



Máster Universitario en Software Libre

Curso Académico 2012/2013

Proyecto Fin de Máster

# Monitorización del consumo eléctrico de una vivienda con Android

Autor: Jesús Urcera López

Tutor: D. Roberto Calvo



***Agradecimientos***

*A mis compañeros del máster, en especial a Laura, Ilias, Antonio y Ricardo,  
por su inestimable ayuda a la hora de superar este reto.*

***Dedicatoria***

*Para mi familia, mi esposa Dioni y mis peques, Andrea, Jesús Adrián y Rodrigo,  
por ser tan pacientes y tolerantes conmigo.*





©2013 Jesús Urcera López. Algunos derechos reservados.

Este obra está bajo una Licencia Creative Commons

Atribución-CompartirIgual 3.0 Unported.

Para ver una copia de la licencia visitar:

[http://creativecommons.org/licenses/by-sa/3.0/deed.es\\_CO](http://creativecommons.org/licenses/by-sa/3.0/deed.es_CO).





# Índice general

0.1. Introducción . . . . .	19
0.2. Objetivos y campo de aplicación . . . . .	19
0.3. Descripción del trabajo . . . . .	20
<b>1. Introducción</b>	<b>21</b>
1.1. Sobre este documento . . . . .	21
1.1.1. Estructura del documento . . . . .	21
1.1.2. Ámbito . . . . .	22
1.1.3. Metodología . . . . .	22
<b>2. Objetivos</b>	<b>23</b>
2.1. Objetivos generales . . . . .	23
<b>3. Descripción del trabajo</b>	<b>25</b>
3.1. Android . . . . .	25
3.1.1. ¿Qué es Android? . . . . .	25
3.1.1.1. Arquitectura . . . . .	25
3.1.1.2. Historia . . . . .	27
3.1.1.3. Versiones . . . . .	28
3.1.1.4. Comparativas con otros sistemas operativos para móviles . . . . .	29
3.1.2. Comunidad . . . . .	30
3.1.2.1. Filosofía de gobierno . . . . .	31
3.1.2.2. Licencias . . . . .	31
3.1.2.3. Estadísticas . . . . .	31
3.2. Hardware . . . . .	33
3.2.1. Introducción . . . . .	34
3.2.2. Conceptos básicos . . . . .	34
3.2.3. Equipo . . . . .	35
3.2.3.1. Sensores de corriente . . . . .	36
3.2.3.2. Sensores de tensión . . . . .	39
3.2.3.3. Energy Meter . . . . .	41

3.2.3.4.	Microcontrolador . . . . .	42
3.2.3.5.	Memoria EEPROM . . . . .	43
3.2.3.6.	Pulsadores y leds . . . . .	43
3.2.3.7.	Aislador . . . . .	44
3.2.3.8.	Bluetooth . . . . .	45
3.2.3.9.	Fuente de alimentación . . . . .	45
3.2.4.	Firmware . . . . .	47
3.2.5.	Diagramas de bloques . . . . .	54
3.2.6.	Esquemas Electrónicos . . . . .	54
3.2.7.	Lista de materiales . . . . .	54
3.2.8.	Diagrama de conexionado del microcontrolador . . . . .	54
3.2.9.	Manual de usuario . . . . .	56
3.2.9.1.	Hardware . . . . .	56
3.2.9.2.	Firmware . . . . .	58
3.2.10.	Fotos del prototipo . . . . .	58
3.3.	Software . . . . .	59
3.3.1.	Introducción . . . . .	59
3.3.2.	Diagrama de bloques . . . . .	60
3.3.3.	Manual de usuario . . . . .	61
3.3.3.1.	Instalación . . . . .	61
3.3.3.2.	Uso . . . . .	62
3.4.	Entorno de desarrollo . . . . .	64
3.4.1.	Java . . . . .	64
3.4.2.	Entorno de desarrollo Eclipse y Android SDK ADT Bundle . . . . .	67
3.4.3.	Forja: GitHub . . . . .	73
3.4.4.	Otras herramientas . . . . .	74
<b>4.</b>	<b>Presupuesto</b> . . . . .	<b>77</b>
4.1.	Costes ejecución material . . . . .	77
4.1.1.	Coste del hardware . . . . .	77
4.1.1.1.	Componentes y materiales electrónicos . . . . .	77
4.1.1.2.	Equipamiento . . . . .	79
4.1.2.	Coste del software . . . . .	79
4.1.3.	Coste por tiempo empleado . . . . .	80
4.1.4.	Coste total de presupuesto de ejecución material . . . . .	80
4.2.	Gastos generales y beneficio industrial . . . . .	80
4.3.	Honorarios de redacción . . . . .	81
4.4.	Coste de ejecución por contrata . . . . .	81
4.5.	Importe total del presupuesto . . . . .	81

<i>ÍNDICE GENERAL</i>	9
<b>5. Conclusiones</b>	<b>83</b>
5.1. Lecciones aprendidas . . . . .	83
5.2. Trabajos futuros . . . . .	83
<b>A. Microcontrolador PIC18LF26K80</b>	<b>85</b>
<b>B. Módulo Bluetooth</b>	<b>95</b>
<b>C. Esquema electrónico</b>	<b>97</b>
<b>D. Contenido del CD/DVD</b>	<b>99</b>
<b>Bibliografía</b>	<b>101</b>



# Índice de figuras

1.	Diagrama de bloques . . . . .	20
3.1.	Arquitectura Android . . . . .	26
3.2.	Fraccionamiento en Android . . . . .	29
3.3.	Evolucion de ventas de smartphones por sistema operativo . . . . .	30
3.4.	Estadísticas del proyecto Android en Ohloh 1 . . . . .	32
3.5.	Estadísticas del proyecto Android en Ohloh 2 . . . . .	33
3.6.	Triángulo de potencias . . . . .	35
3.7.	Sensor de corriente de efecto Hall . . . . .	36
3.8.	Sensor de corriente inductivo tipo pinza CurrentCost . . . . .	37
3.9.	Sensor de corriente inductivo tipo pinza . . . . .	37
3.10.	Shunt resistivo de potencia . . . . .	37
3.11.	Shunt resistivo de baja potencia . . . . .	37
3.12.	Circuito de entrada del sistema. . . . .	38
3.13.	Cálculo de la resistencia shunt de sensado de corriente. . . . .	39
3.14.	Divisor de tensión resistivo. . . . .	40
3.15.	Circuito de sensado de tensión. . . . .	41
3.16.	Familia MCP39xx. . . . .	41
3.17.	Comparativa microcontroladores PIC. . . . .	43
3.18.	Comunicación serie. . . . .	45
3.19.	Cálculo de los valores de la fuente de alimentación. . . . .	46
3.20.	Fuente de alimentación sin transformador. . . . .	47
3.21.	Flujo de ejecución del programa del PIC. . . . .	48
3.22.	Diagrama de bloques . . . . .	54
3.23.	Conexionado del microcontrolador. . . . .	56
3.24.	Foto del prototipo. . . . .	58
3.25.	Otra vista del prototipo. . . . .	59
3.26.	Diagrama de bloques de HEMet . . . . .	60
3.27.	Arrancar la aplicación . . . . .	62
3.28.	Pantalla de inicio de HEMet . . . . .	63
3.29.	Pantalla de datos básicos de HEMet . . . . .	63

3.30. Instalación de Java 1 . . . . .	65
3.31. Instalación de Java 2 . . . . .	65
3.32. Instalación de Java 3 . . . . .	66
3.33. Instalación de Java 4 . . . . .	66
3.34. Instalación de Java 5 . . . . .	67
3.35. Instalación de Java 6 . . . . .	67
3.36. Instalación de Eclipse 1 . . . . .	68
3.37. Instalación de Eclipse 2 . . . . .	68
3.38. Instalación de Eclipse 3 . . . . .	69
3.39. Instalación de Eclipse 4 . . . . .	69
3.40. Instalación de Eclipse 5 . . . . .	69
3.41. Instalación de Eclipse 6 . . . . .	70
3.42. Instalación de Eclipse 7 . . . . .	70
3.43. Instalación de Eclipse 8 . . . . .	71
3.44. Configuración de un AVD 1 . . . . .	71
3.45. Configuración de un AVD 2 . . . . .	72
3.46. Configuración de un AVD 3 . . . . .	72
3.47. Creación de un repositorio en GitHub 1 . . . . .	73
3.48. Creación de un repositorio en GitHub 2 . . . . .	74
3.49. Creación de un repositorio en GitHub 3 . . . . .	74
A.1. Patillaje del PIC18LF26K80 . . . . .	86
B.1. Módulo Bluetooth . . . . .	96



# Índice de cuadros

3.1. Comparativa de tipos de sensores de corriente. . . . .	38
3.2. Listado de componentes y materiales electrónicos. . . . .	55
4.1. Costes de hardware. Componentes y materiales electrónicos. . . . .	78
4.2. Costes de hardware. Equipamiento. . . . .	79
4.3. Costes de Software. . . . .	79
4.4. Coste por tiempo empleado. . . . .	80
4.5. Coste total del presupuesto de ejecución material. . . . .	80
4.6. Gastos generales y beneficio industrial. . . . .	80
4.7. Honorarios de redacción. . . . .	81
4.8. Coste de ejecución por contrata. . . . .	81
4.9. Importe total del presupuesto. . . . .	81



# Resumen

Este trabajo está basado en mi proyecto fin de grado titulado “Sistema de monitorización del consumo de energía eléctrica de una vivienda con Android”. En este trabajo se ha hecho un desarrollo en Android totalmente nuevo y con características avanzadas de la parte de visualización y gestión del equipamiento hardware. El proyecto global pretende desarrollar un sistema que se conecta en el cuadro general de mando y protección de la vivienda para obtener una serie de parámetros (tensión, corriente, potencia, consumo, factor de potencia, . . . ) que, una vez analizados, nos permitan hacer un uso más eficiente de los recursos energéticos eléctricos de nuestra vivienda. Tanto la visualización como la gestión del hardware se realiza mediante la aplicación Android que es el objetivo de este trabajo.

El código fuente de la aplicación, llamada HEmet (Home Energy Meter), está disponible en el repositorio público de software <https://github.com/jurcera/HEmet.git> bajo licencia GPLv3+.

**Palabras clave:** Android, Bluetooth, energía eléctrica, consumo eléctrico, domótica, energía, eficiencia energética.



# Abstract

This work is based on my final degree project entitled “Power consumption monitoring system of a house with Android”. This paper has made a whole new Android development and advanced features of the display and management of hardware equipment. The overall project aims to develop a system that connects to the general control and protection of housing for a range of parameters (voltage, current, power consumption, power factor, Idots) that, once analyzed, will to allow more efficient use of electrical energy resources of our house. Both viusalización as hardware management is done by Android application which is the objective of this work. This paper aims to develop a system that connects in the general control and protection of housing for a range of values (voltage, current, power consumption, power factor, ...). All this information can be displayed in real time on an Android device and can be stored to detect and study possible incorrect consumption and solve them.

The application source code, called Hemet (Home Energy Meter), is available in the public repository <https://github.com/jurcera/HEmet.git> software under GPLv3 + license.

**Keywords:** Android, Buetooth, electric energy, power consumption, home automation, energy, energy efficiency.



# Resumen extendido

## 0.1. Introducción

Hasta no hace mucho tiempo nos podíamos permitir el lujo de desperdiciar recursos energéticos ya que estábamos en período de vacas gordas, pero en la actualidad, que las vacas flacas surgen por doquier, tenemos que ser más eficientes, no sólo eficaces, para obtener lo mejor posible con el mínimo consumo de recursos. Para realizarlo debemos empezar por ver cuánto se está consumiendo para, después de analizar los datos obtenidos, proponer mejoras que nos conduzcan a la eficiencia. Como decía William Thomson, más conocido como Lord Kelvin, “Lo que no se define no se puede medir. Lo que no se mide, no se puede mejorar. Lo que no se mejora, se degrada siempre”.

## 0.2. Objetivos y campo de aplicación

Como hemos visto en el apartado anterior necesitamos conocer, de una forma lo más precisa posible, cómo se realiza el consumo eléctrico de la vivienda para decidir que acciones tomar a la hora de reducirlo. Muchas veces no nos damos cuenta de que se está realizando un mal uso de un elemento hasta que no lo observamos en detalle. Con este sistema se pretende medir, en una primera fase de forma global y en fases sucesivas circuito a circuito, el consumo que se está realizando a lo largo del día para, una vez analizados los datos obtenidos, ver posibles correcciones para su disminución.

En principio, el campo de aplicación está restringido a viviendas ya que los sensores están calculados para valores de corriente desde décimas de amperio (menos de 10w) hasta el medio centenar de amperios (11.500 W) más o menos. Si nos interesa detectar potencias mayores, en pequeños negocios o comercios, tendremos que usar sensores que soporten corrientes más altas pero en detrimento de la precisión en muy bajas corrientes. El sistema funcionará igual sólo habrá que calibrarlo para el nuevo sensor y ya estaría operativo, eso sí, no detectaremos consumos bajos ya que los sensores tampoco lo harán.

