáticas, como la seguridad, la calidad de vida o el confort, el ocio, etc.

1.3 Estado del arte

Hoy en día, la mayoría de los terminales de los cuales disponemos, no son para hacer o recibir llamadas, sino para funciones totalmente distintas como puede ser consultar el correo, navegar por internet, actualizar su estado en las redes sociales. Según un estudio realizado por Prew Research en 2011, España es el país con más teléfonos móviles por habitante, con un 96% por delante de países como E.U.A o China. Es por eso que las grandes empresas conocedoras de estos datos, intentan buscar soluciones tecnológicas para estos terminales móviles.

Además, cualquier invención tecnológica de ahora en adelante busca que el aparato en cuestión sea lo más diminuto posible y si es posible que no tenga que depender de cables o artilugios extra para su uso, por ello, la telefonía móvil está en auge.

Por otro lado, lo que buscan actualmente los usuarios, es no tener que preocuparse porque cada uno de los dispositivos que tiene en su poder haga una única función. Como puede ser el teléfono para llamar, el reloj para mirar la hora, el ordenador para consultar el correo, etc. Lo que se busca hoy en día es que la tecnología actual se preocupe por todas nuestras necesidades, para tener que hacer lo menos posible con la aparatología mínima. En este campo entra perfectamente la rama de la domótica, que no es otra cosa que aplicar la tecnología robótica en los hogares.

Por ejemplo, la comunidad de Madrid a través de un extenso PDF “Domótica como solución de futuro”, daba a conocer a sus habitantes en 2007, la intención de dotar de esta tecnología a las futuras construcciones de inmuebles, explicando los conceptos más básicos, así como de las mejoras que suponía para la calidad de vida. Con este documento, también se hacía un llamamiento a los profesionales que estuvieran interesados en aprender las bases de utilización y montaje, para futuras contrataciones y fomento del empleo.

Con esto, se puede apreciar que los gobiernos apuestan por el futuro, y además lo hacen como algo totalmente positivo, ya que aparte de apostar por las nuevas tecnologías, el ahorro de energía es muy elevado.

Aunque no sólo los gobiernos apuestan por implantar las tecnologías en el día a día de los habitantes, sino que las grandes empresas también lo hacen, y no sólo lo hacen para el sector de la telefonía móvil, sino que investigan para implantarlo en cualquier lugar donde sea eficaz, beneficioso y productivo. Un ejemplo muy claro de que la tecnología robótica está muy presente entre nosotros, es que la compañía de la manzana, “Apple”, en su reciente congreso para desarrolladores realizado este mes de Junio de 2013, dio cabida a una compañía que se dedica a la tecnología robótica y que tubo unos minutos para presentar un nuevo producto, “Anki Drive”, el cual desplegaba una lona con un circuito negro, donde al soltar una serie de cochecitos de juguete manipulados, podían circular entre sí sin sufrir ningún tipo de colisión, ya que éstos incorporan unos sensores capaces de vigilar sus alrededores 500 veces por segundo. Si Apple, la mayor compañía del mundo cree en algo así, es porque realmente va a tener un peso importante en un futuro no muy lejano. La web de la compañía es

www.anki.com. En el siguiente enlace se puede ver el video de la demostración del producto: <http://www.youtube.com/watch?v=o38DpmXnZn8>

Es por eso, que las tecnologías más vanguardistas están muy presentes en cualquier dispositivo, ya sea coches, hogares o dispositivos móviles. Retomando el contacto con la tecnología móvil, podemos decir que las próximas mejoras y funcionalidades que habrán en los dispositivos móviles son cuantiosas. Basándonos en el último congreso celebrado en las Vegas este mismo año sobre electrónica y tecnología de consumo, podemos deducir las tendencias para este 2013, las cuales no son otras que:

- Redes de alta velocidad (LTE), en España por ahora la única operadora que se ha lanzado ha sido Vodafone presentándolo al mercado este mismo mes de junio para las grandes capitales de España. En el mes de julio se añadirán otras operadoras como Orange. Con esta tecnología, que ya funciona en E.U.A y en Canadá, los usuarios podremos navegar hasta cuatro veces más rápido.
- Tecnología NFC para realizar pagos. Tal y como comentábamos anteriormente, se busca que un solo dispositivo pueda realizar las máximas funciones posibles, así que nos olvidaremos de llevar nuestra cartera con multitud de monedas para dejar paso a nuestro teléfono, para realizar los pagos. Durante 2012 ya hemos comprobado que se están gestando proyectos muy interesantes al respecto, como el proyecto piloto en la estación de cercanías de RENFE de Nuevos Ministerios, que permitirá el pago y acceso a los trenes de Cercanías mediante el móvil.

Por otro lado, con el fin de probar la aplicación de este estándar en el pago con móvil, la firma ha desarrollado en colaboración con el Banco Santander el primer sistema de pago NFC multimarca (VISA y MasterCard) que permite a sus usuarios realizar compras de cualquier importe con solo acercar su teléfono móvil al terminal punto de venta.

Esta tecnología tiene un importante potencial de mercado: se estima que en el mundo se habrán vendido alrededor de 1.500 millones de dispositivos con esta tecnología entre 2010 y 2016, con los que se realizarán transacciones globales por valor de 50.000 millones de dólares durante el mismo periodo, según datos de Strategy Analytics.

- Notificaciones Vía Push, con el uso de internet móvil superando el uso de Internet en los ordenadores de mesa, las aplicaciones móviles (Apps) proliferan a una velocidad de vértigo. Las notificaciones Push son una táctica emergente que está empezando a establecerse con fuerza entre aquellas marcas que quieren maximizar su conexión personal con el consumidor a través de las aplicaciones móviles.

Como se puede apreciar, todavía hay mucho por hacer en cuanto a implantar las tecnologías actuales en los distintos dispositivos, y por supuesto todavía hay mucho futuro por delante para investigar sobre ello.

En nuestro caso, es decir, el de la implantación de la tecnología para la seguridad del hogar, nos encontramos con una empresa líder en el sector, como es Securitas Direct, la cual lanzó al mercado una aplicación llamada My Verisure.

Con esta aplicación, el cliente puede conectar o desconectar su alarma estando fuera de casa, conocer el estado de la misma o controlar las horas concretas en las que la alarma ha sido activada o desactivada. Entre los aspectos que más destacan de My Verisure está la posibilidad de ver todo lo que ocurre dentro del hogar o del negocio sin necesidad de que salte la alarma, ni de que ésta esté conectada, ya que permite ver las imágenes de lo sucedido a través del smartphone o la Tablet. También ofrece otras funciones como el control de la domótica del domicilio.

La App My Verisure, disponible para iPhone, Android, Blackberry y Nokia, que puede descargarse en Google Play e iTunes Store, cuenta ya con un 46% de usuarios de iPhone y un 31% de usuarios de Android. Además, la aplicación ha tenido una gran acogida entre los clientes de Securitas Direct: el 63% de las descargas corresponde a clientes con una alarma instalada en el hogar y un 37% corresponde a clientes con alarma en negocios.

Es por eso que este tipo de aplicaciones están en un continuo cambio, y en una continua modificación para mejorar diferentes aspectos, y en este proyecto se ha querido hacer un pequeño homenaje plasmando los diferentes conceptos aprendidos a lo largo de la carrera, hasta llegar a un prototipo que se encarga de hacer la gestión de la seguridad del hogar de una manera inalámbrica.

1.4 Estructura de la memoria:

En este apartado se hace un pequeño anticipo sobre cada capítulo, a modo de introducción. Indicando los aspectos más importantes de cada uno de ellos.

Capítulo 1: Introducción

Se intenta plasmar la esencia del proyecto, explicando cuales han sido las motivaciones y objetivos que han llevado a elegir el tema del proyecto, así como una pequeña introducción para poner en antecedentes al lector.

Capítulo 2: Estudio de Viabilidad

Este apartado demuestra a través de sus diferentes capítulos, que el proyecto en cuestión es viable para su fabricación y uso, y se demuestra mediante una serie de descripciones, definiciones, comparaciones, etc. También se explican cuales van a ser las funcionalidades.

Capítulo 3: Plan de proyecto

Aquí se comenta como se va a llevar a cabo el proyecto, cuales son las fases de creación, calendario de realización, presupuestos, y los pasos que se van a seguir para su desarrollo.

Capítulo 4: Fundamentos teóricos

Aquí se da la oportunidad al lector de aprender los conocimientos necesarios para entender cualquier tema del cual trata el proyecto, es la base sobre los principales conceptos del mismo.

Capítulo 5: Alarma Arduino controlada inalámbricamente por Android

Este apartado toca de lleno el funcionamiento del proyecto, y como se ha llegado a crear tanto la parte mecánica como las partes dedicadas al software y a la creación de la aplicación para el terminal móvil, con capturas de pantalla y anotaciones.

Capítulo 6: Test y Pruebas

En el siguiente capítulo, se comentan las pruebas realizadas y problemas ocurridos hasta llegar a la solución final.

Capítulo 7 : Conclusiones

Este es el último capítulo, y consta de las últimas impresiones, objetivos cumplidos y no cumplidos, así como posibles mejoras para realizar en un futuro.

Bibliografía:

Aquí se añaden todos los links de interés, de los cuales se ha sacado información o se ha nombrado algún concepto de una página web en concreto.

Capítulo 2: Estudio de Viabilidad

2.1 Introducción al Estudio de Viabilidad

En el siguiente capítulo se explica el funcionamiento y el objetivo fundamental de este proyecto. Se quiere montar una alarma casera controlada por un dispositivo móvil con Android. La alarma se conectará a una placa Arduino para poder realizar un mayor control sobre todos los componentes a utilizar. Lo que se pretende es el control inalámbrico de la alarma como recientemente lo están haciendo diferentes compañías de seguridad (Securitas Direct o Prosegur), con la diferencia que este proyecto quiere también mantener el control de la activación, también inalámbrico y no por sensores RFID. Mediante el envío de un código enviado por un terminal Android se recibirá la información en la placa Arduino que ordenará la activación o no de la alarma integrada.

Durante las siguientes líneas se explican los fundamentos necesarios para que el proyecto sea viable, efectivo, eficaz, y con futuro.

2.2 Tipología y palabras clave.

Se puede identificar el proyecto como un proyecto de desarrollo. Las palabras clave que pueden ayudar a identificar dicho proyecto son: Alarma inalámbrica controlada por Android, Alarma controlada por dispositivo móvil, o control de la seguridad a través de dispositivos Android.

2.3 Descripción.

Hoy en día, en el contexto económico en el que nos encontramos, la seguridad conlleva un factor clave en nuestras vidas. Se multiplican las estafas, los robos y es una acción necesaria y casi obligatoria la de ampliar o introducir seguridad al menos en el hogar.

Es por eso, que este proyecto quiere implementar un sistema de control distinto al de las compañías de seguridad actuales, que avisa al usuario de presencias no autorizadas en lugares estratégicos de la casa, con los cuales se puede realizar un mayor control de la casa para conseguir mayor eficacia y rapidez a la hora de llamar a las fuerzas del orden público.

En la actualidad, la mayoría de los ciudadanos llevan siempre consigo dos cosas: la primera son las llaves, y la segunda es un teléfono móvil.

Es por eso, que se ha pensado que una manera fiable para realizar la activación de la alarma en nuestro proyecto sea un dispositivo que siempre esté con el usuario, en este caso, un teléfono con sistema operativo Android.

De esta manera se asegura la activación de la alarma o desactivación siempre que el usuario lo desee, simplemente con abrir la aplicación del teléfono e introducir el código adecuado.

2.4 Objetivos del proyecto.

Los objetivos de este proyecto, son los de detectar de una manera fiable y rápida las presencias no deseadas por el usuario. Con tal de agilizar la posible faena de las fuerzas y cuerpos del orden público previo a una llamada. También se comprueba que se facilita la activación de nuestra alarma, ya que se hace desde el mismo teléfono móvil con la aplicación Android, es decir, de forma inalámbrica.

Éstos son los objetivos principales del proyecto. Aprovechando que se utiliza una placa Arduino, a la cual se le pueden conectar diversos dispositivos o sensores para su posterior uso, posteriormente se verá si según criterios como presupuesto, límite de tiempo, etc., se pueden ir añadiendo módulos como una cámara para detectar la presencia extraña de movimiento o diferentes sensores de proximidad, temperatura, vibración para añadir nuevos objetivos al proyecto.

2.5 Definiciones, acrónimos y abreviaciones.

Arduino:

es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

Android:

es un sistema operativo móvil basado en Linux, que junto con aplicaciones middleware está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos.

Linux:

es un núcleo libre de sistema operativo basado en Unix. Es uno de los principales ejemplos de software libre. Linux está licenciado bajo la GPL v2 y está desarrollado por colaboradores de todo el mundo.

Unix:

es un sistema operativo portable, multitarea y multiusuario. Fue desarrollado en 1969, por un grupo de empleados de los laboratorios Bell de AT&T.

AT&T:

siglas de su nuevo nombre, American Telephone and Telegraph; es una compañía estadounidense de telecomunicaciones.

GPL:

es una licencia creada en 1989 (por Richard Stallman), está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Middleware:

es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos. Éste simplifica el trabajo de los programadores en la compleja tarea de generar las conexiones que son necesarias. De esta forma se provee una solución que mejora la calidad de servicio, seguridad, envío de mensajes, directorio de servicio, etc.

Software Libre:

es la denominación del software que respeta la libertad de todos los usuarios que adquirieron el producto y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado, y redistribuido libremente de varias formas.

Hardware Libre:

se llama hardware libre a los dispositivos cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita.

Bluetooth:

es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en los 2,4 GHZ.

Buzzer:

Altavoz utilizado para dispositivos electrónicos o micro electrónicos.

Electrónica:

es la rama de la física y especialización de la ingeniería, que estudia y emplea sistemas cuyo funcionamiento se basa en la conducción y el control del flujo microscópico de los electrones u otras partículas cargadas eléctricamente.

Sketch:

Es la denominación que obtiene cada aplicación creada para la placa Arduino.

Push:

La tecnología Push es un tipo de comunicación en la que es el servidor el que inicia la petición al cliente (el móvil, por ejemplo) cuando tiene una información o notificación nueva, permitiendo un importante ahorro de recursos y tiempo respecto a la tecnología convencional.

Wiring:

una plataforma de hardware multipropósito con un ambiente de desarrollo para no programadores creado por Hernando Barragán.

Processing:

es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital.

2.6 Partes interesadas.

Por lo que respecta a la creación del proyecto, se pueden identificar diversos grupos que participan en él. Podemos diferenciar el proyecto en partes diferentes, concretamente en cuatro partes:

La primera parte podríamos decir que se trata del apartado hardware.

El técnico en cuestión se encarga de este apartado, donde se realiza el montaje del prototipo y de las pruebas necesarias para un correcto funcionamiento, independientemente de la parte inalámbrica, ya que pertenece a otro grupo distinto.

La segunda parte se encarga del apartado software, el programador en cuestión se debe dedicar de la creación de código necesaria para la comunicación de la parte hardware con la software, así como las funciones correspondientes para el buen uso de la aplicación. También debe procurar crear la aplicación, o mejor dicho el sketch de Arduino para realizar las funciones indicadas.

La tercera parte es el director del proyecto, en este caso el tutor, el cual se encarga de guiar el proyecto, y guiar las fases aportando correcciones, modificaciones o ideas sobre el proyecto.

La cuarta y última parte del proyecto participa directamente el usuario, ya que es quien le da utilidad al programa y finalmente es el que probará la eficacia de todas las partes implicadas.

2.7 Producto y documentación del proyecto.

Se hará entrega de una memoria del proyecto en cuestión. Además, en el capítulo 4, se describe el uso de la aplicación Android, a modo de manual de usuario.

Para finalizar se realizará una prueba empírica para demostrar el funcionamiento del prototipo.

2.8 Estudio de la situación actual

2.8.1 Contexto.

Inicialmente, se tuvieron que hacer pruebas mediante software externo, como son los emuladores o la aplicación Fritzing, debido a la falta de componentes y módulos necesarios para llevar a cabo el proyecto.

Las pruebas se realizaban íntegramente con el ordenador, con necesidad obligatoria del cable USB (Universal Serial Bus) de datos para alimentar la placa Arduino y a su vez para transmitir la información necesaria.

A medida que se iban recibiendo los componentes adicionales, como son el módulo Bluetooth o el alimentador de corriente de la placa de Arduino, el ordenador pasaba a un segundo plano, en el cual sólo se utilizaba para plasmar los datos o anotar posibles detalles a mejorar.

Se tubo que hacer uso de dos módulos Bluetooth distintos, ya que el primero de ellos no funcionaba correctamente, hasta lograr llegar a los resultados con el módulo de iTeed, el cual es el que tenemos actualmente para nuestro prototipo.

Finalmente, se utilizará un alimentador externo no portátil para la placa Arduino, debido a que la batería externa demandada no llegará a tiempo para el final de la entrega del proyecto, con esto, logramos alimentar la placa Arduino pero de forma que tenemos la necesidad de conectarla a un enchufe cercano.

2.8.2 Diagnóstico del sistema.

Los problemas actuales del proyecto son la falta de recursos económicos que nos impiden por ahora conseguir una placa Arduino con el ADK de Google preinstalado, lo que nos facilitaría mucho las cosas a la hora de programar, pero el precio de esa placa triplica el precio de la placa actual.

Otro de los problemas que se pueden llegar a originar son los ocasionados por el traspaso de información inalámbrica ya que todavía no se han realizado las pruebas pertinentes.

Para paliar los posibles problemas se podría realizar una inversión en elementos más precisos para integrar mejor los componentes del prototipo.

2.9 Usuarios y/o personal del sistema.

En este caso especial, al tratarse de un proyecto final de carrera individual, sólo se cuenta con dos personas para llevar a cabo el prototipo.

En este caso son:

Adrián Fernández Medina, el cual se encargará de las funciones software (Programador) tanto para el sketch que llevará cargada la placa Arduino y que se comunicará con el dispositivo móvil, como para la aplicación Android realizada en java, que enlazará con el módulo Bluetooth de Arduino y permitirá realizar el uso del prototipo de una manera muy sencilla e intuitiva para el usuario .

Por otra parte, se encuentra la directora del proyecto que en este caso es Marta Prim Sabrià, la cual se encarga de la supervisión y autorización de las distintas fases del proyecto, así como de analizar las posibles mejoras y corrección de errores que estime oportuna.

2.10 Normativas y legislación.

Actualmente no disponemos de ningún problema con ninguna norma o ley que nos modifique el funcionamiento actual de nuestro proyecto, siempre y cuando no conectemos nuestra alarma a una central, en caso contrario haríamos cumplir la siguiente ley.

<http://www.seguridad-privada.net/ley-seguridad-privada.html>

La cual hace referencia a la seguridad privada.

2.11 Requisitos del sistema

Pasamos a describir los elementos necesarios para llevar a cabo nuestro proyecto y los problemas que nos van surgiendo a lo largo del montaje.

2.11.1 Requisitos funcionales.

Estas son las funciones que presenta el proyecto :

- (1) Validación de usuario para el control de la aplicación .
- (2) Conexión del dispositivo Android con el Módulo Bluetooth .
- (3) Envío de la clave de activación y desactivación inalámbricamente.
- (4) Activación de la alarma con el dispositivo Android mediante Bluetooth.
- (5) Desactivación de la alarma por el mismo método.
- (6) Activación del Altavoz cuando la alarma se conecta sin autorización.
- (7) Activación de una secuencia de LEDS cuando se activa la alarma.
- (8) Activación de otra secuencia de LEDS cuando la alarma no está activada.
- (9) Desconexión del enlace con el módulo Bluetooth.
- (10) Control de errores para situaciones no esperadas.

2.11.2 Requisitos no funcionales.

Si bien es cierto que al inicio del proyecto existían requisitos no funcionales, en la actualidad podemos decir que no podemos enumerar ningún requisito no funcional, sólo podemos realizar mejoras de los requisitos funcionales o ampliarlos.

2.11.3 Restricciones del sistema.

- (1) La aplicación para el dispositivo móvil se desarrollará en Java.
- (2) La aplicación está orientada para dispositivos Android.
- (3) La placa Arduino tiene que estar alimentada por la corriente, con una pequeña batería de 12V, o mediante cable USB a cualquier PC o Mac.
- (4) El envío de información se debe realizar dentro del alcance del Bluetooth

2.11.4 Catalogación y priorización de los requisitos.

En la siguiente Tabla 1, se explican las prioridades para los requisitos funcionales y no funcionales, donde se catalogan como esenciales si son obligatorios de realizar, condicionales si dependen de otro requisito, o opcional en el caso de que su no realización no afecte negativamente al proyecto.

Tabla 1: prioridades requisitos funcionales.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|--------------------|---|---|---|---|---|---|---|---|---|----|
| <i>Esencial</i> | | X | X | | | X | | | | X |
| <i>Condicional</i> | | | | | | | X | X | X | |
| <i>Opcional</i> | X | | | X | X | | | | | |

2.12 Alternativas y selección de la solución

2.12.1 Contratación Alarma Securitas Direct:

http://www.securitasdirect.es/landing/generico.html?gclid=CJSLwr2_xbMCFYXLtAodHmwASg&source=adwords#null

Se podría realizar la instalación de una alarma controlada por una empresa privada, en este caso concreto por Securitas Direct. La cual nos ofrece una aplicación gratuita para realizar el control de la alarma o incluso de la casa, en cuanto a visualizar la cámara por un dispositivo móvil activación y desactivación directamente desde el terminal móvil, muchas de las funciones de la aplicación son de pago. La activación y desactivación también se puede realizar mediante controladores RFID que son entregados por la compañía. La ventaja de este tipo de alarmas son que están conectadas con la policía mediante una centralita de llamada automática. Tiene un coste de 99€ más 36 pagos de 15€, I.V.A no incluido.

2.12.2 Instalación alarma Prosegur Promobile :

<http://www.alarmahogarprosegur.com/?gclid=CKOQ2MnCxbMCFSHtAodSHgAjw>

Instalación de una alarma controlada por una empresa privada, en la cual se tiene control inalámbrico de la alarma por dispositivos móviles pero sólo para realizar la activación y desactivación, también se pueden hacer peticiones de imagen, las cuales nos envían imágenes a tiempo real de lo que capta la cámara, las peticiones son limitadas o de pago. Al igual que la anterior compañía, también tiene conexión directa con la policía. La descarga de la aplicación es gratuita y multiplataforma, pero hay un coste mensual por uso de 2€. También tiene algunas ventajas interesantes como son los detectores de grabación nocturna y que la instalación es sin cables y con batería.

Hay una permanencia de dos años, cuota de alta por ahora desconocida y no indicada en la web, el coste mensual es de 1€ al día hasta el 30 de junio de año 2013, la cuota de alta y el IVA no están incluidos.

2.12.3 Instalación alarma Arduino gestionada por Android:

La gran baza de este proyecto sobre cualquier otro, radica en que es una alarma totalmente móvil y portátil, ya que colocándole una batería externa es totalmente manejable al antojo del usuario, debido a que casi no existe instalación. Otro punto importante, es que la aplicación no es pública, es totalmente única y exclusiva de cada cliente, por lo cual se puede adaptar a sus necesidades, creando o eliminando funcionalidades. Debido a que la alarma está integrada con Arduino, se pueden crear multitud de aplicaciones y usos, como aperturas y cierres de mecanismos mediante servos, o diferentes tipos de controles mediante sensores de luz, temperatura, movimiento.

2.12.4 Solución propuesta para las alternativas:

La solución propuesta es la alternativa número 3, debido a la multitud de posibilidades que ofrece sobre sus competidoras. El hecho de poder realizar diferentes funcionalidades gracias a la placa Arduino, hace que sea una solución muy potente. No tiene conexión directa con la central de alarmas pero se puede llegar a programar diferentes requisitos que puedan llegar a suplir esta función. Otro de los puntos clave, es el de ser una alarma de “quita y pon”, es decir, que siempre que se quiera y gracias a la batería que se puede conectar, se puede realizar la conexión de alarma en cualquier puerta, ventana o cajón siempre que se quiera.

Las otras alternativas comentadas, tienen muy buenas características para su uso, de hecho son las que están actualmente en el mercado. Pero con la alternativa número 3 se quiere dar un punto de vista distinto en cuanto a seguridad se refiere.

2.13 Conclusiones

Las conclusiones a las cuales podemos llegar son que el proyecto es 100% viable. Ya que a día de hoy existe algo parecido en el mercado que funciona y se vende como una novedad importante dentro de la seguridad para el hogar.

La potencia del proyecto radica en lo moldeable que puede llegar a ser en cuanto a funcionalidades, tal como comentábamos anteriormente. Sólo depende del ingenio del desarrollador y de los objetivos que quiera obtener el cliente para tener un control sobre su domicilio, despacho, etc.

También podemos especializar y modificar la aplicación del dispositivo móvil al gusto del cliente, cosa que las alternativas no lo pueden hacer debido a que la aplicación es Standard y no reprogramable. En definitiva el proyecto es un concepto nuevo de seguridad al darle las opciones al cliente para controlar su casa según sus intereses.

Capítulo 3: Plan de proyecto

3.1 Introducción.

Durante las siguientes páginas se describirán las diferentes fases y actividades que se seguirán en este proyecto. Se especificarán los pasos que se llevarán a cabo para detallarlo todo, finalmente en la conclusión introduciremos conclusiones del proyecto para ponerlo en marcha. Esto ayudará a tomar una decisión de si poner o no en marcha el proyecto según los comentarios y anotaciones que se irán sacando durante la fase de creación del plan de proyecto.

3.2 Descripción.

El plan de proyecto de “Gestión de mecanismos Arduino controlados inalámbricamente por Android” describe los diferentes pasos que se seguirán para controlar y desarrollar nuestro proyecto. Son básicos para diferenciar las fases y actividades, y podemos controlar el calendario, presupuestos, y los recursos precisos.

Se utiliza una metodología lineal para el plano del proyecto, ya que las fases y actividades no se pueden desarrollar a la vez, sino que acaba una y comienza la otra.

3.3 Definiciones, acrónimos y abreviaciones.

Hardware libre:

se refiere a los dispositivos los cuales sus especificaciones son de acceso público, ya sea sobre alguna forma de pago, o gratuita.

Banda ISM:

(Industrial, Scientific and Medical) son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética per uso industrial, científico o médico. En la actualidad han sido popularizadas para uso en comunicaciones inalámbricas como el Bluetooth.

GHz:

(Gigahertz) es un múltiple de la unidad de medida Hertz, equivale a 1.000.000.000 Hz, por tanto tiene un período de oscilación de 1 nano segundo.

WBS:

Work Breakdown Structure.

Milestone:

Punto de control.

Diagrama de Gantt:

Cronograma del proyecto.

3.4 WBS (Work Breakdown Structure)

3.4.1 Fases y actividades del proyecto.

Iniciación:

incluye actividades de definición del proyecto, asignación de tutores y matriculación.

Planificación:

incluye estudios de viabilidad y plan de proyecto con los cuales se tomará la decisión final de la continuación del proyecto.

Análisis:

análisis de los requisitos funcionales i no funcionales.

Diseño:

incluye el diseño del proyecto hardware así como el software. Todo lo que tiene que ver con la construcción del prototipo tanto en el ámbito físico, como en el ámbito de la programación.

Desarrollo:

desarrollo de las funcionalidades hardware para después pasar a la aplicación para el dispositivo inalámbrico.

Test y Pruebas:

fase de prueba y correcciones de errores en el caso que hayan pequeños problemas que dificulten la aprobación por parte del técnico.

Generación de documentos:

fase de documentación del proyecto, incluye manuales sobre la utilización de la aplicación y una memoria explicativa.