### 0.3. Descripción del trabajo

El sistema consiste en un circuito que se conecta en el cuadro general de mando y protección de la vivienda para obtener los valores de tensión, corriente, potencias (activa, reactiva y aparente), consumos (activo y reactivo) y factor de potencia. A esta información se podrá acceder en tiempo real mediante un dispositivo Android y también se irá almacenando para generar gráficas de uso con el objetivo de detectar posibles usos incorrectos y solventarlos en la medida de lo posible.

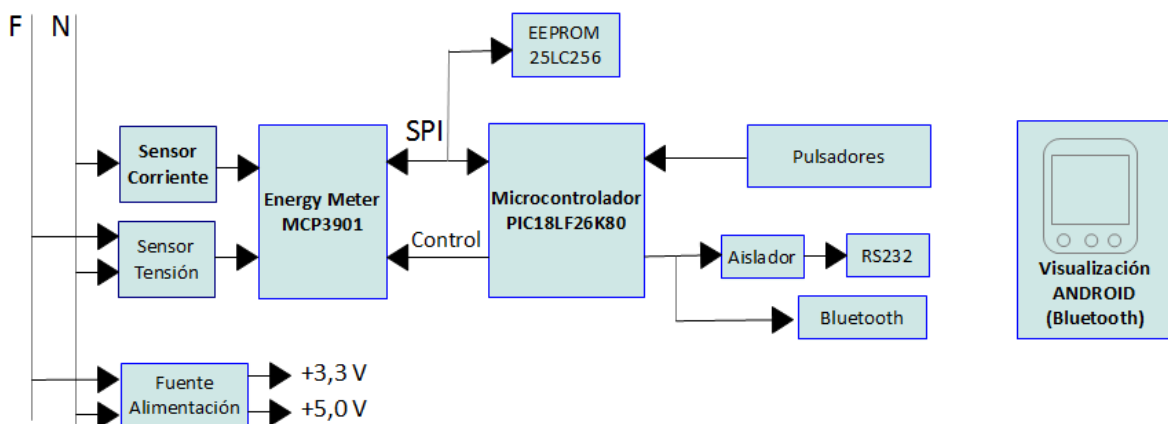


Figura 1: Diagrama de bloques

En el diagrama de la figura 1 podemos ver todos los bloques de los que consta el sistema, que describiremos, con más o menos detalle, a lo largo del trabajo. Entre los reseñables estarían: el sensor de la corriente eléctrica, elemento que se ha de elegir muy bien dependiendo de lo que se desee medir y que en nuestro caso se trata de un shunt resistivo; el Energy Meter (MCP3901), que consiste en un AFE (Analog Front End) de Microchip que dispone de dos ADC síncronos de 16/24 bits de resolución y de tipo Delta-Sigma, diseñado específicamente para este tipo de tareas; el microcontrolador (PIC18LF26K80) que se encarga de toda la parte de calibración y control del Energy Meter, así como, de las funciones de almacenado y envío de datos al elemento visualizador y, para finalizar, el elemento de visualización o de presentación de la información que consistirá en un dispositivo Android tipo teléfono móvil o tableta con conexión Bluetooth. El resto de elementos no son menos importantes pero no se va a profundizar tanto en ellos.



# Capítulo 1

## Introducción

El objetivo principal de esta obra es describir un sistema completo, aunque haciendo principal incapié en la parte de desarrollo Android, que se puede usar para la monitorización del consumo eléctrico de una vivienda o en su defecto de cualquier otro recinto con una entrada única de alimentación con una limitación máxima de potencia de 15.000 w y monofásica.

### 1.1. Sobre este documento

En esta sección vamos a describir cada una de las partes en las que se va a dividir el presente documento.

#### 1.1.1. Estructura del documento

El documento se divide en seis grandes capítulos.

1. Introducción. Habla del propio documento, de su estructura y su ámbito de aplicación.
2. Objetivos. ¿Qué se desean conseguir con este trabajo?
3. Descripción del trabajo. Partes en las que se divide.
  - Android. ¿Qué es?, su historia, sus revisiones y su comunidad.
  - Hardware. Descripción del hardware
  - Software. Descripción del software objeto del presente trabajo.
  - Entorno de desarrollo Android. Elementos necesarios para el desarrollo y para poder compartir el código.
4. Presupuesto. Costes de realización del presente trabajo.
5. Conclusiones. Qué conclusiones se han sacado del presente trabajo.

6. Apéndices. Información ampliada de algún elemento en particular.

### **1.1.2. Ámbito**

Este documento pretende dar la información básica necesaria para la construcción de un equipo similar pero es necesario acudir a las fuentes indicadas en la bibliografía si se pretende ahondar más en el tema.

### **1.1.3. Metodología**

Para realizar el presente documento se ha usado documentación citada en la bibliografía además de mi propia experiencia. Para la creación del presente documento se ha utilizado  $\text{\LaTeX}$ .

# Capítulo 2

## Objetivos

### 2.1. Objetivos generales

El objetivo principal de este documento es ofrecer los conocimientos necesarios para la construcción de un equipo de medición de consumo eléctrico para una vivienda, enfocándonos principalmente en el desarrollo de la parte de monitorización y control desde un dispositivo móvil con Android.

Además del desarrollo del sistema se pretende compartir toda esta información para que, al que le pueda interesar la use o colabore con el proyecto para mejorarlo y ampliarlo.



# Capítulo 3

## Descripción del trabajo

En este capítulo vamos a tratar de describir todos los elementos que forman, en sí, el sistema. Comenzaremos dando unas pinceladas sobre Android, después pasaremos a describir el hardware, el software que corre en el dispositivo Android y para finalizar, hablaremos sobre el entorno de desarrollo utilizado.

### 3.1. Android

Comenzamos con una pequeña descripción de Android.

#### 3.1.1. ¿Qué es Android?

Android es un sistema operativo basado en Linux y pensado inicialmente para teléfonos móviles pero que ahora se está implementando en otro tipo de dispositivos como tabletas, netbooks, relojes, televisión por internet, etc. Una característica fundamental de este sistema operativo es que es completamente libre, es decir, ni para programar en él ni para incluirlo en un teléfono hay que pagar nada, por eso está teniendo tanta difusión entre los fabricantes de dispositivos porque se produce un ahorro importante de costes, ya que, no necesitan desarrollar un sistema operativo propio.

Cualquiera puede bajarse el código fuente, revisarlo, compilarlo e incluso cambiarlo y distribuirlo sin necesidad de pagar nada. Para los usuarios es importante ya que al ser un sistema abierto la detección de fallos y agujeros de seguridad es más rápida y a los fabricantes les permite adaptarlo a sus terminales con gran rapidez.

##### 3.1.1.1. Arquitectura

La arquitectura de Android se muestra en la Figura 3.1 donde se puede apreciar de que se trata de una arquitectura por capas, 4 en particular, estando todas ellas basadas en software libre.

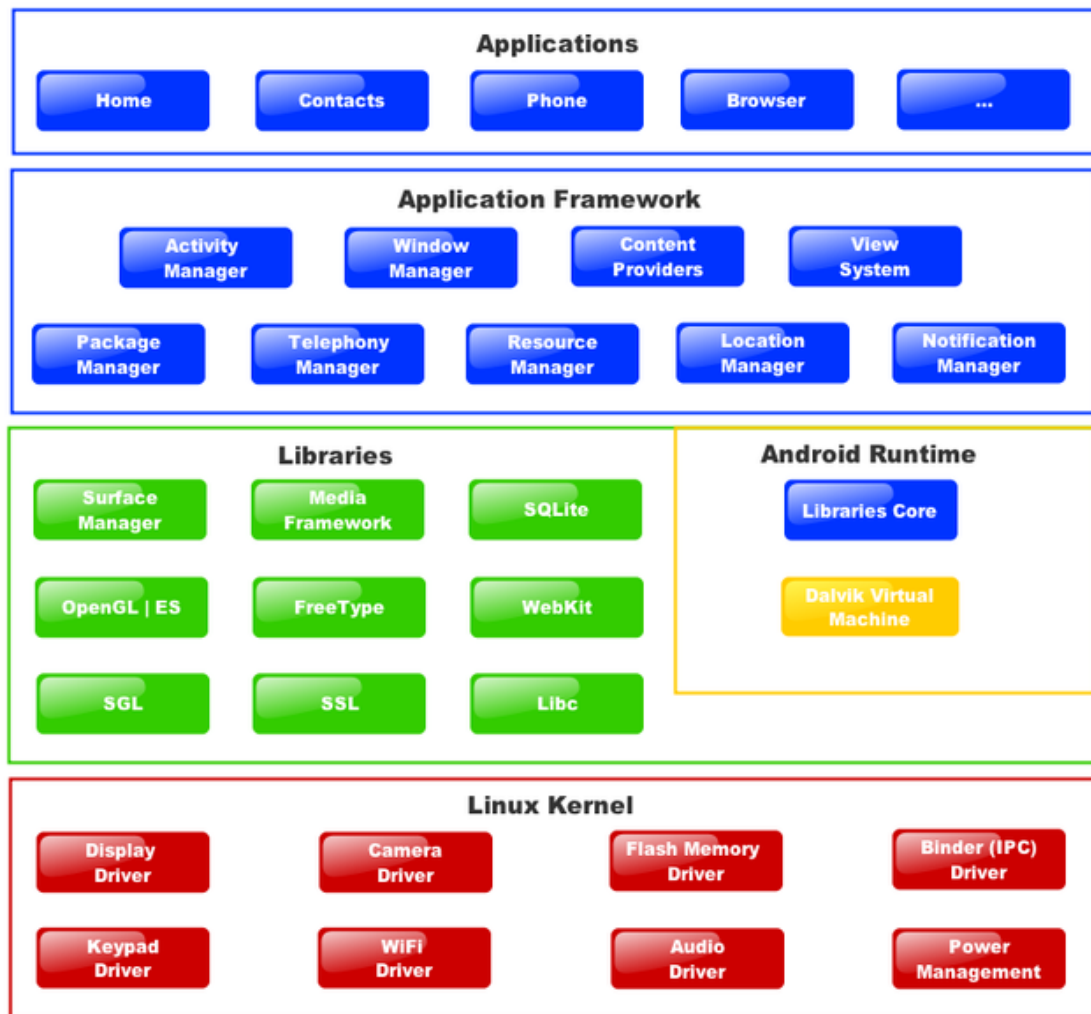


Figura 3.1: Arquitectura Android

En la capa más baja está el kernel de Linux. Actúa como una capa de abstracción entre el hardware y la capa siguiente. Se encarga de tareas como: la seguridad, el manejo de memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos.

La siguiente capa está formada por dos partes, una es el Android Runtime y la otra son las librerías nativas (Libraries). En el runtime se encuentran la máquina virtual Dalvik, desarrollada por Google usando el concepto de máquina virtual de java pero optimizada para trabajar en dispositivos con recursos muy reducidos, y una serie de librerías core de java. Las librerías nativas son librerías C/C++ compiladas para el propio microprocesador y que se usan en componentes de Android como webkit, 3D, SQLite, SSL, ...

En la capa superior (Application Framework) está la plataforma que proporciona una serie de elementos para facilitar el desarrollo de aplicaciones de forma eficiente y potenciando la re-

utilización de componentes, es decir, si ya existe una aplicación que me ofrece un determinado servicio puedo usarlo sin necesidad de reescribir uno nuevo. El desarrollo de aplicaciones se realiza usando el lenguaje de programación Java.

En la última capa están las aplicaciones instaladas en la máquina. Todas ellas corren sobre la máquina virtual Dalvik para garantizar la seguridad del sistema.

### 3.1.1.2. Historia

Android, Inc. fue fundada en Palo Alto, California, en octubre de 2003 por Andy Rubin, Rich Miner, Nick Sears y Chris White para desarrollar, en palabras de Rubin “dispositivos móviles inteligentes capaces de saber la ubicación y las preferencias de su propietario”. En principio pensaban en desarrollar un sistema operativo para cámaras digitales pero viendo el poco mercado que había se decidieron por desarrollar un sistema operativo para teléfonos inteligentes para competir con Symbian y Windows Mobile (iPhone todavía no existía). A pesar de los últimos logros, Android Inc. operaba en secreto, revelando solamente que estaba trabajando en software para teléfonos móviles.

Google adquirió Android Inc., el 17 de agosto de 2005, por lo que es una compañía perteneciente a Google. Los empleados clave de Android Inc., incluyendo Rubin, Miner y White, se quedaron en la empresa después de la adquisición. Se especuló que Google estaba planeando entrar en el mercado de la telefonía móvil con este movimiento. En Google, el equipo liderado por Rubin desarrolló una plataforma para un dispositivo móvil sobre el kernel Linux. Google ofrece la plataforma a los fabricantes de teléfonos y a los operadores de telefonía con la promesa de disponer de un sistema actualizable y flexible. Google había agrupado a fabricantes de hardware y socios de software e indicó a los operadores que estaba abierto a diversos grados de cooperación por su parte.