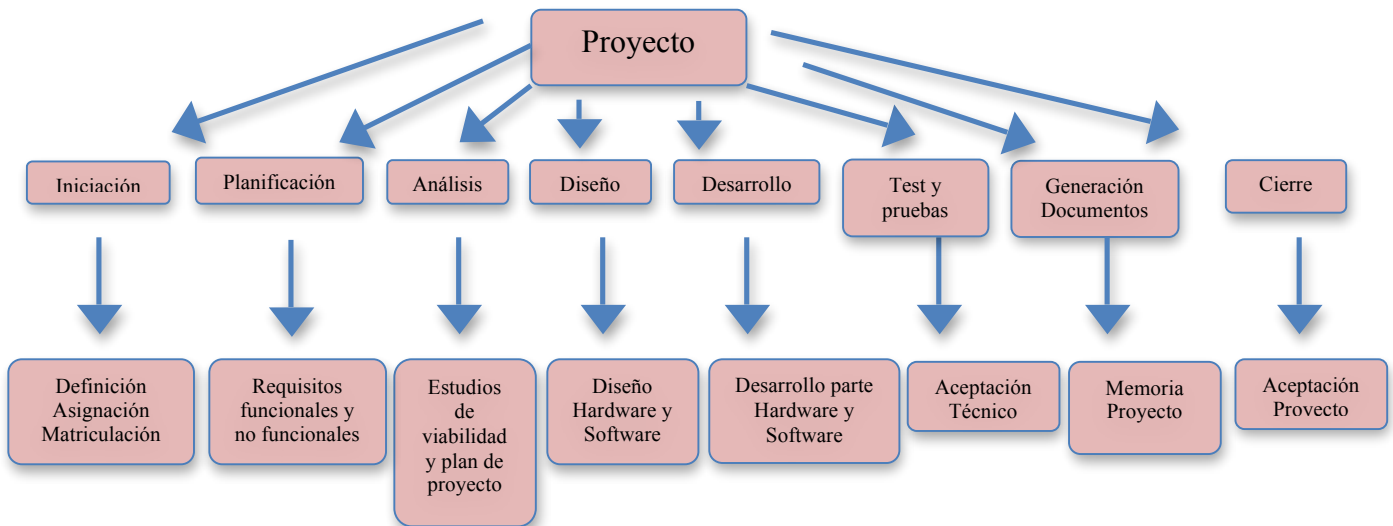
Cierre del proyecto:

fase de cierre, el director de proyecto firma y acepta el cierre del proyecto.

Defensa del proyecto:

defensa del proyecto delante de la comisión.

3.4.2 Diagrama WBS.



3.4.3 Milestones.

Tabla 2: Tabla de milestones o puntos clave

| NOMBRE | DESCRIPCIÓN | FECHA |
|-----------------------|-----------------------|--------------|
| Iniciación | Matriculación | 2/10/2012 |
| Estudio de Viabilidad | Aprobación | 5/11/12 |
| Pla de Proyecto | Aprobación | 26/11/12 |
| Diseño | Bases del proyecto | 10/12/12 |
| Desarrollo | Puesta en marcha | 10/01/13 |
| Test ,Pruebas | Demostración | 4/03/13 |
| Generación Documentos | Redacción de escritos | 26/4/13 |
| Cierre Proyecto | Aprobación | Junio 2013 |
| Defensa del Proyecto | Valoración | Julio 2013 |

3.5 Recursos del proyecto.

En este apartado se explican los recursos que hacen falta per llevar a cabo el proyecto, materiales, personal, calendario, etc.

Por lo que respecta al personal, necesitamos un tutor, un desarrollador, un técnico y un analista.

A diferencia de las funciones del tutor, el resto de los recursos se pueden desarrollar por la misma persona en este caso especial para el proyecto final de carrera, de forma totalmente gratuita.

Por otra parte, los materiales necesarios para realizar el proyecto son, la placa Arduino UNO, conjuntamente con el “kit inventor”, un módulo Bluetooth para la comunicación inalámbrica, alimentación para la placa Arduino mediante transformador o batería y un terminal móvil con sistema operativo Android. La parte de software es totalmente gratuita ya que está desarrollada sobre libre licencia en Java por el sistema operativo Android.

3.5.1 Calendario de recursos.

El tutor está disponible durante todo el proyecto y hará las funciones de supervisor, la función de desarrollador sólo está en la fase de desarrollo ya que se encarga de llevar a cabo la creación de los programas de comunicación entre los aparatos, la función de analista sólo forma parte del proyecto en la fase de análisis en el momento de poner las bases para la creación del software, para acabar la función de técnico, sólo se trabajará en la fase de desarrollo, test y pruebas para dar el visto bueno al correcto seguimiento y cumplimiento del proyecto.

3.6 Calendario del proyecto.

El proyecto está planteado para que tenga una duración de seis meses efectivos. Con una dedicación diaria entre semana de dos horas. La fecha para el comienzo es el 1 de octubre de 2012. La fecha esperada para la finalización es el 30 de marzo de 2013. La duración estimada es de unas 240 horas, repartidas en las diferentes jornadas en cuanto a investigación, desarrollo, pruebas, tutorías, etc. Se hará uso de programas de gestión de proyectos como es el caso de Microsoft Project en su versión 2010.

3.6.1 Dependencias.

En este proyecto las tareas no son independientes, es decir, cada tarea debe comenzar cuando acaba la anterior. No podemos comenzar una tarea nueva sin la totalidad de una anterior, ya que se dependen de sus resultados o de material físico.

Primeramente se comenzará con la iniciación y planificación, posteriormente se pasará al diseño y desarrollo, test y pruebas y finalmente se pasará a documentar los resultados y la información.

3.6.2 Calendario temporal.

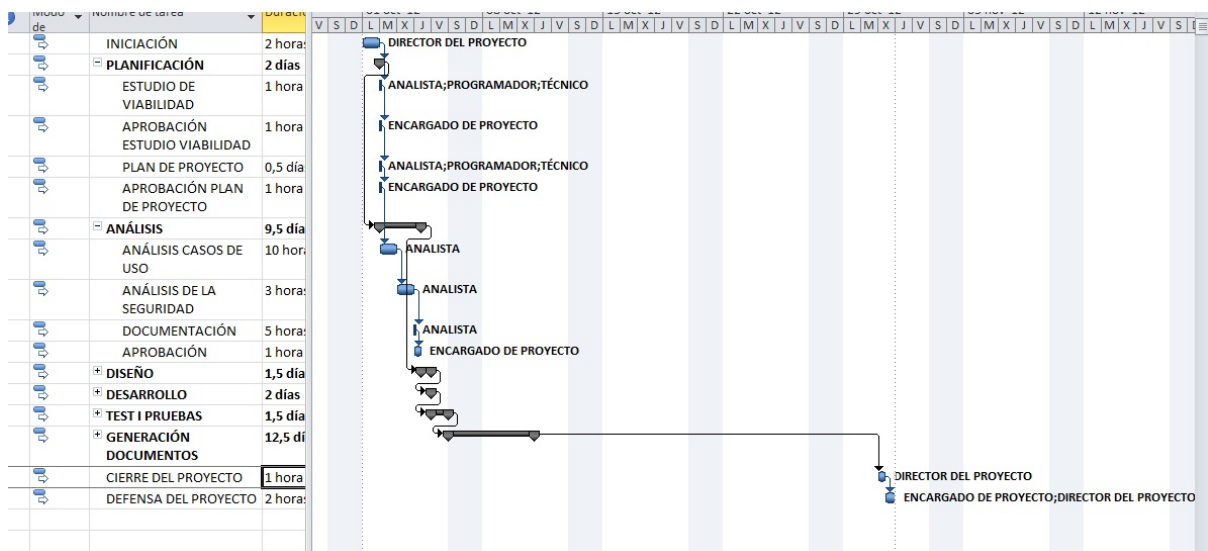


Ilustración 1: Diagrama de Gant

3.7 Evaluación de riesgos.

En este apartado lo que tenemos en cuenta son los posibles problemas que pueden aparecer a la hora de realizar las distintas tareas. A continuación se comentan.

3.7.1 Lista de riesgos.

- 1- En el análisis del proyecto pueden aparecer problemas en los costes no funcionales del personal integrante del proyecto, si el tiempo especificado no es correcto.
- 2- En la fase de pruebas y de test, se puede ralentizar el proyecto por el no funcionamiento en la fase de pruebas.
- 3- La no aprobación de las fases por el encargado del proyecto puede provocar retrasos.

3.7.2 Catalogación de riesgos.

1. MUY GRAVE : se debe cumplir el tiempo para no añadir costes innecesarios.
2. LEVE: se debe realizar las fases anteriores correctamente para no tener que retroceder llegados a este punto, aunque los errores en las fase de pruebas se debe tomar como característica positiva para la corrección
3. GRAVE: la no aprobación de las fases por el encargado viene a determinar que la fase de pruebas se ha pasado con errores por el técnico especializado, lo cual es un atraso grave.

3.7.3 Plan de contingencia.

En el caso del riesgo "1", se tienen que tomar las medidas iniciales necesarias y con una aprobación de todos los recursos que formarán parte del proyecto para tener un compromiso en el cumplimiento de los plazos.

En el caso del riesgo "2", se debe tener muy claro la fase de desarrollo y análisis por tal de no fallar en las pruebas y test. Ya que no se puede provocar retrasos en el proyecto.

Por último, el riesgo "3", hay que tener muy claro el correcto funcionamiento de las tareas antes de presentarlas al jefe de proyecto, por tal de que dé el visto bueno de forma directa.

3.8 Presupuesto

3.8.1 Estimación del coste de personal

Tabla 3: Coste de personal

| RECURSOS | PRECIO/H | HORAS TOTALES | COSTE TOTAL |
|-------------|----------|---------------|-------------|
| Analista | 50€ | 12 | 600€ |
| Programador | 30€ | 50 | 630€ |
| Técnico | 20€ | 36 | 480€ |
| Tutora | * | 9 | * |

*En este caso particular, las horas de la tutora no están contabilizadas al tratarse de un proyecto final de carrera, y no son relevantes para el coste total del proyecto.

Coste de los recursos de personal : $600 + 630 + 480 = 1710$ €

3.8.2 Estimación del coste de los recursos.

Los recursos físicos que necesitamos son los siguientes:

- Kit Inventor Arduino: 91€
- Módulo Bluetooth iTeed: 25€
- Teléfono Android: 25€
- Programa Office 2010 para Mac: 114€
- Altavoz con adhesivo : 1,5€
- Transformador : 15€

El coste total de los recursos es: $91 + 25 + 25 + 114 + 1,5 + 15 = 271,50$ €

El coste total del proyecto es : $1710 + 271,50 = 1981,50$ €

3.9 Conclusiones del plan de proyecto

Por tanto sabemos que a corto plazo la comparación de nuestro proyecto con la competencia es inferior, pero a la larga nos genera unos beneficios por los costes que se pueden amortizar. Ya que, por ejemplo, una contratación de una alarma para la seguridad del hogar con la compañía Prosegur, hoy en día tiene un coste de 30€ mensuales con un período de permanencia de dos años, sin incluir la tarifa de alta (90€) ni el IVA. Nos sale un precio de 820€.

Por tanto en tres años tenemos amortizado todo el coste del proyecto. Está claro que sería mucho antes de lo indicado, ya que estamos tomando en cuenta la comercialización de un solo dispositivo, después se reducirían los costes de una manera drástica. Por tanto se ve futuro en la comercialización del proyecto.

Capítulo 4: Fundamentos Teóricos

En este capítulo se quiere introducir al lector a todos los conocimientos necesarios para llevar a cabo el proyecto. Cuales son los fundamentos básicos para seguir adelante. En este caso, se tocan cuatro temas esenciales que son los pilares maestros para poder desarrollar los objetivos marcados.

4.1. Comunicación inalámbrica :

La comunicación inalámbrica o sin cables es aquella en la que extremos de la comunicación, en este caso el emisor y el receptor, no se encuentran unidos por un medio de propagación físico, sino que se utilizan la modulación de las ondas electromagnéticas a través del espacio. Es por eso, que sólo los emisores y receptores son los dispositivos físicos, entre los cuales no encontramos : antenas, ordenadores portátiles, teléfonos móviles, etc.

En general, la tecnología inalámbrica utiliza las ondas de radiofrecuencia de baja potencia y una banda específica, de uso libre o privada, para transmitir entre dispositivos.

En 1887 Heinrich Rudolph Hertz, un físico alemán, demostró que existían las ondas electromagnéticas y que éstas podrían ser usadas para mover información a grandes distancias, esto, le valió que la unidad con las que son medidas las frecuencias del espectro lleven su apellido (Hertz o Hz).

La base teórica de las ondas electromagnéticas fueron desarrolladas mucho antes por el físico escocés James Clerk Maxwell en 1864. El primer uso de las ondas electromagnéticas fue la telegrafía inalámbrica. Este relevante acontecimiento sería el predecesor de la propagación electromagnética o transmisión de radio.

Utilizando estos conceptos, el italiano Guglielmo Marconi inventa la radio en 1901. La radio fue el primer medio masivo de comunicación inalámbrica y a poco más de 100 años de su invención, las comunicaciones móviles han demostrado ser una alternativa a las redes cableadas para ofrecer nuevos servicios que requieren gran ancho de banda, pero con otros beneficios como la movilidad y el estar comunicado en cualquier lugar, en cualquier momento.

4.2 Ondas electromagnéticas :

Una onda electromagnética consiste en una perturbación que se propaga a través del espacio y a lo largo del tiempo. Como por ejemplo las ondas de sonido que son producidas por la variación de presión que se producen en las moléculas de aire, a este tipo de ondas también se les conoce como ondas mecánicas, ya que necesitan un medio material para ser transportadas, no se propagan por el vacío.

En cambio una onda electromagnética, tiene la particularidad que también puede transmitirse por el vacío, están formadas por una composición de campos eléctricos y campos magnéticos perpendiculares entre sí.

Los campos electromagnéticos son producidos mediante cargas eléctricas en movimiento, por lo que para generar ondas electromagnéticas se necesitan circuitos oscilantes (ilustración 3).

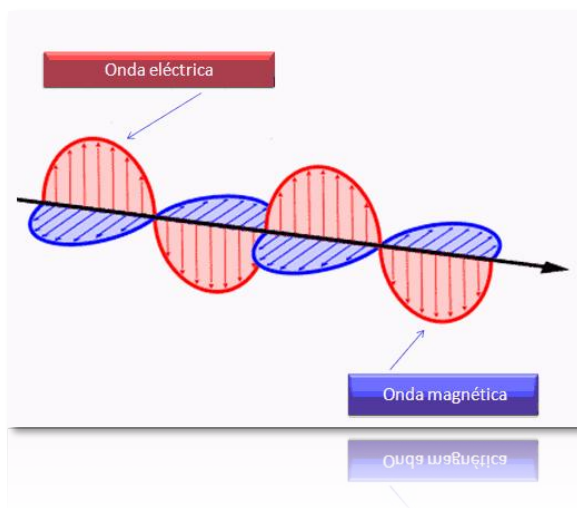


Ilustración 2: aspecto onda electromagnética

El espectro electromagnético (ilustración 4). es muy amplio, también conocido como espectro, va desde unos pocos Hertzios, hasta ondas de frecuencia superiores a 10^{23} Hz.