Se especulaba sobre las intenciones de Google tenía para entrar en el mercado de comunicaciones móviles continuó creciendo hasta diciembre de 2006. Los informes de la BBC y del Wall Street Journal señalaron que Google quería tener su buscador y sus aplicaciones en los teléfonos móviles y estaba trabajando duro para conseguirlo. Los medios de comunicación escrita y online informaron de los rumores de que Google estaba desarrollando su propio teléfono marca Google. Google empezó a definir especificaciones técnicas que mostró a fabricantes de teléfonos y operadores de telecomunicaciones. En septiembre de 2007, InformationWeek publicó un estudio de Evalueserve indicando que Google había presentado varias solicitudes de patentes en el área de la telefonía móvil.

El 5 de noviembre de 2007, la Open Handset Alliance, un consorcio de empresas de tec-

nología como Google, fabricantes de dispositivos como HTC y Samsung, operadores móviles como Sprint Nextel y T-Mobile y fabricantes de chips como Qualcomm y Texas Instruments, se dio a conocer con el objetivo de desarrollar estándares abiertos para dispositivos móviles. Ese día, Android fue presentado como su primer producto, una plataforma móvil basada en la versión del kernel Linux 2.6. El primer teléfono disponible en el mercado capaz de ejecutar Android fue el HTC Dream, lanzado el 22 de octubre de 2008.

Desde 2008, Android ha visto numerosas actualizaciones que han mejorado gradualmente el sistema operativo, añadiendo nuevas funcionalidades y corrigiendo errores de versiones anteriores. Cada versión principal recibe el nombre, en orden alfabético ascendente, de un postre o algo dulce, por ejemplo, la versión 1.5 Cupcake fue seguido por 1.6 Donut. La última versión es la 4.2 Jelly Bean.

En 2010, Google lanzó la serie de dispositivos Nexus formada por una línea de teléfonos inteligentes y tabletas con sistema operativo Android y construido por un socio fabricante. HTC colaboró con Google para lanzar el primer teléfono inteligente Nexus, el Nexus One. La serie ya ha sido actualizado con los nuevos dispositivos, como el teléfono Nexus 4 (LG) y las tabletas Nexus 7 (Asus) y Nexus 10 (Samsung). Google libera los teléfonos Nexus y tabletas para que actúen como sus dispositivos Android emblemáticos, demostrando último software de Android y características de hardware.

El 13 de marzo de 2013, Larry Page anunció en un blog que Andy Rubin se había cambiado de la división de Android para asumir nuevos proyectos en Google. Fue reemplazado por Sundar Pichai, quien también continúa su papel de cabeza de la división de Chrome de Google, que desarrolla Chrome OS.

### **3.1.1.3. Versiones**

Una de las principales desventajas de Android con respecto a la competencia es el fraccionamiento, es decir, la cantidad de versiones en uso que podemos encontrar del sistema en el mercado. Existen dispositivos con Android 2.1, 2.2, 2.3.3, 3.2, 4.0.3, etc. pero también algunos que le han introducido su propia capa de aplicación para que al final no se parezca casi en nada a su origen. Además existen versiones para tabletas (3.x) y para teléfonos inteligentes (2.x), lo que provoca una tremenda confusión, sobre todo para los administradores de dichos equipos, para los desarrolladores y para las personas que tienen que elegir entre unos y otros.

La versión de Android 3.x (Honeycomb) salió para dar cobertura a las tabletas que disponían de una pantalla mucho más grande que los teléfonos inteligentes, pero no tuvo mucho éxito ya que hay menos de un 0,2 % de instalaciones.



Con la aparición de Android Ice Cream Sandwich (4.x), se unificaron las versiones para tabletas y para teléfonos inteligentes pudiéndose ya usar en la mayoría de los terminales. Lo que si debemos tener claro es la capacidad de actualizarse del terminal, ya que en muchos casos los terminales no reciben actualizaciones y se quedan anclados en versiones obsoletas. Tenemos un dispositivo de gama alta que no podemos sacarle todas las prestaciones por falta de soporte de la marca o del operador de telecomunicaciones.

En la Figura 3.2 podemos apreciar el fraccionamiento viendo el porcentaje de uso de las distintas versiones de Android en junio de 2013. Fuente: [ItOHA13]

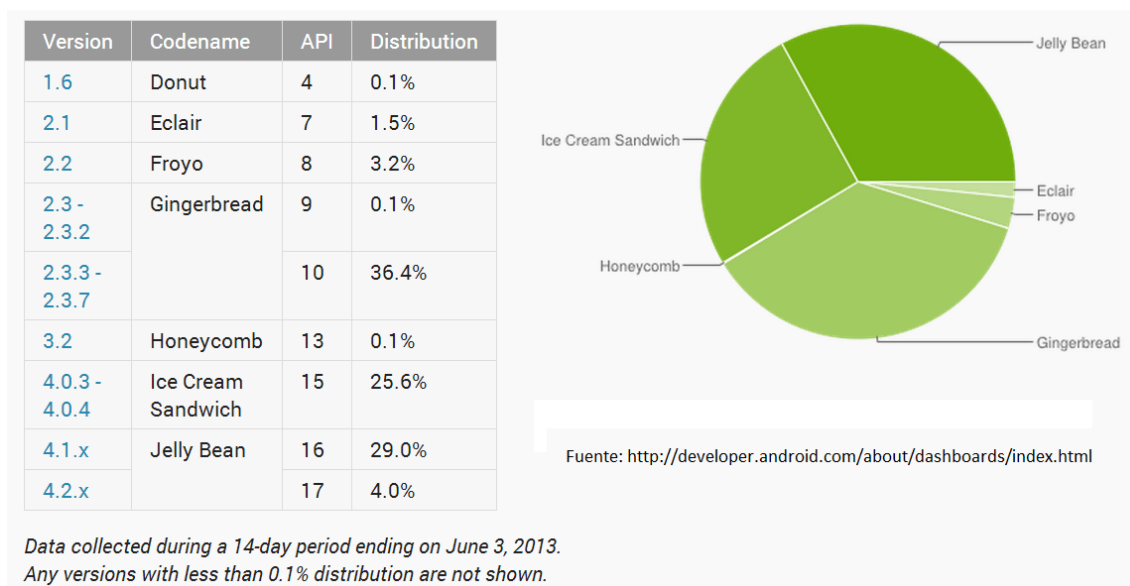


Figura 3.2: Fraccionamiento en Android

#### 3.1.1.4. Comparativas con otros sistemas operativos para móviles

En la Figura 3.3 podemos ver el porcentaje de ventas de teléfonos inteligentes a nivel mundial para el primer cuarto de 2012 (en azul) y para el primer cuarto de 2013 (en naranja). El porcentaje de ventas para Android ha pasado de un 57 % de 2012 a un 75 % en 2013. Caen todos y el que más Symbian (Nokia) que pasa de un 8,5 % a un 0,6 %. El dato curioso es Microsoft que crece de 1,9 % a 2,9 %. La fuente de los datos es Gartner<sup>1</sup> [Gar13].

<sup>1</sup><http://www.gartner.com/newsroom/id/2482816>

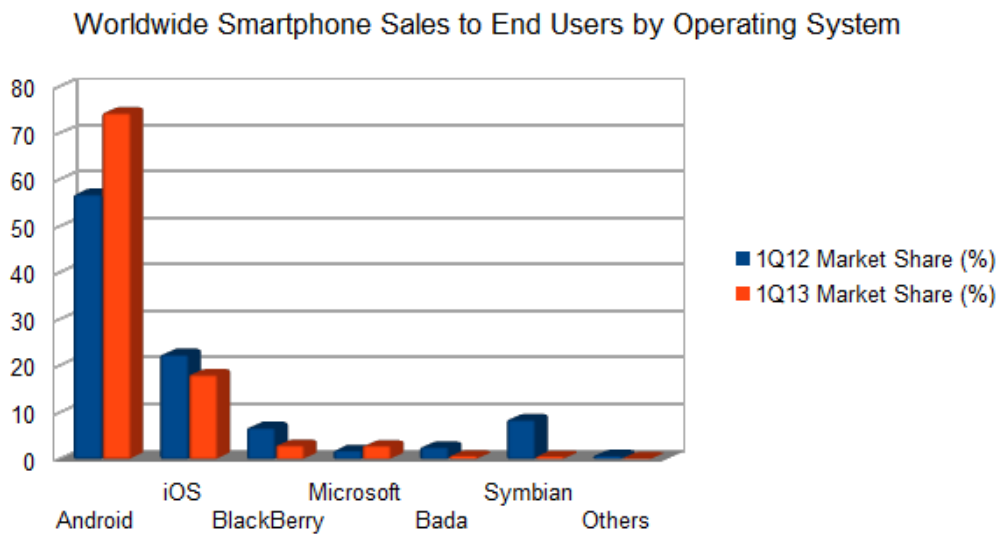


Figura 3.3: Evolucion de ventas de smartphones por sistema operativo

Claramente se puede ver, que a fecha de hoy, Android es imparable alcanzando una cuota de mercado muy alta y teniendo en cuenta que su principal competencia (Apple) está decayendo.

### 3.1.2. Comunidad

La comunidad de Android<sup>2</sup> no es una comunidad al uso dentro de los proyectos de software Libre ya que está totalmente controlada por una empresa, Google. Siendo una característica suya que la última versión del código sólo está disponible para las empresas que participan en el proyecto, para darles una ventaja tecnológica en agradecimiento a sus aportaciones al proyecto, estando justo la versión anterior disponible para el público en general. Aún así la comunidad, como veremos más adelante, no es nada despreciable.

Según Google, Android es una pila de software de código abierto creado para una amplia gama de dispositivos de diferentes formatos. El propósito principal de Android es la creación de una plataforma de software abierto disponible para operadores de telefonía, fabricantes de dispositivos y desarrolladores de código para que sus ideas innovadoras se conviertan en realidad y así crear un producto físico de éxito que mejore la experiencia en movilidad de los usuarios finales. Otra prioridad es asegurar de que no hay ningún punto único de fallo evitando que una única compañía pueda controlar o restringir las innovaciones de cualquier otra. El resultado es un producto de consumo completo y de calidad cuyas fuentes son libres y están disponibles para favorecer la personalización y la portabilidad. Cualquiera puede obtener el código fuente (no la última revisión), modificarlo para que se adapte a sus necesidades y generar un producto

<sup>2</sup><http://source.android.com/>

### 3.1.2.1. Filosofía de gobierno

Android fue creado por una alianza de empresas denominada Open Handset Alliance, lideradas por Google. Hoy en día tanto las empresas de dicho grupo y como muchas otras han invertido muchos recursos, principalmente tecnológicos, para mejorar Android y para poder poner dispositivos Android en el mercado. Las empresas que ha participado creen que es necesario que exista una plataforma abierta por ello se han unido para compartir esfuerzos y crear un producto común de calidad que cualquiera pueda adaptar y personalizar.

Para evitar incompatibilidades, debido a que cualquiera puede hacer lo que quiera, Android gestiona un programa de compatibilidad que todos los que quieran llevar la etiqueta de “Android compatible” deben cumplir. También se aconseja a los fabricantes de dispositivos que cumplan con el Programa de compatibilidad, anteriormente indicado, para que, además de la etiqueta, puedan usar el ecosistema de aplicaciones que se han desarrollado alrededor de Android.

El proyecto de código abierto, como ya hemos dicho, está liderado por Google quien mantiene y desarrolla Android, si bien Android, está compuesto por varios subproyectos que van por su lado. Google gestiona Android como un único producto de software no como un conjunto de piezas reemplazables. La intención es que los fabricantes de dispositivos adopten Android para sus productos sin necesidad de que tengan que preocuparse de implementar una especificación determinada porque Android les proporciona todo lo que necesitan.

### 3.1.2.2. Licencias

La licencia preferida del proyecto open source Android es la Apache Software License, 2.0 (“Apache 2.0”) y la mayor parte del software de Android tiene licencia Apache 2.0. Si bien el proyecto intenta usar la licencia de preferencia puede haber excepciones que se manejan caso por caso. Por ejemplo, los parches del kernel de Linux están bajo la licencia GPLv2.

¿Por qué licencia Apache?. El principal motivo para usar licencia Apache en el entorno de usuario (no en el kernel) es que, tal y como están fomentando el desarrollo de dispositivos abiertos y modificables no creen que deban obligar, a la gente que modifique Android, a usar una licencia determinada, como ocurriría en el caso de usar LGPL.

### 3.1.2.3. Estadísticas

Como he dicho anteriormente, a pesar de no ser una comunidad al uso, Android goza de buena salud tal y como podemos apreciar en los datos mostrados en las figuras 3.4 y 3.5.