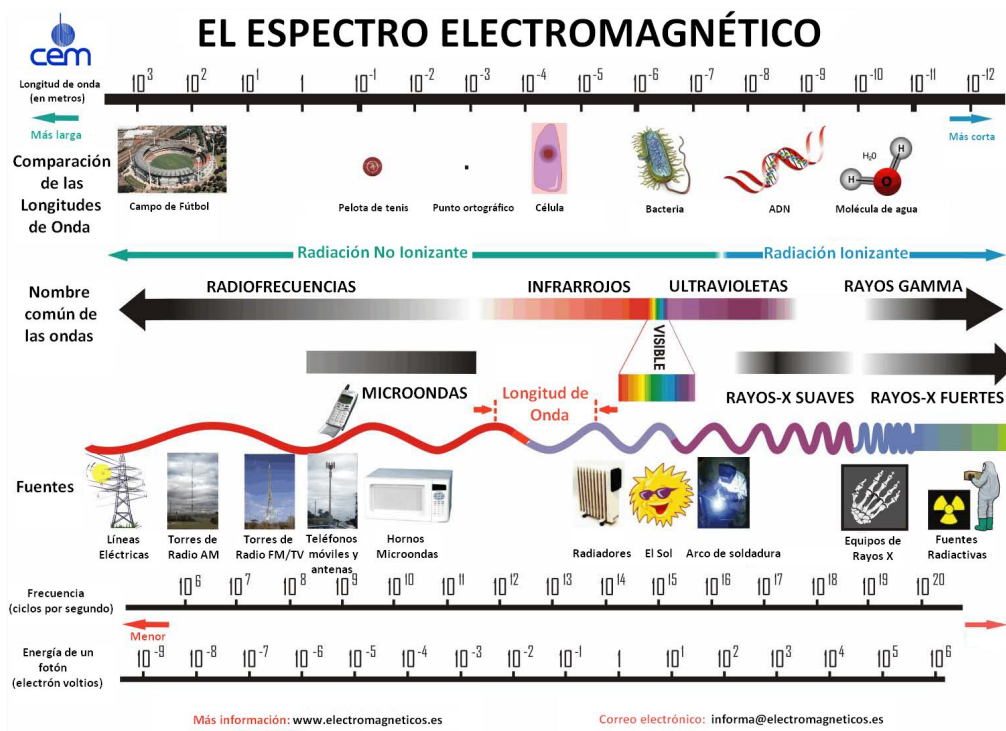


Ilustración 3: espectro electromagnético

Entre los aspectos relacionados con las comunicaciones, presenta un interés especial el rango de frecuencias por debajo de los 10^{10} Hz. Que incluyen las llamadas ondas de radio o las microondas.

4.2.1 Modulación de onda:

Para transmitir una señal mediante ondas de radio, es necesario adaptar dicha señal para que pueda ser transmitida. Este proceso, recibe el nombre de modulación.

La modulación consiste en combinar dos tipos de ondas, una onda moduladora y otra onda portadora.

Onda moduladora: es de baja frecuencia y contiene la información a transmitir.

Onda portadora: es de alta frecuencia, adecuada para la transmisión. Suele ser conocida también como señal de radiofrecuencia

La onda portadora no contiene la información, pero será la encargada de actuar como media de soporte de dicha información. La forma de la onda portadora, es alterada en el proceso de modulación de una manera determinada. Como resultado de la superposición de las dos ondas, se obtiene una nueva onda llamada onda modulada, ésta contiene la información y presenta la frecuencia adecuada para que pueda ser transmitida y recibida.

Existen dos tipos de modulación: amplitud modulada, y frecuencia modulada.

4.2.2 Amplitud modulada y frecuencia modulada :

Amplitud modulada:

La señal portadora es modificada, de forma que su amplitud varía en función de la señal moduladora.

Frecuencia modulada:

La frecuencia modulada intenta subsanar algunas características de la amplitud modulada, por ejemplo el ancho de banda útil para la transmisión de la AM resulta muy restringido y además es una señal muy sensible y poco fiable en cuanto a las interferencias que se puedan sumar a la señal durante su trayecto.

Mediante la frecuencia modulada se consigue transmitir mayores ancho de banda y señales menos sensibles a las interferencias. La señal obtenida tiene un valor de amplitud constante igual al de la onda portadora, siendo variable la frecuencia según el valor de amplitud que tome la onda moduladora. Dado que la información que transmite se ve poco alterada por las interferencias, la FM se utiliza para la transmisión de voz y música.

4.2.3 Comunicaciones terrestres :

Las ondas electromagnéticas se propagan en línea recta. En consecuencia, si queremos enviar una onda de radio de larga distancia, dado que la tierra tiene una superficie redondeada, la señal se alejará y se perderá en el espacio.

Sin embargo, las ondas de radio tienen la propiedad de reflejarse en las capas altas de la atmósfera, en concreto en la ionosfera. Las ondas de radio se van atenuando, y haciendo cada vez más débiles. Por ello, es necesario de disponer de estaciones repetidoras terrestres que amplifiquen y remitan la señal.

Un sistema de comunicaciones inalámbricas terrestres en general está constituido por los siguientes elementos: emisor de radiofrecuencia, antena emisora, estaciones repetidoras, antena receptora, receptor de radiofrecuencia.

La antena emisora, emitirá señales de una sola frecuencia o de un ancho de banda muy reducido, y a una potencia suficientemente grande para que la señal sea captada por el destino deseado. Esto implica que su estructura debe ser lo suficientemente grande como para permitir grandes corrientes o tensiones eléctricas.

La antena receptora, recibe las señales de muchas frecuencias, por lo que su ancho de banda de recepción debe ser mucho más amplio, además estas señales suelen ser normalmente mucho más débiles.

El receptor de radiofrecuencia, es el paso final para las ondas electromagnéticas que son recogidas por la antena, se envían al receptor que se demodula, y se reconstruye la señal inicial.

4.3 Arduino :

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquier interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El micro controlador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing).

Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP). Las placas pueden ser hechas a mano o compradas montadas de fábrica; el software puede ser descargado de forma gratuita.

Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta, así pues se es libre de adaptarlos a cualquier necesidad.

Existen multitud de placas Arduino distintas, algunas ya cuentan con adaptadores integrados o componentes extra como sensores o distintos tipos de conectores. También existe la posibilidad de comprar escudos o shields, para adaptarlos a tu placa Arduino y así hacerla más funcional, como es el ejemplo de este proyecto. El cual se ha tenido que integrar un componente Bluetooth para poder establecer la comunicación con dispositivos móviles. En nuestro caso particular, disponemos de la placa Arduino UNO, la cual pasamos a explicar sus entradas y componentes para tener un mayor conocimiento de la misma.

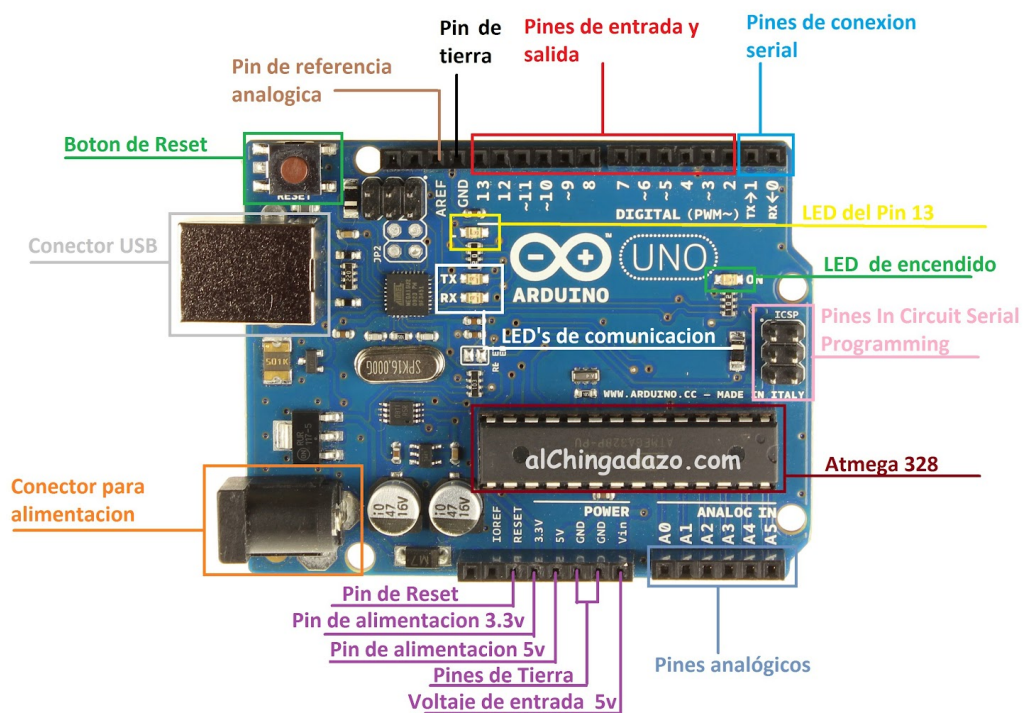


Ilustración 4: esquema placa Arduino Uno

Como podemos apreciar, la placa Arduino UNO, cuenta con numerosos pines de entrada o de salida para poder realizar multitud de conexiones, vamos a explicar para que sirve cada uno de ellos.

- El controlador del Arduino UNO es un ATmega328, tiene 14 pines digitales de entradas y salidas, de las cuales 6 pueden usarse como salidas PWM. Se encuentra en el centro de la placa.
- Los primeros dos pines (0 y 1) están reservados para comunicación serial externa así que no se pueden usar como salidas o entradas por el momento. Están marcados con las letras tx y rx.
- Tenemos también un botón de reset, con este podremos reiniciar el Arduino y que realice de nuevo la rutina de programación. Se encuentra en una de las esquinas de la placa.
- Como verás la placa tiene varios LEDs , uno de encendido, dos de comunicación serial (Tx y Rx), y un LED indicador. Este último es muy importante, ya que podemos visualizar algún programa simple en el, como encenderlo o hacer que parpadee. Para poder controlar este LED se hace con el pin 13, el cual lleva una resistencia por lo que no es necesario añadir otra más.

- Los pines ICSP se refiere a Programación Serial En Circuito, esto nos permite cargar un programa a algún otro Arduino u otro micro controlador de la serie AVR pero que cuente con el bootloader.
- El plug de alimentación sirve para fuentes externas y cuando quieras dejar tu Arduino en un proyecto que hayas hecho, puedes conectar alguna batería de preferencia de 9v.
- En la parte inferior tiene 6 entradas analógicas (A0 - A5), éstas son muy importantes a la hora de usar sensores y nos permite ver el voltaje que tenga el sensor en una escala de 0 a 1024. Además estos pines pueden ser configurados como salidas no es muy recomendable pero cuando es muy necesario se pueden usar, los pines analógicos se corresponden con los números 14 al 19.

Por lo que respecta a las especificaciones técnicas tenemos lo siguiente:

| | |
|--------------------------------------|------------|
| Microcontrolador | ATmega328 |
| Voltaje de operación | 5V |
| Voltaje de entrada recomendado | 7-12V |
| Corriente máxima por pines de salida | 40 mA |
| Corriente máxima para pin de 3.3v | 50 mA |
| Velocidad del reloj | 16 MHz |
| Tamaño de EEPROM | 1024 bytes |

Para alimentar la placa Arduino, existen una serie de entradas para dotar de corriente a todo el circuito 3,3V, 5V y GND.

La entrada de 3,3V entrega un voltaje de 3,3V y soporta 50mA como carga máxima. En el pin de los 5V salen los 5V del regulador de la placa, ya sea que esté conectado por medio del USB o una fuente externa.

Por último tenemos el GND, en dicha placa contamos con 3 pines de tierra para mayor comodidad.

En la actualidad nos podemos encontrar muchas páginas web por internet, donde poder comprar cualquiera de los componentes, así como pequeños ejemplos o proyectos que sirven de ayuda. A continuación se enumeran alguna de las páginas donde poder aprender a programar esta tecnología y también donde poder hacerse con uno de estos dispositivos.

www.arduino.cc
www.bricogeek.com

www.electan.com

www.ro-botica.com/ARDUINO

www.aprenderrobotica.com

También, en la propia web de Arduino nos indican las tiendas donde poder comprar según la localidad geográfica, ejemplos de introducción y “sketches”, que son los pequeños programas que se pueden ir creando. La ventaja de Arduino, es que el único límite que existe es el del programador, ya que se pueden llegar a crear proyectos muy diversos y a la vez útiles. Una buena prueba de ello son los links de los videos de estos proyectos.

<https://www.youtube.com/watch?v=T-lizo1RCL8&list=PL6DD4FCA2DE3BC1E8>

(Recopilatorio de videos de proyectos utilizando Arduino)

<https://www.youtube.com/watch?v=pcisMwOXMkA>

(Ordenador de a bordo para vehículos)

<https://www.youtube.com/watch?v=wuFOO40paJg>

(Girasol guiado por luz a través de sensores)

<https://www.youtube.com/watch?v=f3gFuku60zY>

(Casa domótica)

<https://www.youtube.com/watch?v=zMEZfkO9GMY>

(Cuadróptero guiado por Arduino)

4.3.1 “Hola mundo” Arduino :

Como en cualquier lenguaje de programación igual que en el deporte, no se puede correr sin antes saber andar, es decir, no se puede empezar a programar nada sin antes haber compilado un “Hola Mundo”, que no es nada más que el compilar las líneas de código más sencillas y simples para poder ejecutar una aplicación, normalmente por primera vez.

En el caso de Arduino, el hola mundo es el sketch “Blink” (parpadear en inglés). Es un ejemplo en el cual se le va dando corriente a un led conectado a la placa Arduino de forma intermitente haciendo que éste parpadee, para poder ver el funcionamiento básico de la programación. El código es el siguiente :



```

Blink | Arduino 1.0.3
Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
  
```

Ilustración 5: Sketch Blink en Arduino

Sobretodo, hay que tener en cuenta un par de claves que son la tónica para cualquier aplicación o sketch de Arduino. Como son la declaración de variables, el void setup(), y el void loop().

La declaración de variables las realizaremos en las primeras líneas de código, así como el valor de las mismas.

La función void setup() es la primera que se ejecuta cuando abrimos nuestra aplicación, y es en ella donde tenemos que indicar los pines que se van a utilizar de la placa, y si serán de entrada o de salida.

Para acabar, nos encontramos con la función void loop(), la cual es una función cíclica que se va repitiendo una y otra vez. En este caso, le indicaremos que ponga el pin

número 13 con valor a 1 durante un segundo, y que cambie a 0 durante el mismo instante de tiempo, así conseguimos un efecto de intermitencia.

Una vez tenemos nuestro código podemos asegurarnos antes de cargarlo en la placa que el código es correcto. Para ello clicamos el primer botón de la parte superior de nuestra aplicación Arduino, identificada con un símbolo de check, si el resultado es positivo, podremos cargar la aplicación en nuestra placa con el botón inmediatamente contiguo al primero, el cual nos traspasa y guarda el sketch en la memoria de la placa.

El botón con el símbolo de folio doblado es para crear un documento nuevo. El botón con la flecha apuntando hacia arriba es para abrir un sketch y el de la flecha hacia abajo es para guardarlo.

El botón que aparece a la derecha de la ventana, en forma de lupa, es para abrir la comunicación en serie con nuestro Arduino, lo cual significa que si lo tenemos conectado vía USB, podremos enviar y recibir información de forma recíproca.

4.4 Qué es Android :

Android es un sistema operativo basado en Linux, diseñado principalmente para móviles con pantalla táctil como teléfonos inteligentes o tabletas inicialmente desarrollados por Android, Inc., que Google respaldó económicamente y más tarde compró en 2005, Android fue presentado en 2007 junto la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles.

Android tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. Actualmente , se han sobrepasado las 700.000 aplicaciones disponibles para la tienda de aplicaciones oficial de Android: Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica, un framework OpenCore, una base de datos relacional SQLite, una interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

4.4.1 Desarrollar Android :

Las aplicaciones se desarrollan en lenguaje Java, con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C o C++, Google App Inventor, un entorno visual para programadores novatos.

El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software provisto por Google el cual se puede descargar gratuitamente.

Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos.

A continuación se añade una imagen sobre el esquema de desarrollo de aplicaciones Android a través de App Inventor.



Ilustración 6: aplicación App Inventor

4.4.2 Hola mundo Android :

Por defecto, cuando creamos un nuevo proyecto en Android, nos aparecen una serie de carpetas con información que nosotros no necesitamos o eso pensamos. Cualquier aplicación Android tiene una estructura, y se deben seguir una serie de pasos para el correcto funcionamiento de la misma.

Para empezar, explicaremos las carpetas más importantes que se crean a la hora de crear un proyecto nuevo para Android.

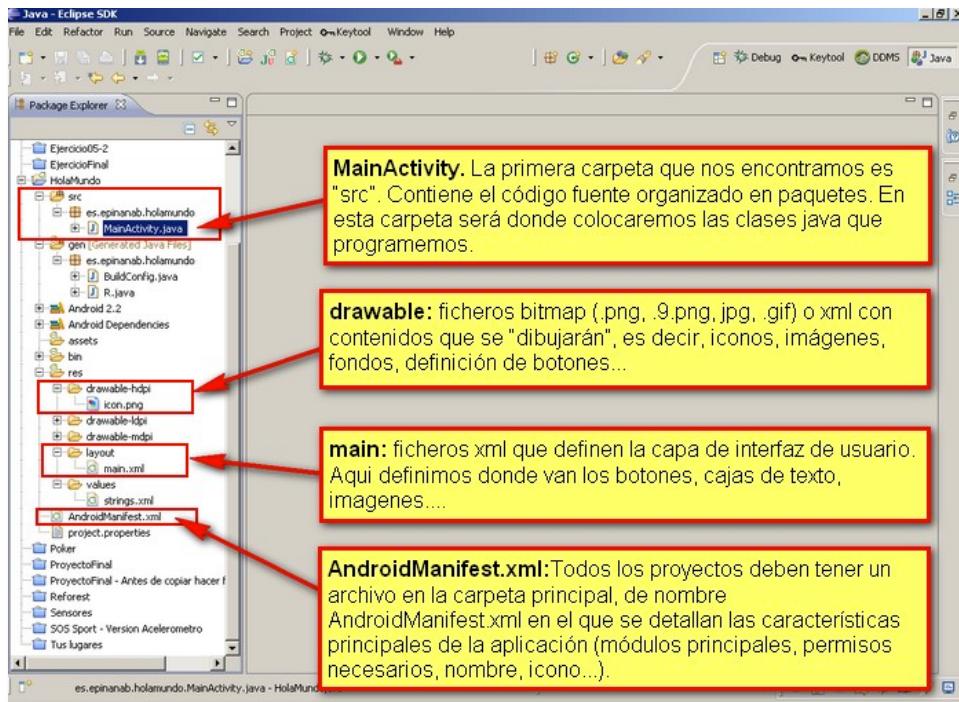


Ilustración 7: árbol de carpetas Eclipse

Cuando creamos una aplicación, lo primero que se hace es leer el código que aparece en la clase principal (MainActivity), porque así se especifica en el archivo AndroidManifest.

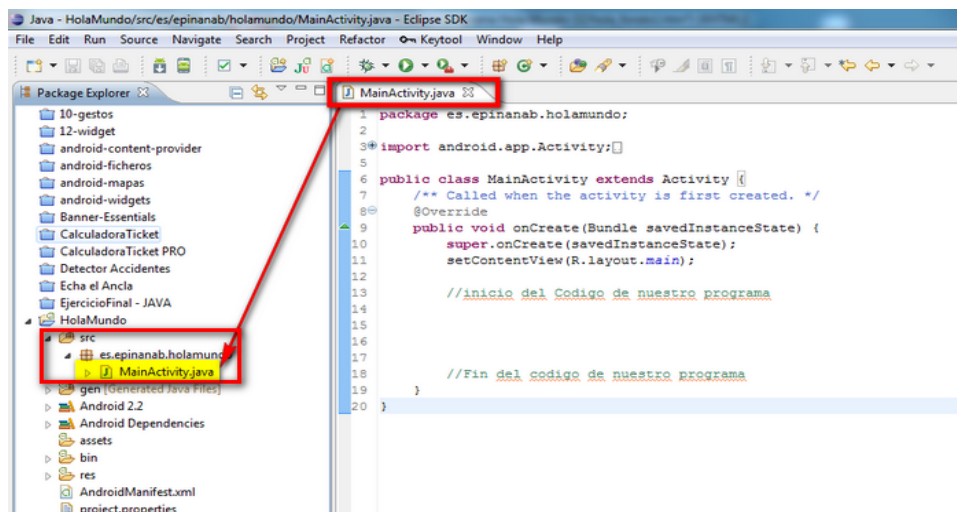


Ilustración 8: clase MainActivity de Eclipse

Las actividades son las encargadas de mostrar las interfaces de usuario e interactuar con él. En ella se encuentran distintos métodos, como por ejemplo el método OnCreate que es lo primero que se va ejecutar cuando iniciemos la activity.

Por último, debemos comunicarle a la actividad la pantalla que vamos a asociar para esa actividad, y eso lo hacemos con `setContentView(R.layout.main)`. Es importante decir que por cada pantalla habrá una actividad distinta.

En la carpeta `drawable` (`res`) se guarda el icono de nuestra aplicación, tiene que tener una resolución concreta, para el caso de los iconos para las aplicaciones es de 72x72 píxeles.

En la carpeta `Values` (`res`) se guardan todas las cadenas de texto que usaremos en nuestra aplicación para los botones, cuadros de texto, mensajes a los usuarios, etc. La estructura de ficheros es XML.

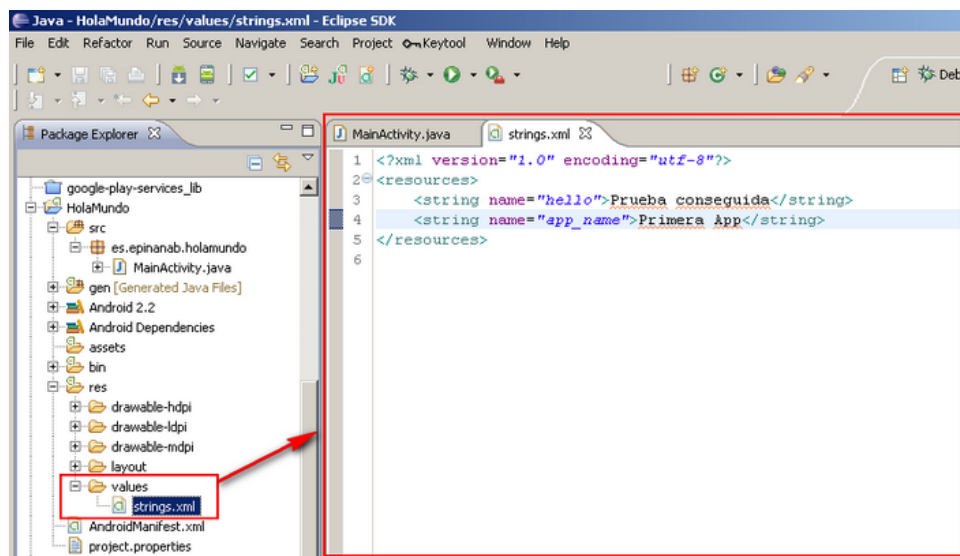


Ilustración 9: carpeta Values de Eclipse

En la carpeta de `Layout` (`res`), tenemos una de las características más importantes de la aplicación, el diseño y la estructura de la actividad.

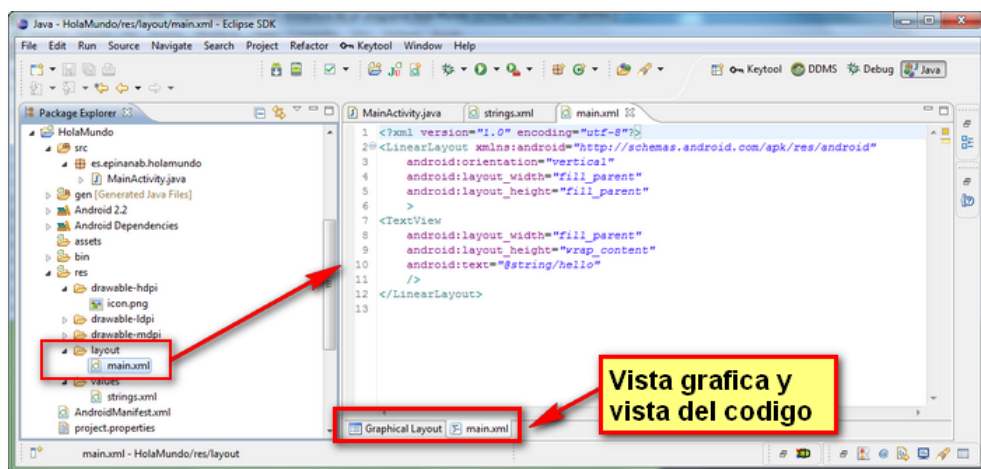


Ilustración 10: carpeta layout Eclipse

La vista gráfica nos permite comprobar como queda nuestro diseño, es muy útil para visualizar nuestros elementos.

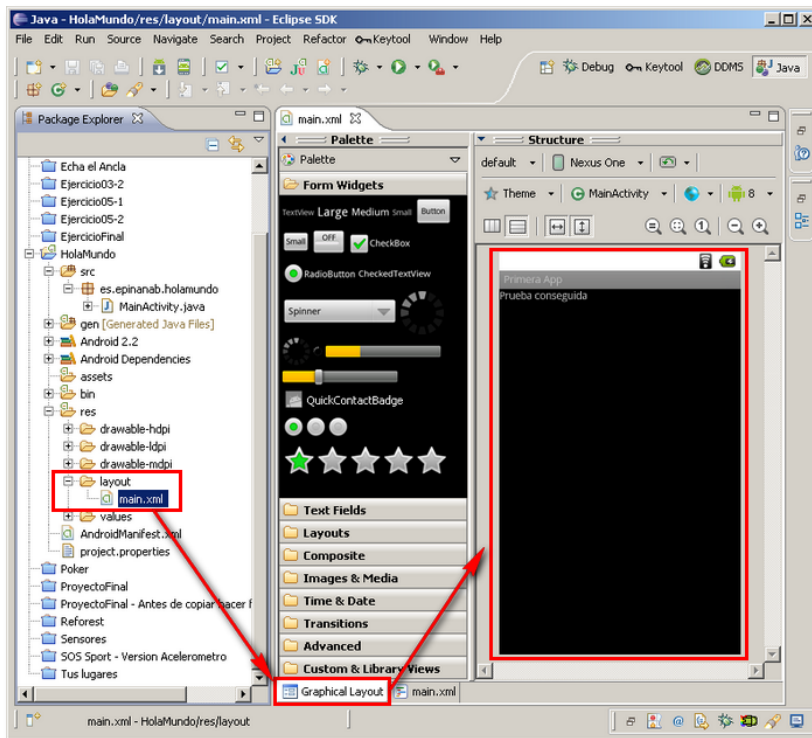


Ilustración 11: main.xml de la carpeta layout Eclipse

Por último, decir que todos los proyectos deben tener un archivo en la carpeta principal llamado AndroidManifest.xml, en el que se detallan las características principales de la aplicación, cada activity se define aquí, los widgets, módulos principales, permisos necesarios, nombre de la App, etc.

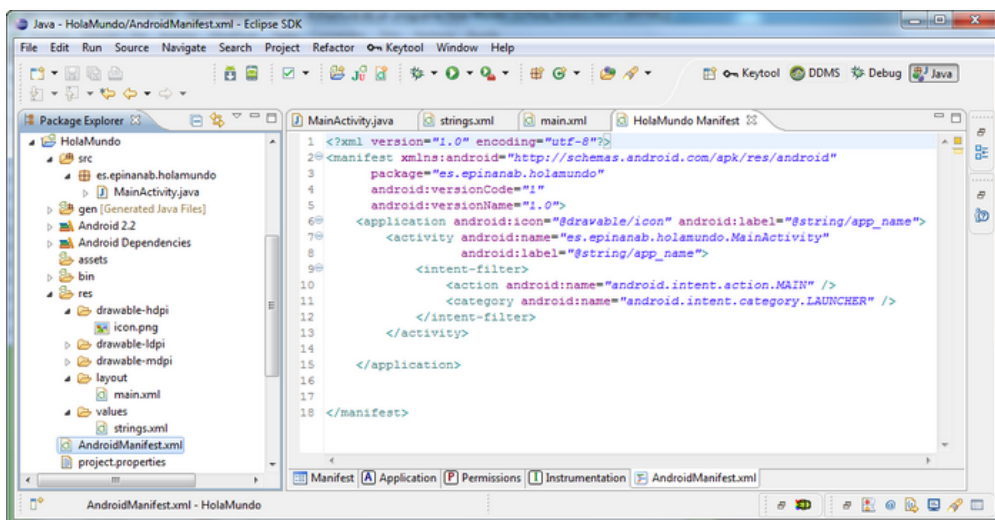


Ilustración 12: AndroidManifest.xml de Eclipse

Finalmente, una vez hemos descrito todas las líneas de código necesarias, debemos exportar nuestro proyecto para crear el archivo ejecutable, en este caso con extensión “.apk”, este archivo lo pasaremos mediante USB a nuestro dispositivo móvil y lo instalaremos para probarlo de forma física.