Figura 3.4: Estadísticas del proyecto Android en Ohloh 1

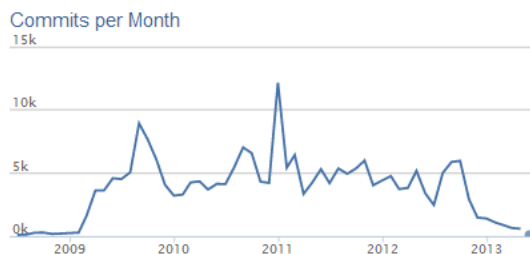
Como resumen podemos decir que:

- Que desde julio de 2008 se han escrito mas de 15 millones de líneas de código.
- Está principalmente escrito en lenguaje C.
- Que 5 años de desarrollo real equivalen a casi 5.000 años de esfuerzo según el modelo COCOMO.
- Coste estimado de 270 millones de dólares según el modelo COCOMO.
- Media de 5.000 commits/mes hasta octubre de 2012 que está cayendo hasta en los últimos meses hasta 600 commits/mes.

- La mayoría de los contribuidores de hallan en Europa y Norteamérica, al igual que los usuarios.

**Activity**

<p><b>30 Day Summary</b>                  May 12 2013 — Jun 11 2013                  426 Commits                  53 Contributors                  including 6 new contributors</p>	<p><b>12 Month Summary</b>                  Jun 11 2012 — Jun 11 2013                  30611 Commits                  Down -25340 (45%) from previous 12 months                  536 Contributors                  Down -7 (1%) from previous 12 months</p>
---	---



**Community**

**Ratings**

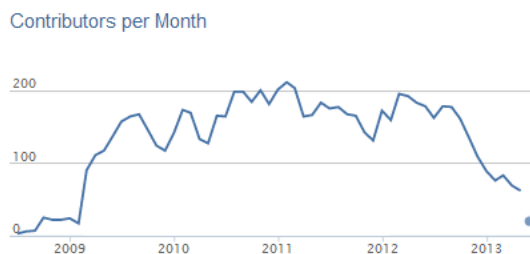
119 users rate this project:  
 ★★★★★ 4.4/5.0

Click to add your rating  
 ☆☆☆☆☆

Review this Project!

**Most Recent Contributors**

Xavier Ducrohet	Tor Norbye
...n D. Carlstrom	Andrew Hsieh
Wink Saville	Ben Cheng



<b>Project Summary</b> News Settings Sharing Widgets Related Projects	<b>Code Data</b> Languages Cost Estimates	<b>SCM Data</b> Commits Contributors	<b>Community Data</b> Users Ratings & Reviews User & Contributor Locations
---	---	--	---

[About](#) [Forums](#) [Terms](#) [Privacy](#)

Copyright © 2013 Black Duck Software, Inc. and its contributors. Some Rights Reserved. Unless otherwise marked, this work is licensed under a [Creative Commons Attribution 3.0 Unported License](#).  
 Ohloh® and the Ohloh logo are trademarks of Black Duck Software, Inc. in the United States and/or other jurisdictions. All other trademarks are the property of their respective holders.



Figura 3.5: Estadísticas del proyecto Android en Ohloh 2

La fuente de datos es ohloh [BDS13].

## 3.2. Hardware

En esta sección vamos a describir los elementos que guardan relación con el Hardware del sistema.

### 3.2.1. Introducción

Con este sistema se pretende obtener la información de consumo eléctrico de una vivienda con el objetivo de detectar posibles problemas e intentar corregirlos. En la situación socio-económica actual cualquier ahorro, por pequeño que sea, será siempre bienvenido. Además estaremos colaborando con la sostenibilidad del planeta y a reducir el efecto invernadero y, por ende, el calentamiento global. Parecen todas ventajas pero, además se ha tenido muy presente, en la elección de componentes, el factor precio, eso sí, sin reducir la seguridad ni la fiabilidad, para conseguir un producto al alcance de cualquiera, ya que, la tecnología debe estar al alcance de todos.

En los siguientes apartados, comenzaremos por un poco de teoría básica y luego iremos comentando uno a uno todos los bloques en que se ha dividido la parte hardware del sistema, dando explicaciones de la metodología empleada para la elección de uno u otro elemento y para finalizar hablaremos del firmware, la otra parte imprescindible del equipo.

### 3.2.2. Conceptos básicos

La fórmula para saber la energía eléctrica que se está consumiendo en una vivienda nos dice que dicha energía es igual al producto de la potencia absorbida por la cantidad de tiempo que se ha requerido esa potencia. Por lo tanto vamos a comenzar definiendo la potencia, sus tipos y cómo saber la eficiencia de una instalación.

La potencia nos indica la cantidad de energía absorbida o entregada por un elemento por unidad de tiempo. En una instalación hay tres tipos de potencia disponibles, representadas por el famoso triángulo de potencias:

- Potencia activa que se representa por la letra  $P$  y se mide en vatios [W] o kilovatios [kW]. Es debida a los elementos resistivos puros con una bombilla de incandescencia.
- Potencia reactiva que se representa por la letra  $Q$  y se mide en voltamperios reactivos [VAr] o kilo voltamperios reactivos [kVAr]. Es debida a los elementos almacenadores de energía como bobinas y condensadores que se encuentra, por ejemplo, en un motor eléctrico que es un ejemplo de bobina.
- Potencia aparente que se representa por la letra  $S$  y se mide en voltamperios [VA] o kilo voltamperios [kVA]. Es la suma vectorial de las potencias activa y reactiva.

En el triángulo de potencias podemos ver gráficamente las relaciones entre las tres potencias definidas anteriormente.

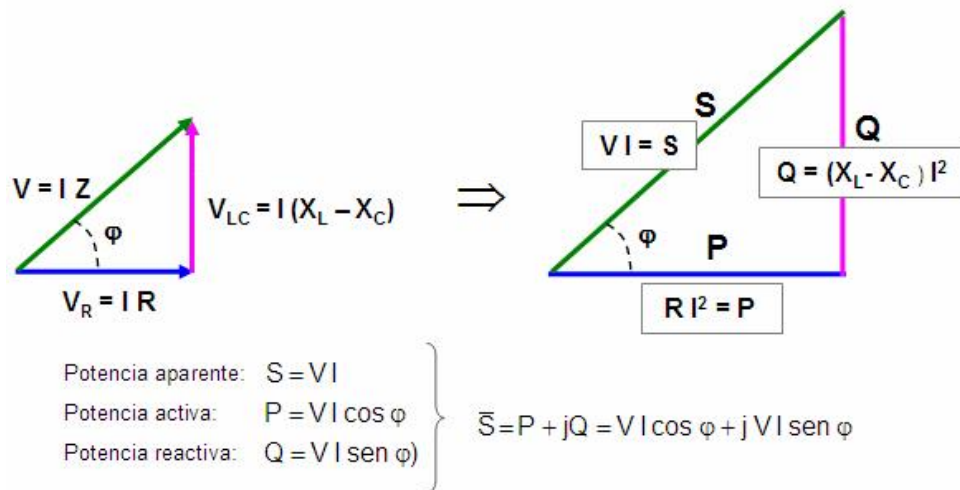


Figura 3.6: Triángulo de potencias

Un método para medir la eficiencia del sistema de distribución eléctrica es el factor de potencia. El factor de potencia se define como la relación entre la potencia activa suministrada a la carga y la potencia aparente suministrada a la instalación. El factor de potencia se expresa con un número entre 0 y 1. Un factor de potencia de 1, significa que el cien por cien de la potencia suministrada a la instalación se utiliza para generar trabajo real. Un factor de potencia de cero significa que la potencia que se suministra a la instalación no está siendo utilizado para generar trabajo real. En realidad, el factor de potencia nunca es 0 ó 1, siempre está en algún punto intermedio. Lo ideal es que el factor de potencia sea 1 o lo más cercano a él.

Hasta hace unos años, antes de la llegada de la era digital, la mayoría de las cargas eran, bien resistivas puras, bien inductivas ya que la mayoría de los elementos existentes se comportaban de esa forma, con lo que, para mejorar el factor de potencia cuando aparecían muchas cargas inductivas, se procedía a añadir condensadores y asunto resuelto. En la era digital todo está computerizado o es programable y está alimentado por fuentes de alimentación conmutadas que utilizan rectificadores y transistores de conmutación para regular la tensión. Estas fuentes y equipos digitales se comportan como cargas capacitivas, por lo tanto, la instalación de condensadores de corrección puede ser contraproducente empeorando el factor de potencia aún más. En la actualidad se están empleando tecnologías de corrección de forma de onda para eliminar tanto el efecto inductivo como el capacitivo de las cargas y así mejorar el factor de potencia.

### 3.2.3. Equipo

Visto el diagrama de bloques de la figura 1 y como se comentó anteriormente, vamos a desgajarlo bloque a bloque para ir explicando cada uno de ellos. La idea inicial es ver lo que hay en el mercado relativo al bloque que vamos a construir y elegir según unos requerimientos que

vamos a marcar. Por un lado habrá unos requerimientos globales a todo el sistema y por otro unos específicos de cada bloque.

NOTA: Parte de este diseño está basado en varios documentos y notas de aplicación [Inc12a], [KQ09], [Inc12b], [Inc10b], [Inc13], [Kin09], [Inc10a] y [Inc12c] proporcionados por el fabricante del producto.

Los requerimientos globales que deberán cumplir los elementos a incluir en el presente trabajo, serán, por orden de importancia, los siguientes:

1. Tamaño y peso.
2. Precio.
3. Prestaciones.
4. Curva de aprendizaje/Facilidad de uso.

Para resumir, queremos algo que sea pequeño y poco pesado, que cueste poco, pero sea seguro y fiable, que nos proporcione un rendimiento alto y que podamos usarlo de forma rápida y eficiente. A primera vista parece una quimera pero, poco a poco, iremos viendo que es posible hacer grandes cosas con recursos reducidos. Una de las premisas a seguir es que cualquiera pueda disponer de un elemento de este tipo en su casa sin realizar grandes desembolsos. La tecnología debe estar al alcance de todo el que quiera usarla sin que su coste sea un impedimento para poder hacerlo.

### 3.2.3.1. Sensores de corriente

Los tipos de sensores más utilizados, en la actualidad, para el sensado de corrientes son los que mostramos a continuación:

1. Sensores de efecto Hall integrados. Ver figura 3.7.
2. Sensores inductivos tipo pinza. Ver figuras 3.8 y 3.9.
3. Shunts resistivos. Ver figuras 3.10 y 3.11.

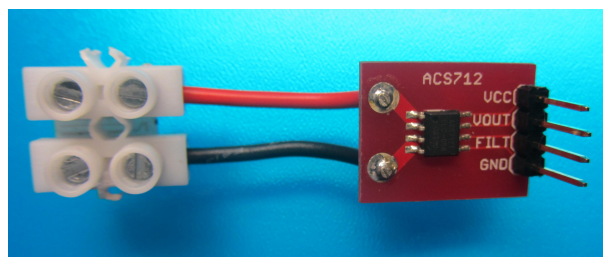


Figura 3.7: Sensor de corriente de efecto Hall





Figura 3.8: Sensor de corriente inductivo tipo pinza CurrentCost



Figura 3.9: Sensor de corriente inductivo tipo pinza

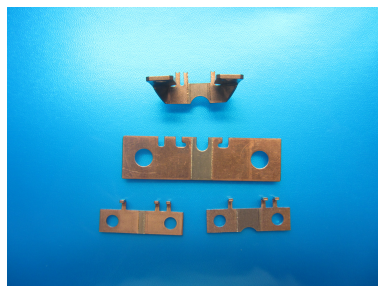


Figura 3.10: Shunt resistivo de potencia

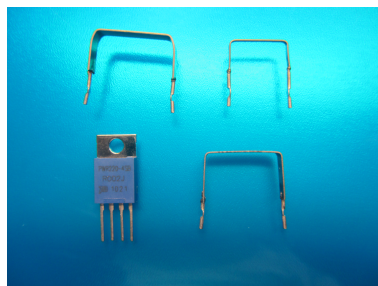


Figura 3.11: Shunt resistivo de baja potencia

Para la elección del sensor de corriente se han tenido en cuenta los siguientes factores a la hora de tomar la decisión:

- Tamaño y peso
- Precio
- Rango de medida
- Facilidad de uso

Después de analizar la información obtenida de fabricantes y suministradores, en la tabla 3.1, podemos ver un resumen de las valoraciones de cada uno de los tipos.

Tipo	Tamaño/Peso	Precio	Rango	Uso
Efecto Hall	pequeño	medio	pequeño	difícil
Pinza	grande	alto	medio	medio
Shunt resistivo	mediano/pequeño	bajo	grande	fácil

Cuadro 3.1: Comparativa de tipos de sensores de corriente.

Como podemos ver en el esquema de la figura 3.12, el sensor de corriente elegido finalmente consiste en una resistencia tipo shunt que va en serie con la fase de la red. En los terminales marcados como FASE IN y NEUTRO IN se conectan a la red eléctrica y entre FASE OUT Y NEUTRO OUT se conecta la carga a medir. La corriente que precisa la carga pasa a través del shunt y genera una pequeña tensión proporcional a la corriente que la atraviesa que se introduce en el Energy meter para su medición.

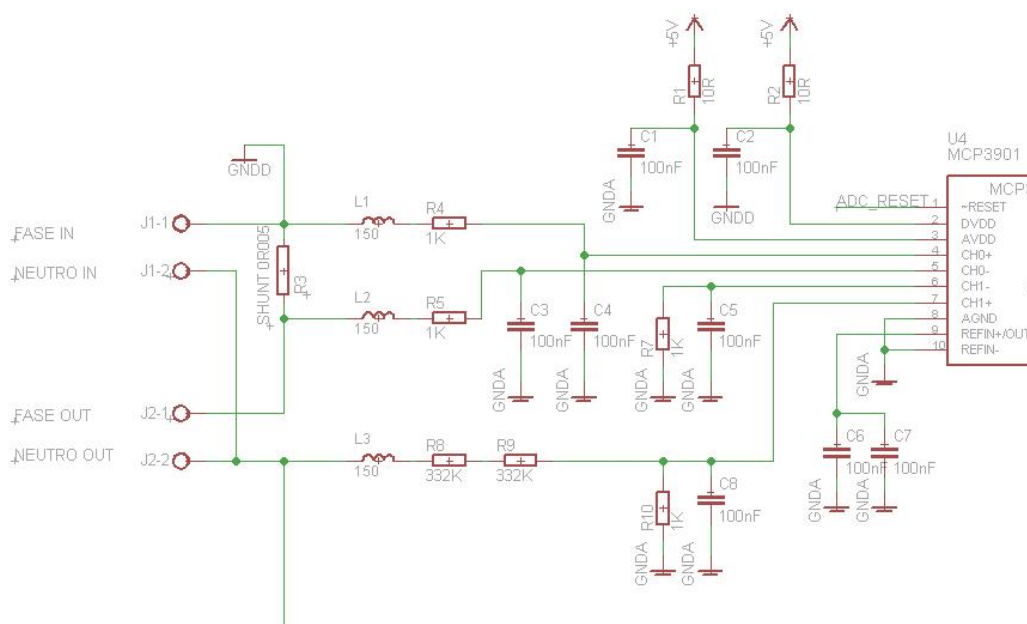


Figura 3.12: Circuito de entrada del sistema.

Ahora pasamos a realizar el cálculo del valor de la resistencia shunt, para lo cual debemos tener en cuenta los valores de  $I_{MAX}$  e  $I_{MIN}$  a medir y el rango de entrada del MCP3901 que es de  $980mV_{pp}$  aunque se puede modificar mediante el PGA (amplificador de ganancia programable) interno de cada canal.

Para realizar los cálculos he creado una hoja de cálculo con distintos valores de tensión, corriente y resistencia shunt. Como mi intención es medir desde 10 mA hasta 30 A aproximadamente, entonces de la hoja de cálculo se obtiene el valor del shunt que más nos interese, en este caso el de  $5m\Omega$ . Ver figura 3.13.

	A	B	C	D	E	F	G	H	I	J	K	L
1				R	R	R	R	R	R	R	R	
2	Potencia (230V) [Wef]	Corriente [ef [A]	Corriente [pp [A]	0,1	0,05	0,025	0,02	0,01	0,005	0,002	0,001	
3	2,30	0,010	0,028	0,0028	0,0014	0,0007	0,0006	0,0003	0,0001	0,0001	0,0000	Vpp
4	11,50	0,050	0,141	0,0141	0,0071	0,0035	0,0028	0,0014	0,0007	0,0003	0,0001	Vpp
5	23,00	0,100	0,283	0,0283	0,0141	0,0071	0,0057	0,0028	0,0014	0,0006	0,0003	Vpp
6	115,00	0,500	1,414	0,1414	0,0707	0,0354	0,0283	0,0141	0,0071	0,0028	0,0014	Vpp
7	230,00	1,000	2,828	0,2828	0,1414	0,0707	0,0566	0,0283	0,0141	0,0057	0,0028	Vpp
8	1150,00	5,000	14,142	1,4142	0,7071	0,3536	0,2828	0,1414	0,0707	0,0283	0,0141	Vpp
9	2300,00	10,000	28,284	2,8284	1,4142	0,7071	0,5657	0,2828	0,1414	0,0566	0,0283	Vpp
10	3450,00	15,000	42,426	4,2426	2,1213	1,0607	0,8485	0,4243	0,2121	0,0849	0,0424	Vpp
11	4600,00	20,000	56,569	5,6569	2,8284	1,4142	1,1314	0,5657	0,2828	0,1131	0,0566	Vpp
12	5750,00	25,000	70,711	7,0711	3,5355	1,7678	1,4142	0,7071	0,3536	0,1414	0,0707	Vpp
13	5980,00	26,000	73,539	7,3539	3,6770	1,8385	1,4708	0,7354	0,3677	0,1471	0,0735	Vpp
14	6210,00	27,000	76,368	7,6368	3,8184	1,9092	1,5274	0,7637	0,3818	0,1527	0,0764	Vpp
15	6440,00	28,000	79,196	7,9196	3,9598	1,9799	1,5839	0,7920	0,3960	0,1584	0,0792	Vpp
16	6670,00	29,000	82,024	8,2024	4,1012	2,0506	1,6405	0,8202	0,4101	0,1640	0,0820	Vpp
17	6900,00	30,000	84,853	8,4853	4,2426	2,1213	1,6971	0,8485	0,4243	0,1697	0,0849	Vpp
18	8050,00	35,000	98,995	9,8995	4,9497	2,4749	1,9799	0,9899	0,4950	0,1980	0,0990	Vpp
19	9200,00	40,000	113,137	11,3137	5,6569	2,8284	2,2627	1,1314	0,5657	0,2263	0,1131	Vpp
20	10350,00	45,000	127,279	12,7279	6,3640	3,1820	2,5456	1,2728	0,6364	0,2546	0,1273	Vpp
21	11500,00	50,000	141,421	14,1421	7,0711	3,5355	2,8284	1,4142	0,7071	0,2828	0,1414	Vpp
22												
23												
24	Vef [v]	230										
25	Vpp [v]	651										
26	Vef [v] (comprobación)	230										

Figura 3.13: Cálculo de la resistencia shunt de sensado de corriente.

### 3.2.3.2. Sensores de tensión

El objetivo es reducir la tensión a un nivel que pueda ser leído por el Energy Meter sin dañarlo pero debemos mantener las características de la señal para no falsear la muestra. Para realizar esta tarea podemos hacerlo usando dos tipos de sensores:

1. Sensores de tipo inductivo o transformadores.
2. Divisor de tensión resistivo.

Sensores de tipo inductivos o transformadores. Su principal ventaja es el aislamiento galvánico entre la tensión a medir y la medida, pero tiene dos inconvenientes de mucho peso, y nunca mejor dicho: el primero es el peso y volumen del elemento, el segundo son los valores máximos

de salida que se requieren, que al ser muy bajos requerirían a su vez de un divisor resistivo con lo que añadiríamos nuevos elementos que si se eligiese desde el principio el divisor resistivo, y el tercero es el precio ya que los transformadores no son baratos.

El divisor de tensión. Está formado por dos resistencias en serie. Tiene las ventajas del poco peso, la sencillez y el precio bajo pero hay que tener en cuenta que el calentamiento por el efecto Joule hace que su valor varíe y por tanto puede falsear la información. Está claro que sus ventajas superan a sus inconvenientes que podremos paliar con resistencias más inmunes a la temperatura, eso sí, un poco más caras. Por todo lo anterior elegimos este tipo como mejor solución.

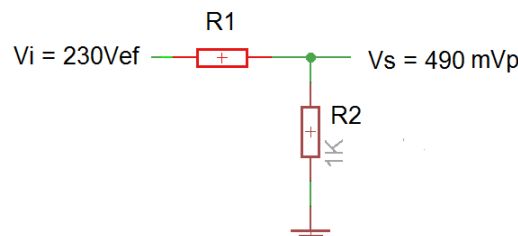


Figura 3.14: Divisor de tensión resistivo.

Para realizar el cálculo del divisor resistivo, figura 3.14, deberemos tener en cuenta la tensión de entrada de la red  $230V_{EF}$  y la tensión de entrada del MCP3901 que es de  $490mV_p$ . Para convertir la tensión eficaz de la red en tensión de pico deberemos multiplicarla por  $\sqrt{2}$  y para obtener la relación entre la entrada y la salida, ecuación 3.1, dividiremos la tensión de la red ( $V_i$ ) en valor pico por la tensión de entrada al MCP3901 ( $V_s$ ):

$$Rel = \frac{230\sqrt{2}}{0,490} = 664 \quad (3.1)$$

Como  $R_2 = 1k\Omega$  entonces  $R_1 = 664k\Omega$  y para que la potencia disipada en la  $R_1$  se reparta uso dos resistencias en serie de  $332k\Omega$ .

Y el circuito final queda como se puede ver en la figura 3.15, siendo  $R_8$  y  $R_9$  los equivalentes a la  $R_1$  de nuestro cálculo.

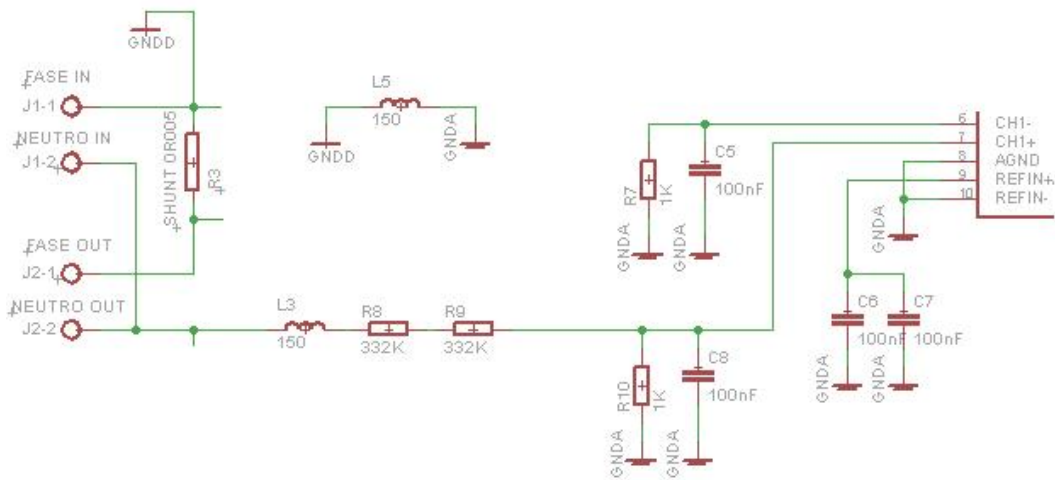


Figura 3.15: Circuito de sensado de tensión.

### 3.2.3.3. Energy Meter

Después de intentar realizar la captura de la tensión y corriente de varias formas distintas e independientes sin éxito, descubrí la familia MCP39XX, ver figura 3.16 de Microchip Technology Inc. que resuelve la mayoría de los problemas que se planteaban. Los miembros de esta familia disponen, al menos, de dos canales independientes, síncronos y diferenciales de captura de señales analógicas del tipo Delta-Sigma con muy baja distorsión y alta inmunidad al ruido.

**MICROCHIP**

Energy Measurement Products

Reset All Filters Show/Hide Columns Download

To SORT a column click the column header. Use CTRL key to select multiple values.

Product	Volume Pricing	Measurement Error	Dynamic Range	Gain Selection	Voltage Reference Drift ppm typ	PSRR % typ	I <sub>ldd</sub> (mA) Max.	D <sub>idd</sub> (mA) Max.	AV <sub>dd</sub> (V) Min.	AV <sub>dd</sub> (V) Max.	DV <sub>dd</sub> (V) Min.	DV <sub>dd</sub> (V) Max.	Packages
MCP3901	\$1.44	91 dB SINAD	24-bit resolution	1:32	15	-77 dB	2.1	0.8	4.5	5.5	2.7	5.5	20/QFN 20/SSOP 208mil
MCP3903	\$3.09	91 dB SINAD	24-bit resolution	1:32	12	-68 dB	9	1.7	4.5	5.5	2.7	3.6	28/SSOP 208mil
MCP3905A	\$1.34	0.1%	500:1	1:16	15	0.01%	3	2	4.5	5.5	4.5	5.5	24/SSOP 208mil
MCP3905L	\$1.36	0.1%	500:1	1:16	15	0.01%	3	2	4.5	5.5	4.5	5.5	24/SSOP 208mil
MCP3906A	\$1.47	0.1%	1000:1	1:32	15	0.01%	3	2	4.5	5.5	4.5	5.5	24/SSOP 208mil
MCP3909	\$1.48	0.1%	1000:1	1:16	15	0.01%	2.8	1.2	4.5	5.5	4.5	5.5	24/SSOP 208mil
MCP3911	\$1.40	94.5 SINAD	24-bit resolution	1:32	7	-73 dB	1.7	0.28	2.7	3.6	2.7	3.6	20/QFN 20/SSOP 208mil

Volume pricing is for budgetary use only, shown in United States dollars. The prices are representative and do not reflect final pricing. Contact your local Microchip sales representative or distributor for volume and/or discount pricing.