Hay otra posibilidad de probar el funcionamiento de los programas sin necesidad de instalar el archivo en el teléfono o Tablet, y es utilizando el emulador que podemos instalar en Eclipse, el único inconveniente es que no se pueden probar una serie de funciones como el osciloscopio o las funciones Bluetooth. Sólo debemos clicar el botón de “run” en nuestro Eclipse, y automáticamente se instalará la aplicación en el emulador seleccionado.

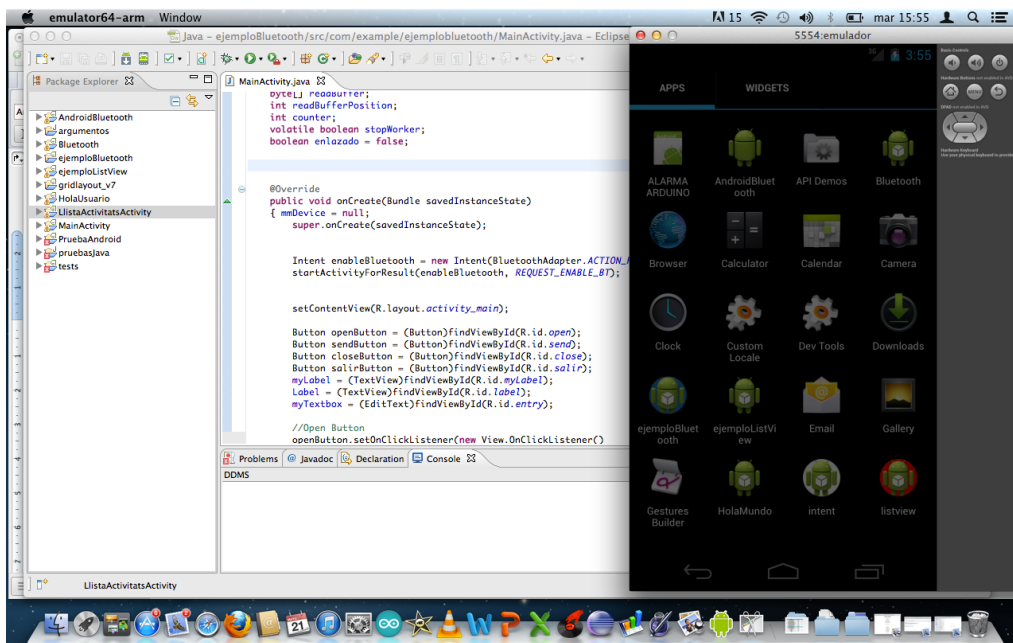


Ilustración 13: emulador Android en Eclipse

Capítulo 5: Alarma Arduino controlada inalámbricamente por dispositivos Android

En este apartado se explican los pasos necesarios para llegar a diseñar y a fabricar el proyecto en cuestión. Dividiéndolo en dos partes, la primera Hardware y la segunda Software. Se va a relatar paso a paso las fases que se han seguido, los problemas ocasionados según se iban desarrollando las fases, como se han solucionado, etc.

Para comenzar se describe la parte Hardware (prototipo Arduino), ya que éste no necesitaba de una aplicación propia para probarse.

5.1 Parte mecánica (HARDWARE)

En primer lugar, debíamos conseguir los elementos necesarios para poder empezar a montar pequeños circuitos, con el fin de llegar a montar algo parecido a una alarma que se vinculara con nuestra aplicación futura. Necesitábamos una serie de componentes que hasta el momento no teníamos, como una placa protoboard, LEDs, resistencias, y lo que es más importante la placa Arduino.

Después de una larga búsqueda por tiendas especializadas de la zona sin obtener un resultado positivo, pasamos a mirar por las tiendas online de internet para hacernos con alguno de estos dispositivos de la forma más económica y más rápida posible.

Finalmente, encontramos que se comercializan una serie de kits de Arduino que incluyen muchos de los componentes con los cuales podemos llegar a diseñar algo parecido a las ideas que tenemos para nuestro proyecto.

Así que decidimos comprar el KIT INVENTOR ARDUINO en la web de bricogeek.

Es un kit para personas que se quieren introducir en este mundo de electrónica de la mano de Arduino, el coste no nos parece caro para la cantidad de componentes que incluye, y pagamos 79,90€ más los gastos de envío.

Este es link del kit en la misma página de BricoGeek :

<http://www.bricogeek.com/shop/arduino/380-arduino-inventor-kit.html>

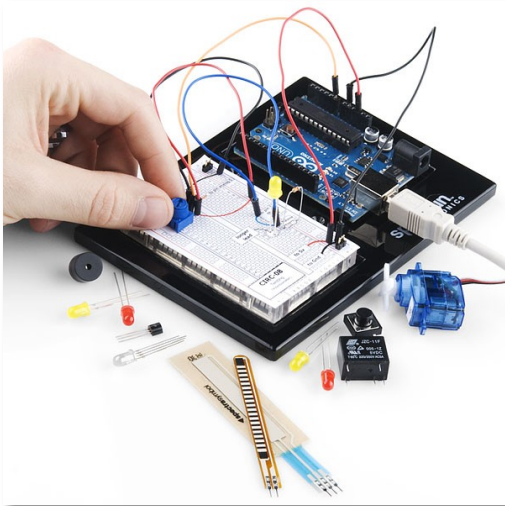


Ilustración 14: kit inventor Arduino

El Kit incluye:

- Arduino UNO USB (r3)
- Mini protoboard y soporte
- Manual impreso satinado a color (36 páginas, en inglés)
- 12 Pegatinas de circuitos (overlays)
- 74HC595
- Transistor 2N2222
- Diodo 1N4148
- Motor DC Motor con cables
- Pequeño servomotor
- Relé 5V
- Sensor de temperatura TMP36
- Sensor Flex
- Potenciómetro lineal
- Potenciómetro táctil
- Cable USB para programar
- Set de cables
- Fococélula LDR
- Diodo LED RGB
- Diodos LED rojos y verdes
- Potenciómetro 10K
- Zumbador Piezo
- Grandes pulsadores de 12mm
- Resistencias de 330 y 10K
- Pines macho

Este kit incluye una serie de ejemplos muy sencillos de realizar, para probar los diferentes componentes del kit, así como para coger las bases de programación de Arduino, las cuales son muy parecidas al lenguaje en C.

Al principio muchos de los ejercicios que se proponían en el kits de iniciación resultaron bastante frustrantes por el poco conocimiento en cuanto a conexiones, etc. Así que resultó de gran utilidad una aplicación llamada Fritzing.

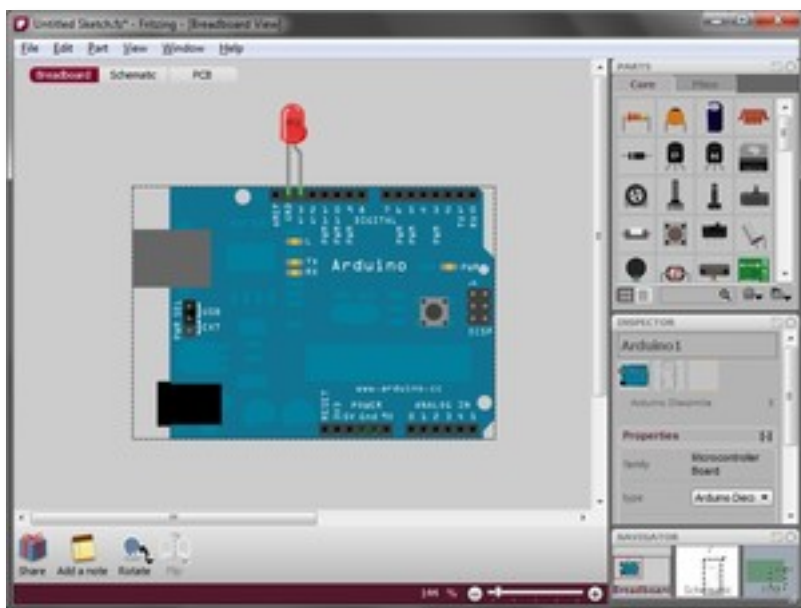


Ilustración 15: aplicación Fritzing

Es una aplicación gratuita y multiplataforma que intenta simular cualquiera de las placas Arduino disponibles en el mercado y los componentes que se le pueden acoplar a ellas. Además es una aplicación open-source o de código libre y gratuita.

Este es link de la página oficial : <http://fritzing.org/>

Esta aplicación, nos permitió hacer muchísimas pruebas de una manera virtual antes de pasar a la acción con la placa física, y así no perder tiempo innecesario en pruebas que no nos llevarían a ningún lado.

Una vez teníamos claro los pasos que deberíamos seguir y las fases que debería tener la parte mecánica, procedemos a montarla.

Inicialmente, el prototipo consta de tres LEDs con sus correspondientes resistencias que se activaran dependiendo del carácter que le pasemos vía serie, por el USB de nuestro Mac. De la misma manera, si le pasamos el carácter adecuado a nuestra placa, hará que nuestro buzzer reciba la corriente necesaria para emitir un sonido como si de una alarma se tratara, dentro de las características sonoras que nos puede proporcionar el buzzer del KIT de Arduino.

Una de las mejoras que se hicieron en el prototipo, fue la de poder enviar una cadena de caracteres para realizar la activación y desactivación del Buzzer, y no sólo el envío

de un solo carácter. Con esto conseguimos tener más seguridad a la hora de que el usuario tenga que interactuar con la alarma.

Otra de las mejoras que se modificaron, fue la de sustituir el Buzzer de serie que incorporaba el Kit Inventor, por un interruptor de lengüeta conectado a un altavoz preparado para utilizarse como avisador acústico en caso de apertura de puerta o de alarma. El precio es muy económico y le da al prototipo un enfoque más orientado a la seguridad, y se asemeja más a una alarma.



Ilustración 16: altavoz con interruptor de lengüeta

El funcionamiento es sencillo, en el interior del dispositivo hay una pila que da voltaje al altavoz. Si se separa una de las dos partes del imán, se activa provocando un efecto en el interruptor de lengüeta, cerrando el circuito, y a su vez haciendo que suene el altavoz. Obteniendo un sonido muy superior y más estridente que el del Buzzer.

A partir de este momento, ya podemos activar y desactivar la alarma, dándole voltaje al pin digital de la placa Arduino, el cual está conectado al dispositivo descrito anteriormente.

En esta fase del proyecto, la comunicación de información se hacía mediante un cable USB entre la placa Arduino y el monitor serie de Arduino del ordenador. El siguiente paso, era el de conseguir un módulo Bluetooth que permitiera la comunicación sin cables. Para ello, necesitábamos conseguir un módulo externo Bluetooth y adaptarlo a la placa, ya que la placa que poseemos (Arduino UNO) no dispone de ningún tipo de conexión inalámbrica.

Así pues, nos hacemos con una Shield (una armadura) para nuestro Arduino UNO que incluye un módulo Bluetooth, la shield nos permite acoplar encima de la placa cualquier módulo, encajándola en los pines superiores.

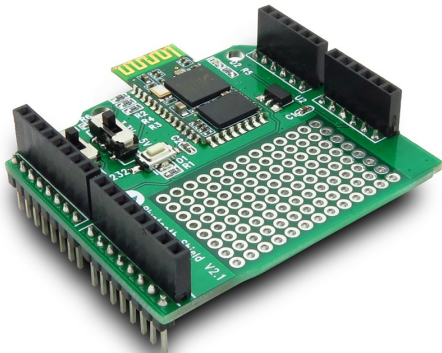


Ilustración 17: shield Bluetooth para Arduino

En la página oficial del dispositivo, nos indican como se debe conectar, la api para programarlo, características, etc.

http://www.seeedstudio.com/wiki/Bluetooth_Shield

Realmente, no se debe modificar ninguna línea de código para conectar el módulo Bluetooth de iTeed, debido a que la comunicación, aunque inalámbrica, sigue siendo comunicación en serie, y el dispositivo está a la escucha de recibir información constantemente. Sólo deberíamos utilizar la Api del dispositivo, si quisiéramos modificar algún parámetro de serie, como el nombre del dispositivo, el código de enlace Bluetooth, etc.

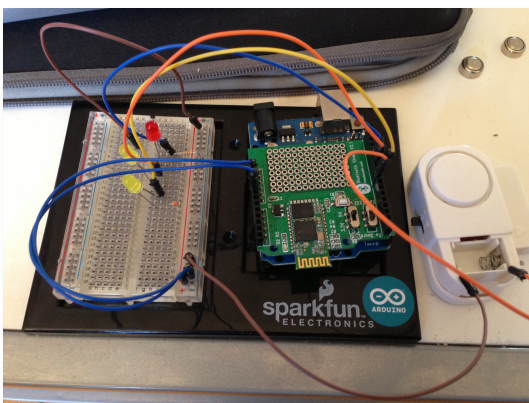


Ilustración 18: proyecto en ejecución

Para probar el dispositivo, debemos comprobar que el led luminoso del accesorio Bluetooth, se encuentra parpadeando en verde de forma intermitente en cuanto recibe corriente. Para comprobar si está emparejado, el led luminoso se debería cambiar de estado, a fijo, en vez de intermitente.

Para ello debemos enlazar el dispositivo Bluetooth con el dispositivo móvil, e introducir el código de emparejamiento, en este caso, al no haberlo modificado vía software, sigue siendo el código que nos provee el fabricante, en este caso “1234”.

Como actualmente sólo nos estamos dedicando a la parte Hardware, es decir, no tenemos ninguna aplicación propia creada en Android, debemos descargar alguna aplicación de Google Play a través de nuestro dispositivo inalámbrico, en nuestro caso un teléfono HTC Desire HD con sistema operativo Android 2.3 (Froyo).

La aplicación que elegimos es S2 Bluetooth, una App gratuita y muy sencilla que nos va a permitir realizar nuestras pruebas de conexión con el dispositivo, así como de intercambio de información entre el emisor y el receptor.