Products Applications Design Support Training Sample & Buy About Us Contact Us

©Copyright 1998-2012 Microchip Technology Inc. All rights reserved. Shanghai ICP Recordal No.09049794

Figura 3.16: Familia MCP39xx.

Como podemos ver en la figura 3.16, su precio es muy bajo y las prestaciones, como hemos mencionado anteriormente son muy altas, incluyendo casi toda la funcionalidad requerida en un sólo chip.

Como nuestro sistema es monofásico nos decidimos por el MCP3901 que además dispone de un puerto SPI para poder conectarlo al microcontrolador desde el que se puede gestionar toda la información recibida y transmitirla a otros dispositivos, en nuestro caso un dispositivo Android.

El único inconveniente detectado es que trabajar con él no es sencillo con lo que la curva de aprendizaje es larga.

#### 3.2.3.4. Microcontrolador

En la actualidad hay cientos de microcontroladores distintos en el mercado con lo que hacer un análisis de todos ellos iba a ser demasiado complicado, por lo tanto, marcamos una serie de requisitos, que detallo a continuación y buscamos dentro de la familia de los PIC desarrollada por Microchip Technology Inc. que ya conocemos así se reduce casi a cero el periodo de aprendizaje.

Los requisitos que nos marcamos fueron:

- capacidad de proceso
- memoria de datos y de programa
- Precio
- Tamaño
- Consumo
- periodo de aprendizaje

Necesitamos un microcontrolador con una capacidad de proceso importante ya que tiene que realizar una gran cantidad de operaciones con los datos obtenidos del energy meter para enviarnos a nosotros la información que precisamos. También necesitamos gran cantidad de memoria de programa y de datos para tratar toda la información indicada anteriormente y, para finalizar, queremos que sea barato, de reducidas dimensiones y de bajo consumo.

Product	Volume Pricing	Program Memory KBytes	Self-write	EEPROM Data Memory	RAM	I/O Pins	Pin count	Max. CPU Speed MHz	CPU Speed MIPS	Internal Oscillator	# of A/D Ch.	Cap Touch Channels	Digital Communication
PIC18F1220	\$1.16	4	No	0	1536	23	18	32	8	16 MHz	2	8	
PIC18F1230	\$1.20	8	No	0	2048	24	20	40	10	16 MHz, 32 kHz	4	9	
PIC18F1320	\$1.23	16	Yes	128	3328	25	20	40	10	16 MHz, 32 kHz	6	10	
PIC18F1330	\$1.26	24	Yes	256	3648	32	23	42	10.5	32 kHz	8	10	
PIC18F13K22	\$1.27	32	Yes	1024	3800	34	40	48	12	8 MHz, 32 kHz	7	11	
PIC18F13K50	\$1.30	48	Yes	1024	3808	35	44	64	16		8	12	
PIC18F14K22	\$1.32	64	Yes	1024	3828	36	64	64			9	13	
PIC18F14K50	\$1.34	96	Yes	1024	3882	39	90	90			10	14	
PIC18F2220	\$1.37	128	Yes	1024	3896	49	100	100			11	16	
PIC18F26K20	\$1.65	64	Yes	1024	3936	25	28	64	16	16 MHz, 32 kHz	10		0-UART 1-A/E/USART 0-SPI 0-I2C 1-MSSP(SPI/I2C)
PIC18F26K22	\$1.90	64	Yes	1024	3896	25	28	64	16	16 MHz, 32 kHz	17	17	0-UART 2-A/E/USART 2-MSSP(SPI/I2C)
PIC18F26K80	\$2.18	64	Yes	1024	3648	24	28	64	16	16 MHz, 32 kHz	8	8	2-A/E/USART 0-SPI 1-MSSP(SPI/I2C)

Volume pricing is for budgetary use only, shown in United States dollars. The prices are representative and do not reflect final pricing. Contact your local Microchip sales representative or distributor for volume and / or discount

Figura 3.17: Comparativa microcontroladores PIC.

Partiendo de que sea barato tenemos que irnos a las familias de 8 bits pero al requerir altas prestaciones nos iremos a la más potente de 8 bits, los PIC18, como queremos que ocupe poco nos iremos a los de 28 pines y tipo LF porque es de bajo consumo. Resumiendo necesitamos un PIC18LF2xxx y como queremos uno de altas prestaciones, y después de darle varias vueltas a las hojas de características de los micros, hemos optado por el PIC18LF26K80.

Ver apéndice A para la descripción de las características del micro.

### 3.2.3.5. Memoria EEPROM

La función de la memoria EEPROM es principalmente almacenar los parámetros de calibración y de energía para poder totalizar el consumo energético. Como no tenemos grandes requerimientos optamos por una memoria con un encapsulado pequeño y barata. Como la mayoría de los componentes están desarrollados por Microchip Technology nc. pues nos decidimos por adquirir una memoria del mismo fabricante, en este caso la 25LC256, de 256 kbits.

### 3.2.3.6. Pulsadores y leds

El sistema dispone de tres pulsadores y dos led para interactuar con el usuario.

El pulsador S1 sirve para entrar en modo calibración colocando una carga resistiva pura que genere una corriente de 5A, durante el proceso el LED1 parpadeará y una vez haya terminado el proceso de calibración permanecerá encendido fijo durante 5 segundos.

El pulsador S2 es para poner todos los contadores a cero. Si el proceso se ha realizado de forma satisfactoria se encenderá el LED2 durante 5 segundos, si se detecta algún problema parpadeará.

El pulsador S3 es para el reseteo del microcontrolador.

El LED1 además de la función indicada anteriormente, en el modo de funcionamiento normal generará un destello cada vez que se alcance un valor de energía activa consumida que se puede configurar.

El LED2, al igual que el LED1 también, en su funcionamiento normal, genera un destello al alcanzar un valor de energía reactiva consumida, también configurable.

### 3.2.3.7. Aislador

Como tenemos una masa “caliente” (hot or live ground), es decir que la fase de la red eléctrica está conectada a la masa de nuestro circuito, no podemos conectar al mismo ningún otro dispositivo ya que la masa de nuestro circuito al juntarse con la masa del otro dispositivo daría lugar al correspondiente cortocircuito. Por esta razón si queremos conectar cualquier dispositivo por cable a nuestro equipo hay que usar un aislador para que nos aisle las masas, el único inconveniente es que el equipo externo a conectar debe proporcionar una tensión de 3,3V para alimentar la etapa de salida del aislador.

El aislador que hemos empleado nosotros es un ADUM1201 desarrollado por Analog Devices, Inc. de tipo inductivo que tiene mejor respuesta que los de tipo optoacoplado. Es de doble canal para usarlo con las señales TX y RX de la comunicación serie y dispone de alimentación independiente en cada uno de los lados, como ya hemos mencionado antes, para garantizar el aislamiento.



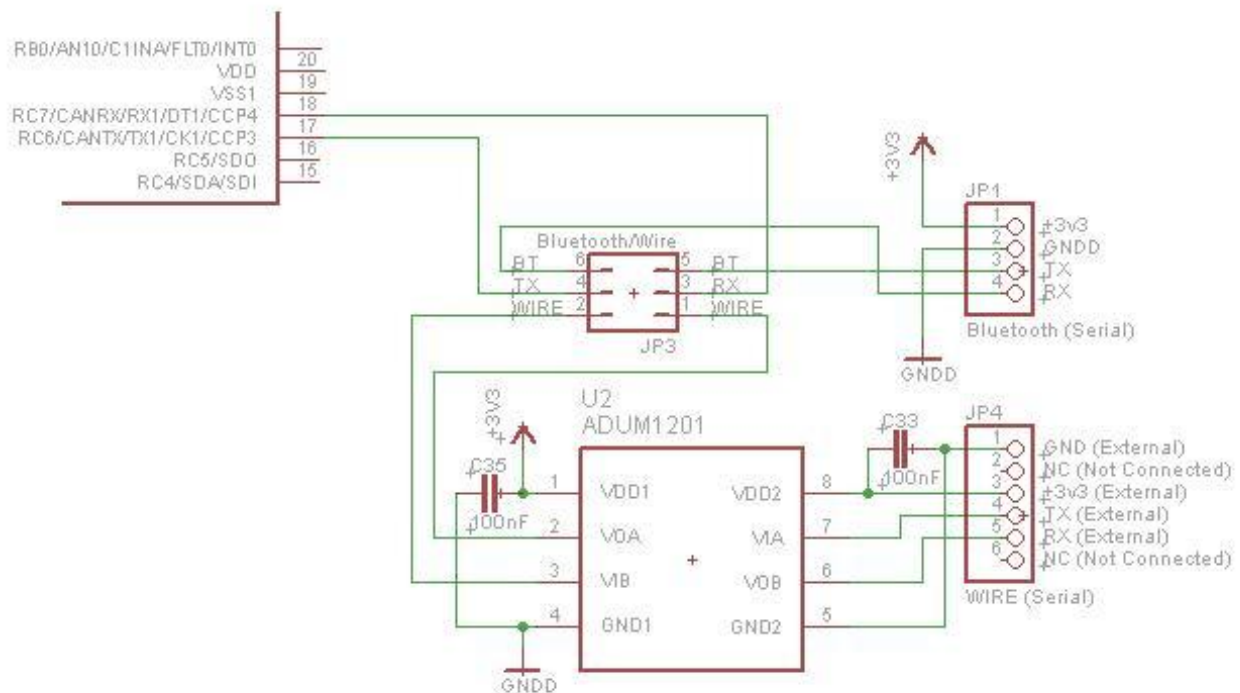


Figura 3.18: Comunicación serie.

### 3.2.3.8. Bluetooth

Para solventar el problema de la masa “caliente” indicada en el apartado ?? y para poder conectarnos fácilmente desde un dispositivo Android se ha optado por disponer de una conexión Bluetooth con el exterior. Dicha conexión se realiza mediante un módulo Bluetooth básico y de uso muy extendido, ver apéndice B, por su facilidad de manejo. Sólo es necesario conectar los puertos de TX, RX y la alimentación correspondiente y ya sólo falta emparejar el otro dispositivo para realizar el enlace.

Este tipo de módulos es muy fácil de conseguir en Ebay por alrededor de 5 €, es de tipo esclavo con un alcance de 10 m., más que suficiente para lo que queremos y con unas dimensiones de 27 x13x 2mm y 1 g. de peso. El inconveniente visto es el consumo que ronda los 45mA y nos ha hecho tener que redimensionar la fuente de alimentación original.

### 3.2.3.9. Fuente de alimentación

Para realizar la fuente de alimentación y debido a requerimientos de espacio y peso, se ha optado por una sin transformador de tipo capacitiva [Con04] ya que la resistiva requiere una resistencia de alta potencia que disipa mucho calor y que es voluminosa.

Una fuente de alimentación sin transformador de tipo capacitiva mantiene constante la tensión de salida  $V_{OUT}$  cuando la corriente de salida  $I_{OUT}$  sea menor o igual que la corriente de entrada  $I_{IN}$ , siendo R1 quien limita la corriente de entrada  $I_{IN}$ . Dicha R1 se elige para que no disipe demasiada potencia.

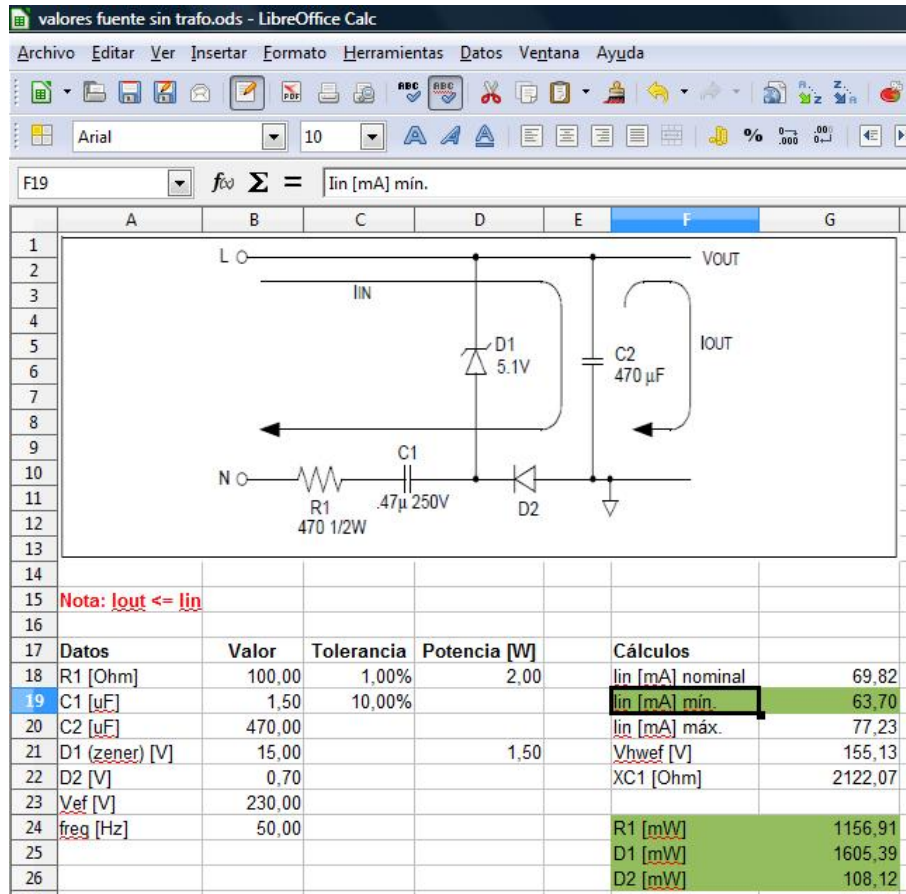


Figura 3.19: Cálculo de los valores de la fuente de alimentación.