Una vez abrimos la aplicación, directamente se nos abre un warning, avisándonos que la aplicación utiliza la comunicación Bluetooth y que debemos activar dicha función de nuestro dispositivo.

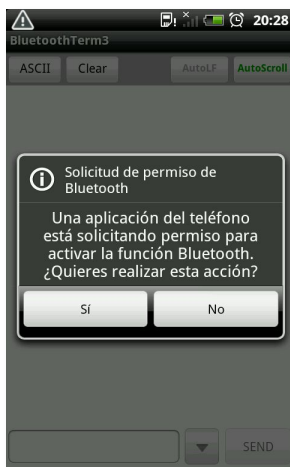


Ilustración 19: aplicación solicitando permiso Bluetooth

Una vez que apretamos al botón de “sí”, éste es el estado de nuestra aplicación.



Ilustración 20: aplicación S2 Bluetooth

Como nos indica en la parte superior, aún no estamos conectados a ningún dispositivo, por tanto, es lo primero que deberíamos de hacer. Para ello, apretamos el botón de menú de nuestro dispositivo, y nos aparece el siguiente desplegable.

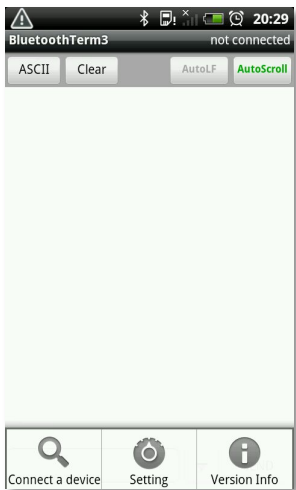


Ilustración 21: menú aplicación S2 Bluetooth

Para conectarnos a los dispositivos Bluetooth, clicamos a “Connect a device” y nos aparecen los dispositivos que reconoce la placa Arduino a su alcance.



Ilustración 22: S2 Bluetooth conectando a un dispositivo

Escogemos nuestro dispositivo de la lista que nos aparece, y una vez introducimos el código de emparejamiento, proporcionado por el fabricante, ya tenemos enlazados los dos dispositivos tal como queríamos.



Ilustración 23: S2 Bluetooth conectando a iTeard



Ilustración 24: S2 Bluetooth conectado a iTeard

La primera imagen, nos muestra como el dispositivo está intentando conectar con el Itead (Bluetooth de Arduino), y así nos lo indica en la parte superior de la App con el mensaje de texto “connecting”. Una vez se ha conectado, el propio Arduino nos devuelve un mensaje indicando si la alarma está conectada o no.

Si queremos conectar o desconectar la alarma, los pasos que debemos seguir son los siguientes, primero escribimos la cadena que activa o desactiva la alarma en el text edit de la parte inferior del dispositivo, y a continuación le damos al botón “send”, el cual nos envía la información escrita. Si la cadena enviada es correcta, activará o desactivará nuestra alarma.

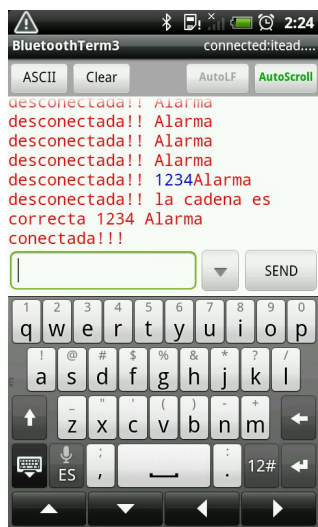


Ilustración 25: mostrando mensaje de alarma conectada

La validación de la parte Hardware ha sido todo un éxito. Vemos que funciona correctamente la parte mecánica, y la parte inalámbrica. Por lo tanto lo que nos queda ahora es crear nuestra propia aplicación que se comuniquen con la placa Arduino, con el objetivo de poder hacer exclusivo y más funcional la App para futuros clientes, así como personalizarla, según las necesidades del cliente.

5.2 Parte no mecánica (SOFTWARE)

Este proyecto tienes dos partes relacionadas con el software, una es la parte de la programación Android, la cual hablaremos más adelante. Y otra, es la parte que se encarga de programar el funcionamiento de la placa Arduino según nuestros intereses.

Vamos a detenernos en esta parte, para entender como funciona nuestro pequeño sketch realizado en Arduino.

En el apartado anterior, relacionado con el primer programa Arduino, también llamado “Hola Mundo” por los desarrolladores, explicábamos que hay dos partes importantes en todos los programas implementados para Arduino, por un lado tenemos la función loop, la cual es cíclica y se repite una y otra vez. Y la parte de la función setup, la cual se ejecuta únicamente al inicio de poner en marcha el sketch.

Sobre estas dos partes se ha desarrollado nuestra aplicación Arduino que pasamos a comentar:

Primeramente se ha realizado la declaración de variables y su inicialización, tanto de pines digitales como de variables booleanas o de la propia cadena de activación.

```

int ledPin9 = 9;
int ledPin11 = 11;
int ledPin10 = 10;
int speakerPin = 8;
int x = 0;
int y = 5;
int errores = 0;
char vstart[] = "1234";
boolean conectada = false;
boolean activando = false;

```

Seguidamente, nos encontramos con la función `setup`, en la cual debemos de indicar los pines que van a ser de entrada o de salida así como el valor del Baud Rate de la comunicación en serie.

```

void setup()
{
  Serial.begin(9600);
  pinMode(ledPin9,OUTPUT);
  pinMode(ledPin10,OUTPUT);
  pinMode(ledPin11,OUTPUT);
  pinMode(speakerPin, OUTPUT);
}

```

```

char cadena[24];
byte contador=0;

```

En las líneas posteriores, nos encontramos de lleno con la función `loop`, en la cual se realiza constantemente comprobaciones a través de la condición `if-else`, para ver si la alarma está conectada o no, y así activar unos LEDs u otros. Dentro de la cláusula condicional, hacemos una llamada a la función `conectando`, ésta, no es más que una función que realiza una serie de parpadeos en los LEDs de la placa, para indicar que la alarma se está activando o desactivando.

```

void loop()
{

  if(conectada == false)
  {
    // la alarma esta desconectada
    digitalWrite(ledPin11,HIGH);
    digitalWrite(ledPin10,LOW);
    delay(1000);
    digitalWrite(ledPin10,HIGH);
    digitalWrite(ledPin11,LOW);
    delay(1000);
  }
}

```

```

activando = false;
digitalWrite(8,LOW);
Serial.println(" Alarma desconectada!!");
}
else
{
  // la alarma esta conectada

  if(activando == true)
  {
    // la alarma esta conectada y ya se ha activado
    digitalWrite(ledPin11,HIGH);
    digitalWrite(ledPin10,HIGH);
    Serial.println(" Alarma conectada!!");
  }
  else
  {
    // la alarma se va a activar, se pasa al proceso de conectando
    conectando();
  }
}
}

```

Una vez salimos del if-else nos encontramos con unas líneas de código muy importantes, ya que son las encargadas de recibir toda la información que recibe la placa vía serie, estas líneas comprueban que la transmisión vía serie esté disponible, y si lo está hace una comprobación de caracteres para ver la cadena introducida es la correcta. Para cumplir esto, se llama a la función “comando” la cual se encarga de realizar la comparación de las variables y de actualizar el estado de las variables.

```

if(Serial.available())
{
  memset(cadena, 0, sizeof(cadena));
  while (Serial.available()>0)
  {
    delay(5);
    cadena[contador]=Serial.read();
    contador++;
  }
  // cadena[contador]=0;
  contador=0;
}

```

```

if (comando()==true)
    {
        //la alarma esta conectada y ya podría sonar
        conectada = true;
        Serial.println("Alarma conectada!!!");
        digitalWrite(8,HIGH);
    }

}

}

boolean comando()

{

//aquí se comprueba si el código es correcto
boolean n=false;
for(int i=0; i<=4;i++)
    {

        if((cadena[i]==vstart[i]))
            {
                n=true;
            }
        else
            { Serial.println(" El código no es correcta");
              Serial.println(cadena);
              return false;
            }

    }
}

Serial.println("incorrecto");
delay(1000);
Serial.println(cadena);
if(conectada == true)
{
    //el código es correcto y como la alarma estaba conectada se desconecta
    conectada = false;
    Serial.println(" Alarma desconectada!!!");
    digitalWrite(8,LOW);
    conectando();
    activando = false;
    return false;
}
else return true;

```

```
}
```

Por último, tenemos la función que hacíamos referencia anteriormente, la función conectando, son unas pequeñas líneas de código que hacen más visual la activación y desactivación de la alarma, jugando con los pines digitales y con el tiempo de encendido y apagado de ellos, se pueden llegar a lograr grandes efectos visuales.

```
void conectando()
{
  // se va a proceder a la activación o desactivación de la alarma

  for(x = 0; x<= 10; x++)
  {
    digitalWrite(ledPin10,LOW);
    digitalWrite(ledPin11,LOW);
    digitalWrite(ledPin9,HIGH);
    delay(100);
    digitalWrite(ledPin9,LOW);
    digitalWrite(ledPin10,HIGH);
    delay(100);
    digitalWrite(ledPin10,LOW);
    digitalWrite(ledPin11,HIGH);
    delay(100);
    digitalWrite(ledPin11,LOW);
    digitalWrite(ledPin10,HIGH);
    delay(100);
    digitalWrite(ledPin10,LOW);
    digitalWrite(ledPin9,HIGH);
    delay(100);
    digitalWrite(ledPin9,LOW);

  }
  for(y = 5 ; y >= 1; y--)
  {
    Serial.println(y);

    delay(1000);
  }

  activando = true;

}
```

Éstas son las líneas que lleva cargada la memoria de nuestra placa Arduino UNO y la cual se ponen en marcha cada vez que recibe la corriente necesaria para su uso.

Una vez explicado el sketch utilizado para nuestro proyecto, pasamos a la App para el dispositivo móvil. Como se ha comentado en líneas anteriores, para programar Android se necesita conocer los conceptos de Java, y conocer la Api de Android que nos permite interactuar con dispositivos móviles y utilizar las funciones especiales que éstos llevan, sensores, osciloscopios, etc.

Para desarrollar la aplicación vamos a utilizar el programa Eclipse, un programa gratuito y multiplataforma, el cual nos ofrece la posibilidad de crear aplicaciones para Android, así como de poder probarlas a través de un emulador que podemos instalar.

Antes de nada, debemos instalar el SDK de Android, esto es simplemente el software de desarrollo de Android, que incorporamos a nuestro IDE de trabajo, el cual nos instala todas las bibliotecas y librerías que integran Android, es decir las Apis de Android. Por supuesto, todo es gratuito y multiplataforma. De hecho el proyecto se está realizando íntegramente en sistema operativo Mac de Apple.

La base de la aplicación consiste básicamente en conectar y desconectar la alarma mediante el envío de una cadena de caracteres. Primero el usuario se debe validar para poder acceder al menú de envío de la contraseña de activación.

Si el resultado es satisfactorio, el usuario podrá enlazar el dispositivo, cerrar la conexión Bluetooth, regresar al principio de la aplicación o enviar datos.

La primera activity de nuestra aplicación consiste en validar al usuario. Con este bloqueo en la aplicación, nos garantizamos que el usuario está autorizado para interactuar con la aplicación, y no es un usuario desconocido.



Ilustración 26: validación App del proyecto

En la aplicación, hemos creado el usuario “adri”, con contraseña “1234”.

Si no introducimos los datos correctamente, la aplicación nos hace un recuento hasta tres de errores de validación, al tercer error la aplicación finaliza, por seguridad.

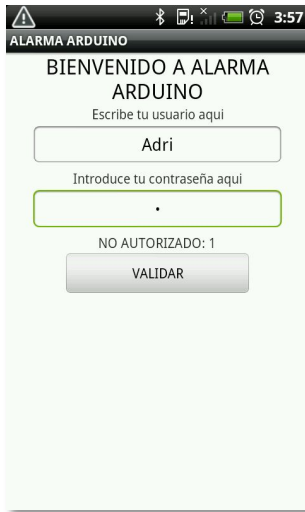


Ilustración 27: error de validación App proyecto

Si la validación se ha realizado de manera positiva, nos aparece una nueva “activity” con funcionalidades distintas que ahora comentamos.



Ilustración 28: solicitud de permiso App proyecto

Por defecto, si la conexión Bluetooth de nuestro dispositivo no está activada nos aparece un “warning”, indicándonos que la aplicación va a utilizar este tipo de conexión y nos da la opción de activar esta función.



Ilustración 29: menú App proyecto

Después de activar la función Bluetooth, nos encontramos de lleno con la base de nuestra aplicación.

Como se puede observar, consta de cuatro botones, y un apartado de texto editable para enviar la cadena que activa o desactiva la alarma.

Siempre que apretamos a los botones de enviar o cerrar, sin haber enlazado antes, nos aparecerá un mensaje indicando que debemos de enlazar el dispositivo previamente. Una vez hemos realizado el enlace con nuestro módulo en Arduino, la aplicación nos muestra un aviso, indicando el estado de la alarma en ese instante.



Ilustración 30: mensaje para enlazar



Ilustración 31: mensaje de alarma desconectada

A partir de ahí, ya podemos enviar mensajes, o cerrar la conexión, o salir de la activity hacia la pantalla anterior.

Si enviamos la clave correcta a nuestro Arduino, nos cambiará el estado de la alarma a "Alarma conectada" y podremos ver el resultado en la secuencia de LEDs.



Ilustración 32: mensaje alarma conectada

Capítulo 6: Test y pruebas

En el próximo capítulo se explican las pruebas realizadas para llegar al óptimo funcionamiento del proyecto. Tanto de la parte dedicada al hardware, como al software.

6.1 Arduino:

La realización de la parte mecánica con la placa Arduino, ha resultado ser bastante agradecida, puesto que no se ha realizado con muchas dificultades. Los componentes que integra el kit de Arduino funcionan a la perfección, tanto resistencias, como los LEDs, como los cables de conexión.

En la única ocasión que ha habido algún inconveniente, tal como ya se comentaba en algún capítulo anterior, ha sido con el módulo de conexión Bluetooth. El cual no recibía carga de la placa Arduino, después de cerciorarse una y otra vez de que el montaje era correcto mediante el datasheet disponible en la siguiente dirección web ([http://www.seeedstudio.com/wiki/Grove - Serial Bluetooth](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth)), se llegó a la conclusión de que el fallo era debido a la alimentación interna del módulo. Y se tomó la decisión de adquirir otra unidad de enlace entre Arduino y la aplicación Android, adquiriendo una Shield Bluetooth de la marca iTeed, a través de la siguiente página web, donde también se especifica el datasheet y las características técnicas (<http://imall.iteadstudio.com/im120417006.html>).