Para el cálculo de la fuente de alimentación se han empleado las ecuaciones: 3.2, 3.3, 3.4 y 3.5 trasladadas a una hoja de cálculo que podemos ver en la figura 3.19.

$$I_{IN} = \frac{V_{HFRMS}}{X_{C1} + R1} \geq I_{OUT} \quad (3.2)$$

Donde  $V_{HFRMS}$  es el valor eficaz de media onda de una señal alterna senoidal, que calculamos en la ecuación 3.3.

$$V_{HFRMS} = \frac{V_{PICO} - V_Z}{2} = \frac{\sqrt{2}V_{RMS} - V_Z}{2} \quad (3.3)$$

Donde  $V_{PICO}$  es la tensión pico de la red eléctrica y  $V_Z$  la tensión del zener.

Ahora calcularemos la  $X_{C1}$ .

$$X_{C1} = \frac{1}{2\pi f C1} \quad (3.4)$$

Resumiendo:

$$I_{IN} = \frac{\sqrt{2}V_{RMS} - V_Z}{2(X_{C1} + R1)} \quad (3.5)$$

NOTA: Según el cálculo teórico, que se ve en la figura 3.19, el valor mínimo de corriente de entrada que se puede obtener es de 63,70 mA, sin embargo empíricamente la máxima corriente obtenida han sido 41 mA.

En la figura 3.20 podemos ver el resultado final.

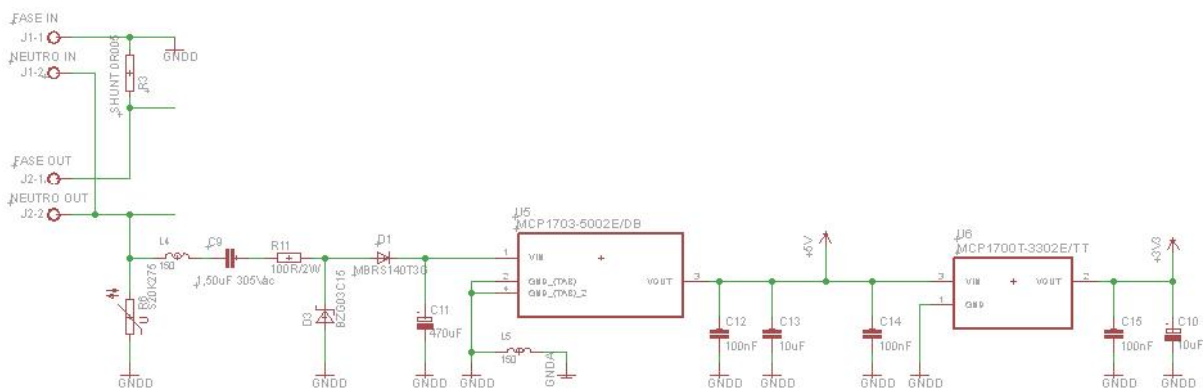


Figura 3.20: Fuente de alimentación sin transformador.

### 3.2.4. Firmware

El firmware es el software que corre en el microcontrolador y se encarga de todas las tareas necesarias para el correcto funcionamiento del sistema.

En el diagrama de la figura 3.21 se muestran todas las tareas que desarrolla el firmware, dicho software está escrito en C con alguna inclusión de ensamblador para obtener un mejor rendimiento del micro.

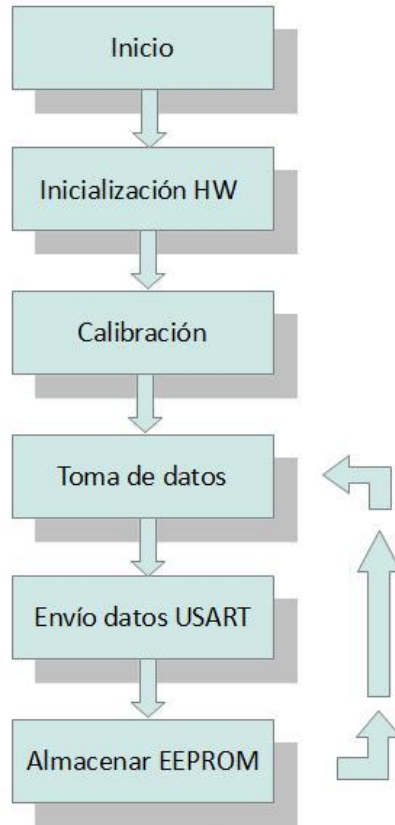


Figura 3.21: Flujo de ejecución del programa del PIC.

En el listado siguiente aparece la función principal del programa que desarrolla el organigrama de la figura 3.21.

```

/*****

```

```

File Name:      Main_EnergyMeter.h
Dependencies:   Microchip Peripheral Library
Processor:     PIC18F26K80
Company:       Microchip Technology, Inc.

```

```

Copyright 2010 Microchip Technology Inc. All rights reserved.

```

```

Microchip licenses to you the right to use, modify, copy and
distribute Software only when embedded on a Microchip microcontroller
or digital signal controller that is integrated into your product or
third party product (pursuant to the sublicense terms in the
accompanying license agreement).

```

```

You should refer to the license agreement accompanying this Software
for additional information regarding your rights and obligations.

```

*SOFTWARE AND DOCUMENTATION ARE PROVIDED ?AS IS? WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY, TITLE, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL MICROCHIP OR ITS LICENSORS BE LIABLE OR OBLIGATED UNDER CONTRACT, NEGLIGENCE, STRICT LIABILITY, CONTRIBUTION, BREACH OF WARRANTY, OR OTHER LEGAL EQUITABLE THEORY ANY DIRECT OR INDIRECT DAMAGES OR EXPENSES INCLUDING BUT NOT LIMITED TO ANY INCIDENTAL, SPECIAL, INDIRECT, PUNITIVE OR CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA, COST OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR ANY CLAIMS BY THIRD PARTIES (INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF), OR OTHER SIMILAR COSTS.*

<i>Author</i>	<i>Date</i>	<i>Revision History / Comments</i>
---------------	-------------	------------------------------------

<i>Harsha Jagadish</i>	<i>01 – Jan – 2011</i>	<i>Initial Version v1.0</i>
------------------------	------------------------	-----------------------------

<i>Craig King</i>	<i>15 – Feb – 2011</i>	<i>Port to 65J90 v1.1</i>
-------------------	------------------------	---------------------------

<i>Jesus Urcera</i>	<i>15 – Sep – 2012</i>	<i>Port to PIC18LF26K80</i>
---------------------	------------------------	-----------------------------

*\*\*\*\*\* \*/*

**#include** "RegDef\_EnergyMeter.h"

**#include** "HardwareProfile\_EnergyMeter.h"

**#include** "MeteringEngine\_EnergyMeter.h"

**#include** "Config\_EnergyMeter.h"

*//NOTE: It is very important to use linker file associated with //project. This linker is configured to use zero initialization //of all global variables which otherwise may affect the functionality*

*/\* \*\*\*\*\* DEVICE CONFIGURATION PARAMETERS \*\*\*\*\* \*/*

*\* Extended Instruction mode is disabled*

*\* Reset on Stack overflow is disabled*

*\* Watch Dog Timer Enabled*

*\* Watch Dog timer set for 2s time out*

*\* Program memory is not Code Protected*

*\* Int. to External switch over in oscillator disabled*

*\* Fail Safe monitor is enabled*

*\* Oscillator is configured in High Speed HS mode*

*\* Hardware Debugger disabled*

```

*          CCP2 output is configured on defalut pin
*****/
#pragma config  XINST=OFF, STVREN=OFF, WDTEEN=OFF, WDTPS = 512
#pragma config  CP0=OFF
#pragma config  IESO=OFF, FCMEN=ON, FOSC=INTIO2

//NOTE: Excercise caution before the addition/deletion of variables or
//relocating them as few are used in assembly which are bank specific
//NOTE: Donot add or delete any register from below upto comment

//***** GLOBAL ENERGY COMPUTATION Variable Definitions *****

/***** GUI UPDATABLE REGISTERS *****/
#pragma udata          CALIBRATION_BANK=0x200
MODE1_REG              MODE1;
STATUS1_REG           STATUS1;
CALB_CONTROL          CAL_CONTROL;
UINT16                LINE_CYC;
UINT16                LINE_CYC_CNT;
QWORD_VAL             RAW2_I_RMS;
UINT16                RAW_I_RMS;
UINT16                I_RMS;
QWORD_VAL             RAW2_V_RMS;
UINT16                RAW_V_RMS;
UINT16                V_RMS;
UINT16                FREQUENCY;
UINT16                PERIOD;
UINT32                RAW_POWER_ACT;
QWORD_VAL             POWER_ACT;
UINT32                POWER_APP;
QWORD_VAL             RAW_POWER_REACT;
UINT32                POWER_REACT;
QWORD_VAL             RAW_ENERGY_ACT;
QWORD_VAL             RAW_ENERGY_APP;
QWORD_VAL             RAW_ENERGY_REACT;
UINT8                I_ABS_MAX;
UINT8                V_ABS_MAX;

//***** CALIBRATION REGISTERS *****/
INT8                  PHASE_COMPENSATION;
INT8                  PHASE_COMPENSATION_90;

```

```

INT16          OFFSET_I_RMS ;
UINT16         GAIN_I_RMS ;
INT16          OFFSET_V_RMS ;
UINT16         GAIN_V_RMS ;
INT32          OFFSET_POWER_ACT ;
UINT16         GAIN_POWER_ACT ;
UINT16         GAIN_ENERGY_ACT ;
UINT16         GAIN_NUMR_ENERGY_ACT ;
UINT8          GAIN_DENR_ENERGY_ACT ;
INT32          OFFSET_POWER_REACT ;
UINT16         GAIN_POWER_REACT ;
UINT16         GAIN_ENERGY_REACT ;
UINT16         GAIN_NUMR_ENERGY_REACT ;
UINT8          GAIN_DENR_ENERGY_REACT ;
UINT16         GAIN_ENERGY_APP ;
UINT8          CREEP_THRESHOLD_MINUTE ;
UINT8          CREEP_THRESHOLD_SECOND ;
UINT16         METER_CONSTANT ;
UINT8          CF_PULSE_WIDTH ;
UINT16         MODEL_DEF ;
UINT16         CAL_STATUS ;
UINT16         MAXIMUM_CURRENT ;
UINT16         CALIBRATION_VOLTAGE ;
UINT16         CALIBRATION_CURRENT ;
UINT16         CALIBRATION_FREQUENCY ;
UINT16         CALIBRATION_LINE_CYCLE ;
#pragma udata  ENG_REGS=0x300
UINT32         ENERGY_ACT ;
UINT32         ENERGY_REACT ;
UINT32         ENERGY_APP ;

```

```

/*****

```

```

Function      : void main(void)

```

```

Description : The function calls various functions for complete
calculation of energy.