Posteriormente, el problema que se originó fue con el altavoz.

Inicialmente, el Buzzer que incluía el “Kit Inventor” no convenía como alarma, debido al poco volumen que emitía. Por consiguiente, se compró un altavoz el cual se activa con la presencia de un campo magnético, gracias al interruptor de lengüeta que lleva en el interior. Dicho altavoz, emite un sonido mucho más estridente, el cual se utiliza a modo disuasorio para cualquier persona no autorizada.

El principal problema que hubo, fue que el buzzer del kit de Arduino, no consumía apenas corriente, y el sonido, aunque flojo, se podía oír perfectamente. En cambio, el nuevo altavoz requería de mayor intensidad, el cual el circuito existente no podía proporcionar, y cuando se realizaron las pruebas pertinentes, el altavoz no llegaba a emitir el sonido indicado por el fabricante, sólo se apreciaba una pequeña vibración en el mismo, el cual indicaba que el funcionamiento no era el correcto, así que fue obligatoria la modificación de la estructura mecánica del proyecto, para añadir un transistor, el cual actúa como amplificador de señal.

Una vez añadido este elemento, el funcionamiento del prototipo es el esperado y se pasa a la creación de la aplicación propia en Android.

6.2 Android:

La aplicación creada en Android, se desarrolló sin ningún tipo de contratiempo en cuanto a la creación de menús, botones o validación de usuario. Los problemas empezaron a llegar en cuanto se quería realizar la conexión con el dispositivo Bluetooth. Pero después de mirar varios ejemplos de utilización e incluso distintos proyectos Arduino, que también precisaban de la utilización de módulos Bluetooth, se solventaron los errores, y la conexión con la placa UNO de Arduino se realizaba sin ningún tipo de problemas.

Una de las curiosidades que desconocía de la programación en Android, es que cada vez que se pasa de la vista vertical a horizontal, de la pantalla del terminal móvil, la aplicación pierde los datos guardados, o incluso el emparejamiento Bluetooth.

En esta ocasión, para la aplicación a la cual le vamos a dar uso, no necesita una vista apaisada para su funcionamiento, ya que no se logra obtener ningún beneficio.

Por ese motivo se desactivó el cambio de vista para la pantalla, bloqueando solamente la vista "portrait" o retrato, solventando el problema de raíz.

En esta ocasión la web para desarrolladores de Android, ofrece muchos ejemplos y toda la Api necesaria, para indagar en los recovecos de este sistema operativo, que ofrece multitud de características según la versión del mismo. Por ese motivo, no se ha ralentizado la programación de la aplicación.

Capítulo 7: Conclusiones

A partir de los capítulos anteriores, hemos descubierto como hemos desarrollado el proyecto, hasta llegar al objetivo final. En el siguiente capítulo, se explican las conclusiones, así como los objetivos que se han cumplido y los que no. Las fases que nos han causado más problemas, como se han solucionado y las que podrían mejorar más, aún si cabe, el proyecto actual.

7.1 Objetivos:

Cuando se comenzó a desarrollar el proyecto, los objetivos iniciales eran claros, el poder activar algún dispositivo acústico, que se semejara a una alarma de una manera inalámbrica, a través de una aplicación creada en Android. Y que hubiera algún tipo de indicador luminoso, el cual indicara el estado de esa alarma para el usuario. Finalmente, todos estos objetivos se han cumplido sin mayores dificultades.

Los indicadores luminosos, en este caso mediante LEDS rojos y amarillos indican cuando está la alarma activada, si los LEDS están encendidos de forma fija, o si los LEDS están parpadeando, nos indica que la alarma está desconectada. Cuando los LEDS realizan el proceso de encendido o apagado de la alarma, se produce una secuencia luminosa distinta a lo descrito anteriormente.

Otro de los puntos claves, era que el proyecto debía parecer a una alarma, y para ello, y después de probar el Buzzer que incluía nuestro kit inventor de Arduino sin convencer, se compró un altavoz con un sonido más similar al de un avisador acústico más disuasorio, que incorporaba un interruptor de lengüeta con un imán, que al separarlo, emitía un sonido mucho más estridente, el cual convenció para añadirlo al prototipo de Arduino.

Para finalizar el apartado mecánico, otro punto que hacía que el prototipo, fuera considerado como una alarma, era el no estar atado mediante cables a una fuente de corriente externa a ello, como el ordenador, o la propia corriente, así que se añadió una batería que adaptada al conector de corriente de Arduino, hacía que el prototipo fuese totalmente portátil, sin necesidad de dependencia de aparatología externa.

Estos son los objetivos más importantes de la parte mecánica.

De la parte software, obviamente eran los de realizar la conexión inalámbrica vía Bluetooth, y que la activación y desactivación de la alarma funcionaran correctamente.

En mi opinión, el punto clave del prototipo, era la conexión inalámbrica, este punto supuso algún que otro quebradero de cabeza, debido a la dificultad que conllevaba. Pero finalmente, después de indagar en la API de Android y probar una y otra vez distintos ejemplos de conexión, la comunicación inalámbrica se logró sin mayores problemas.

Una vez que se dominaba la comunicación sin cables, la preocupación se centraba en el envío de datos, que no supuso grandes problemas. Puesto que al realizarse una comunicación en serie, el sketch de Arduino lo recibía de la misma manera que a través del cable USB.

Es decir, que con la aplicación que ya había hecha con Arduino, ya nos podíamos comunicar inalámbricamente mediante la conexión Bluetooth. Ahora sólo se tenía que dedicar las horas necesarias para programar la aplicación personalizada para activar y desactivar la alarma.

7.2 Objetivos no cumplidos:

Como hemos comentado ya, los principales objetivos se han cumplido, pero es cierto, que inicialmente se pensó en realizar alguna otra característica para incorporar al prototipo final.

Uno de los objetivos no cumplidos ha sido el de no poder realizar la comunicación Bluetooth para los dispositivos de forma genérica, es decir, la aplicación está preparada para el módulo de iTeard del que disponemos, en otras palabras, nuestra App ya sabe a que módulo Bluetooth conectarse y que dispositivo quiere empezar a realizar la conexión. Después de realizar decenas de pruebas con la Api de Android y no lograr encontrar los emparejamientos Bluetooth de los dispositivos que están en el alcance, se pasó a introducir la Mac Adress única del teléfono móvil para realizar la conexión de manera única.

El punto positivo de este objetivo no cumplido podría llegar a ser una ventaja, ya que otro usuario con la misma aplicación no podría conectarse, ya que la Mac Adress no coincidiría con la que indica la App, por consiguiente, se consigue reforzar mucho la seguridad de toda la solución.

Por último, para realizar la comunicación Bluetooth entre el Arduino y cualquier aplicación, inicialmente se adquirió un módulo Bluetooth que se conectaba a la placa UNO vía cable. Después de cargar diversos sketches de configuración, de ver que recibía corriente y de hablar con el fabricante para comprobar que estaba bien conectado sin obtener ningún resultado de funcionamiento, se pasó a la compra de otro adaptador Bluetooth de otra marca, vía Shield, con esto se consiguió realizar la comunicación con el segundo dispositivo de una manera sencilla y simple, ya que la conexión que tiene es mucho más intuitiva que la anterior.

Estos han sido los dos contratiempos más importantes que nos hemos encontrado, con los cuales se ha logrado alcanzar una solución alternativa a la primera para lograr los objetivos principales.

7.3 Posibles mejoras :

En definitiva, las posibles mejoras se podrían contemplar en los objetivos no cumplidos. Pero además, y aprovechando la potencia de Arduino, si incorporáramos una cámara y un pequeño servo que a través de la aplicación de Android pudiéramos controlar de manera inalámbrica, y a su vez visualizar el contenido en la misma aplicación, esto dotaría a nuestro prototipo de un gran nivel de control y protección.

Otra de las posibles mejoras que se podrían realizar es la del desarrollo de la aplicación móvil para un entorno multiplataforma, ya funciona actualmente para dispositivos móviles Android (Teléfonos y tabletas), pero se podría llegar a implementar también para dispositivos iOS ya que también tienen mucha cuota de mercado.

En cuanto a la parte mecánica, los materiales que se han utilizado para realizar el prototipo son básicos, por ejemplo, el interruptor de lengüeta se podría intercambiar por algo más sofisticado, como un detector de movimiento, de ruido, o un sensor por infrarrojos. Todo esto, encarece más el precio del prototipo, por supuesto.

Otra de las posibles mejoras, puede ser la de añadir un display, táctil o no, para informar de forma visual al usuario, o para incluso también poder enviar la información de forma cableada a la placa Arduino.

También se podría conectar la Placa Arduino al circuito eléctrico del hogar a través de relés, para que cuando se activara la alarma por intrusión, se encendieran todas las luces de modo disuasorio, realmente.

Otra característica, en cuanto al Software, podría ser que cada vez que saltara la alarma, nos apareciera un mensaje de alerta en el teléfono y una opción rápida de llamar al número de la policía. Hay cientos de usos para este tipo de dispositivos, en el cual la imaginación hace el resto, sólo hace falta dedicar algo de tiempo y tener la inspiración adecuada

7.4 Conclusiones personales :

Para acabar, me gustaría expresar las conclusiones que se han sacado a lo largo de la realización del proyecto.

Con toda sinceridad puedo decir que he aprendido mucho realizando el proyecto durante estos meses. Primeramente he aprendido a realizar un proyecto de larga duración, el cual tienes que planear con antelación, paso a paso a través de sus fases, y que normalmente, esas fases nunca se acaban cumpliendo.

Por otro lado, considero que es un proyecto bastante completo, debido a que incorpora dos ramas muy importantes de la informática, y de la tecnología en general, como son: un parte de software, en la cual se crea una aplicación desde cero para

realizar una comunicación inalámbrica, y que se integre completamente con el dispositivo mecánico. Y otra parte mecánica, la cual realiza la acción controlada a través de un terminal móvil, y que se necesitan conocimientos de electrónica para realizar su montaje.

Para cada una de las dos ramas del proyecto, el nivel de aprendizaje ha sido altísimo, ya que desde el proyecto hacia atrás, no había programado nada para dispositivos Android, y de la misma forma, anteriormente al inicio del proyecto, mis conocimientos sobre la plataforma Arduino eran cero, realmente no sabía que existía.

Así que hoy por hoy puedo decir que mi inquietud por dominar estas dos ramas de la tecnología fue muy grande, hasta llegar a cumplir los objetivos. Y que gracias a haber realizado el proyecto, con toda seguridad iré realizando más pruebas para mejorar este prototipo o incluso otro nuevo muy distinto a este.

Sólo lamento la falta de tiempo para investigar más y mejores opciones para cumplir todas las metas que tenía en mente, pero estoy seguro que un futuro cercano podré lograrlas sin problemas.

Bibliografía:

http://www.uab.cat/Document/541/595/Normativa_PFCNovembre2010.pdf

Normativa de proyectos de ingeniería técnica.

<http://www.seguridad-privada.net/ley-seguridad-privada.html>

Normativa sobre la seguridad privada.

<http://www.areatecnologia.com/>

Teoría y conceptos sobre electrónica y tecnología.

<http://www.arduino.cc/es/>

Página oficial de Arduino, software, ejemplos, foro sobre Arduino.

<http://appinventor.mit.edu/>

Página oficial para descargar la aplicación Mit App Inventor para Android.

<http://tcrobotics.blogspot.com.es/2011/05/tutorial-arduino-uno-adk-de-google.html#more>

Ejemplo y tutorial sobre Android y Arduino.

<http://www.androidizados.com/desarrollo/2011/01/27/android-comenzar-a-desarrollar-aplicaciones/>

tutorial para empezar a desarrollar Android.

<http://es.scribd.com/doc/90209128/Libro-basico-Arduino>

Tutorial con ejemplos y conceptos sobre Arduino

<http://fritzing.org/>

Página oficial sobre la aplicación que emula a Arduino.

<http://www.basic4ppc.com/>

Página oficial sobre la aplicación Basic4Android.

<http://www.javahispano.org/android/2012/2/1/arduino-amarino-android.html>

Conceptos sobre la aplicación amarino para Android.

<http://www.aprendeandroid.com/menu.htm>

Curso sobre Android desde cero.

<http://www.ilektron.com/2011/04/arduino-el-puerto-serie-parte-i/>

Comunicación serie para Arduino.

<http://www.elboby.com/2011/10/esquema-arduino/>

Conceptos sobre la placa Arduino.

<http://blog.iteadstudio.com/play-bluetooth-with-arduino/>

Conceptos sobre el modulo Bluetooth de Arduino

<http://danimtzc.blogspot.com.es/2012/03/aplicacion-bluetooth-para-android.html>

Pasos para comunicar una aplicación Android por Bluetooth.

http://es.wikicode.org/index.php/ArrayAdapter_android

Crear un array en Android.

<http://android-er.blogspot.com.es/2011/05/get-list-of-paired-bluetooth-devices.html>

Enlazar dispositivos en una aplicación Android.

<https://www.youtube.com/watch?v=X-Z4TiJTtIY>

Video sobre Bluetooth en Android.

<http://bellcode.wordpress.com/2012/01/02/android-and-arduino-bluetooth-communication/>

Comunicación entre Android y Arduino por Bluetooth.

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Manual usuario Arduino

<http://www.arduino.cc/en/Guide/ArduinoXbeeShield>

Conceptos sobre la Shield Bluetooth de Xbee

<https://sites.google.com/a/divinechildhighschool.org/electronics/Home/Arduino-Lessons>

Ejercicios sobre Arduino.

<http://developer.android.com/index.html>

Página oficial para desarrolladores de Android.

<http://www.youtube.com/user/pekepowah/videos>

Videos sobre proyectos y tutoriales de Arduino

<http://www.cursomicros.com/arduino/20-proyectos-con-arduino.html>

Grandes proyectos sobre Arduino.

<http://www.microsoft.com/project/>

Microsoft Project:

http://www.es.gnu.org/P%C3%A1gina_Principal
GNU España.

ADRIÁN FERNÁNDEZ MEDINA

22/6/2013

A handwritten signature in blue ink, appearing to read 'Adrián', with a large, stylized flourish extending to the right.