```

```

The flow of execution follows:

```

- \* *Energymeter hardware initialization*
- \* *Calibration and Firmware register initialization*
- \* *initiate begining of Cycle function to calculate
and accumulate energy for each cycle of input*

- \* *Update the GUI through USART*
- \* *Save energy periodically to EEPROM*

*Arguments : None*

*Remarks : None*

```

*****/
/* ***** CONVENTIONS IN NAMING *****
* COMPUTATION CYCLE – Completion of User specified cycles before
* accumulation of energy
* LINE CYCLE – Collection of every 128 samples across one input
* cycle of voltage & current
*****/

```

**void main(void)**

```

{
    BoardInit(); //Inicializa la placa
    //Store energy in case of WDT reset
    Store_EN_To_EEP_On_WDT_Reset();
    //NOTE: Donot clear WDT prior to this statement which
    //otherwise will clear WDT reset indication in RCONbits.TO
    ClearWDT();
    Init_Variables(); //Inicializa variables a sus valores
    ClearWDT();
    Init_Delay(); //Added delay to stabilize MCP3901 voltage
    ClearWDT();
    // This function will initialize all the variables to default
    // values and initialize all peripherals
    Init_EnergyMeter();
    puts1USART("Init_EnergyMeter() _OK!\r\n"); //Only DEBUG

//***** Main Functionality *****
//* Begin initialization fo all accumulation registers to zero in
//* Every_COMPUTATION_CYC_Start()
//* At Every sine cycle , get an intermediatery update After
//* 128 sine cycles (Computation Cycle) calculate all parameter
//* Repeat the above process
//*****
    while(1)
    {
        // This function Every_COMPUTATION_CYC_Start() will initialize
        // all the variables which are used in process of acquisition

```



```

// of "N" no of samples and in calculation.
Computation_Cycle_Start ();
do
{
    Signal_Cycle_Begin ();

    Process_GUI_Command_Loop ();

    if ( (! IsCalibrationModeEnabled ()) || \
        UpdateParametersForCalibration () )
    {
        ComputeSignalFrequency ();
        ProcessInstantaneousActivePower ();
        ProcessInstantaneousReactivePower ();
        ComputeActiveEnergy ();
        ComputeReactiveEnergy ();
        ComputeActivePower ();
        ComputeReactivePower ();
        ComputeRMSVoltage ();
        ComputeRMSCurrent ();
        ComputeApparentPower ();
        Check_SD_ADC_Config ();
        SaveEnergyEvery24Hours ();
        Signal_Cycle_Complete ();
        LINE_CYC_CNT++;
        if ( IsComputationCycleCompleted () )
        {
            break ;
        }
    }
    else
    {
        break ;
    }
} while ( LineCyclesRem -- );
// Loop till the "N" cycles are completed before
// initiating new cycle.
// Clears Calibration update cycle
Computation_Cycle_Complete ();

```

```

}
```

```

}
```

### 3.2.5. Diagramas de bloques

El equipo está formado por los bloques que aparecen en la figura 3.22. Aparte está el software que corre en el dispositivo Android.

- Sensores de corriente, tensión y conversión AD (Energy Meter).
- Microcontrolador y almacenamiento de configuración y calibración.
- Comunicación serie (Aislador, Salida serie y Bluetooth).
- Alimentación.

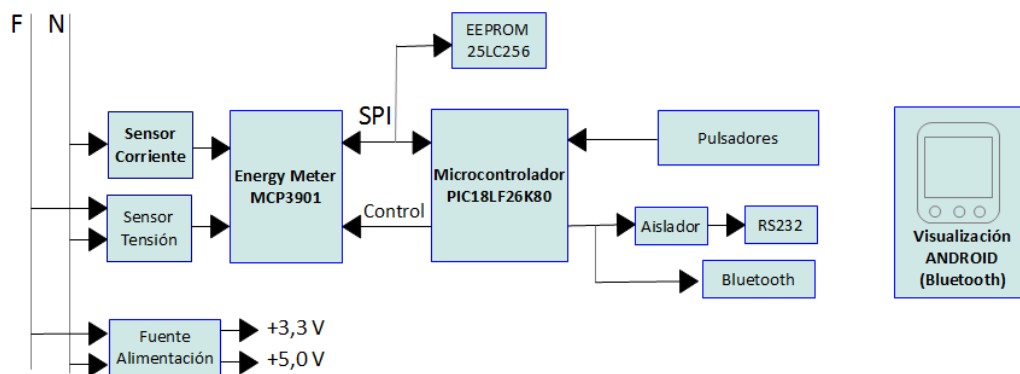


Figura 3.22: Diagrama de bloques

### 3.2.6. Esquemas Electrónicos

El esquema electrónico se puede ver en el anexo C.

### 3.2.7. Lista de materiales

En la tabla 3.2 podemos ver el listado de todos los elementos empleados en la construcción del equipo. En la columna de Dispositivo aparece el nombre con el que está definido el componente dentro de paquete de diseño electrónico Eagle PCB Software.

### 3.2.8. Diagrama de conexionado del microcontrolador

En este diagrama podemos ver que puertos del microcontrolador se están usando y que hay conectado a cada uno de ellos.

<b>Dispositivo</b>	<b>Valor)</b>	<b>Identificador</b>
10-XX	Pulsador rojo/negro	S1, S2
10-XX	Reset	S3
LEDCHIP-LED0603	APT1608EC	LED1, LED2
DIODE-SMB	MBRS140T3G	D1
BZG03C15	BZG03C15	D3
R-EU-R0603	10R	R1, R2
RCA4070	SHUNT 0R005	R3
R-EU-R0805	1K	R4, R5, R7, R10
S20K275	S20K275	R6
R-EU-R1206	332K	R8, R9
R-EU-R2512	100R/2W	R11
R-EU-R0603	4K7	R12, R13, R15
R-EU-R0603	1K	R14, R16, R19
R-EU-R0603	10K	R17
R-EU-R0603	1k2	R18, R21
C-EUC0603	100nF	C1, C2, C5, C6, C7, C8, C12, C14, C15, C25, C29, C30, C31, C33, C35
C-EUC1206	100nF	C3, C4
C-EU225-087X268	1,5uF 305Vac	C9
CPOL-EUB	10uF	C10
CPOL-EUG	470uF	C11
C-EUC0603	10uF	C13
CPOL-EUA/3216-18R	10uF	C101, C103
C-EUC0603	100nF	C102
25LC256-I/SN	25LC256-I/SN	U3
LI0805H151R-10	150 ohm	L1, L2, L3
LI1806C151R-10	150 ohm	L4, L5
PIC18F25K80-I/SS	PIC18LF26K80-I/SS	U1
ADUM1201	ADUM1201	U2
MCP3901	MCP3901	U4
MCP1703-5002E/DB	MCP1703-5002E/DB	U5
MCP1700T-3302E/TT	MCP1700T-3302E/TT	U6
PINHD-1X4	Bluetooth (Serial)	JP1
PINHD-1X5	ICSP	JP2
JP3Q	Bluetooth/Wire	JP3
PINHD-1X6	WIRE (Serial)	JP4
MKDSN1,5/2-5,08	MKDS 3/2-5,08	J1, J2
Varios		Caja (CNMB/2V/KIT), PCB, módulo Bluetooth

Cuadro 3.2: Listado de componentes y materiales electrónicos.

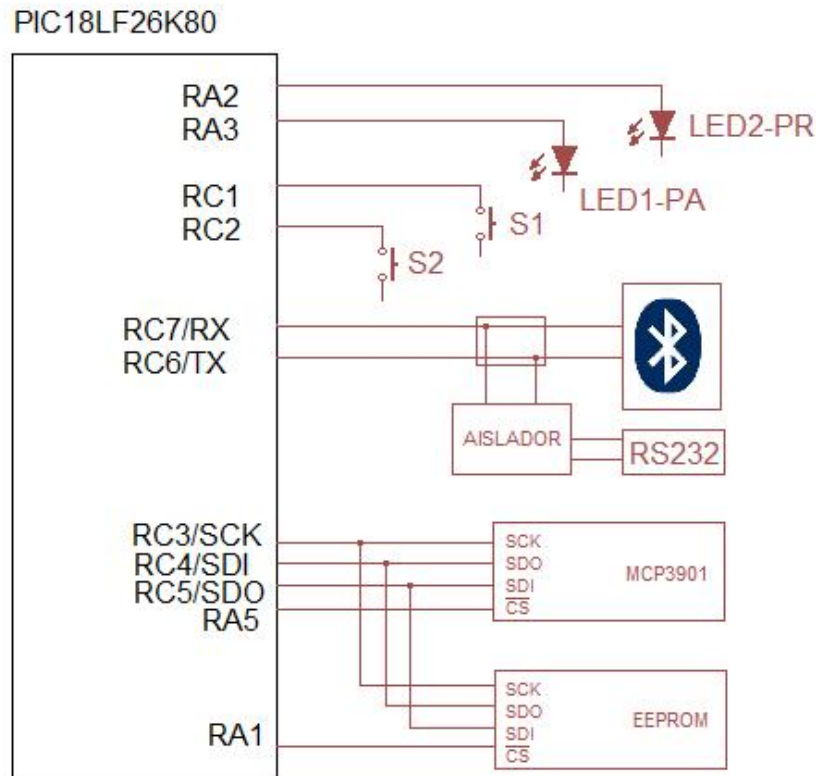


Figura 3.23: Conexión del microcontrolador.

### 3.2.9. Manual de usuario

En este apartado vamos a presentar la información mínima que debería conocer el usuario para poder usar nuestro sistema.

#### 3.2.9.1. Hardware

La placa dispone de una serie de conectores, jumpers, pulsadores y leds cuyas funciones describimos a continuación:

- El conector JP1 es donde se conecta el módulo Bluetooth. Se ha tomado como referencia la disposición de la mayoría de los módulos Bluetooth disponibles en el mercado. El patillaje es el siguiente:
  - 1: +3,3V.
  - 2: GND.

- 3: TX (del módulo BT).
- 4: RX (del módulo BT).
- El conector JP2 sirve para realizar la reprogramación del micro. Para realizar esta tarea el equipo debe estar desenchufado de la red eléctrica para evitar cortocircuitos. El patillaje es el siguiente:
  - 1: Vpp/ MCLR.
  - 2: Vcc placa.
  - 3: GND placa.
  - 4: ICSPDAT/PGD.
  - 5: ICSPCLK/PGC.
- El jumper JP3 es para seleccionar la salida por Bluetooth o por cable aislado. En modo cable aislado hay que hacer un puente entre las patillas 1-3 y otro entre las patillas 2-4. Para modo Bluetooth los dos puentes serán entre 4-6 y entre 3-4. El patillaje es el siguiente:
  - 1: RX cable.
  - 2: TX cable.
  - 3: RX.
  - 4: TX.
  - 5: RX Bluetooth.
  - 6: TX Bluetooth.
- El conector JP4 es para la conexión por cable aislado. Se ha tomado como referencia el conector típico de un cable FTDI con alimentación de 3,3 V. Siendo el patillaje el siguiente:
  - 1: GND.
  - 2: No conectar.
  - 3: +3,3V.
  - 4: TX (del cable).
  - 5: RX (del cable).
  - 6: No conectar.
- El pulsador S1 sirve para entrar en modo calibración colocando una carga resistiva pura que genere una corriente de 5A, durante el proceso el LED1 parpadeará y una vez haya terminado el proceso de calibración permanecerá encendido fijo durante 5 segundos.

- El pulsador S2 es para poner todos los contadores a cero. Si el proceso se ha realizado de forma satisfactoria se encenderá el LED2 durante 5 segundos, si se detecta algún problema parpadeará.
- El pulsador S3 es para el reseteo del microcontrolador.
- El LED1 además de la función indicada anteriormente, en el modo de funcionamiento normal generará un destello cada vez que se alcance un valor de energía activa consumida que se puede configurar.
- El LED2, al igual que el LED1 también, en su funcionamiento normal, genera un destello pero cuando se alcanza un valor de energía reactiva consumida configurable.

### 3.2.9.2. Firmware

El firmware no requiere de manual de usuario.

### 3.2.10. Fotos del prototipo

Aquí incluyo un par de fotos del prototipo usado para la realización de las pruebas.

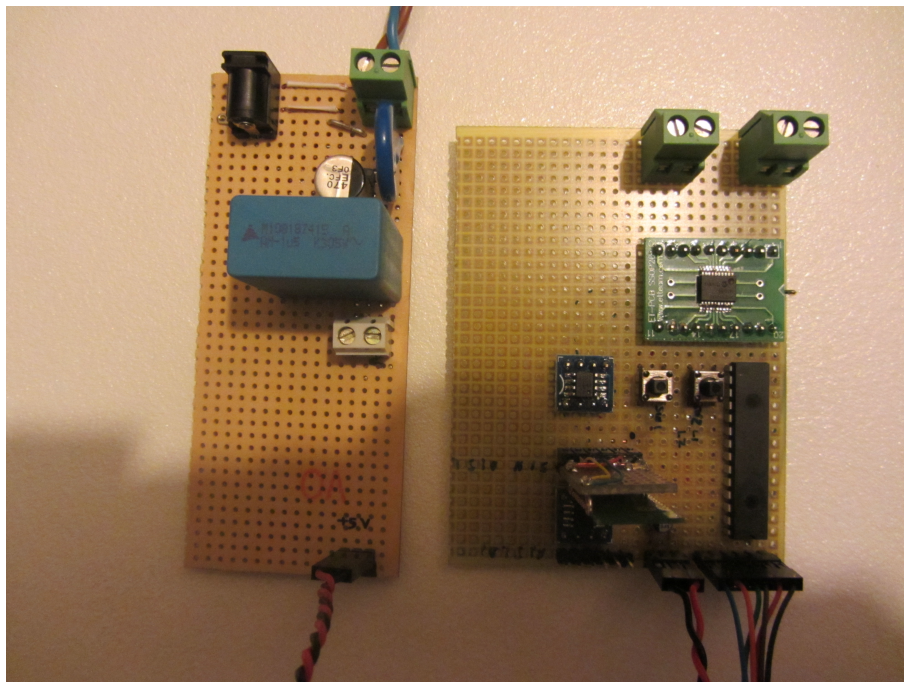


Figura 3.24: Foto del prototipo.

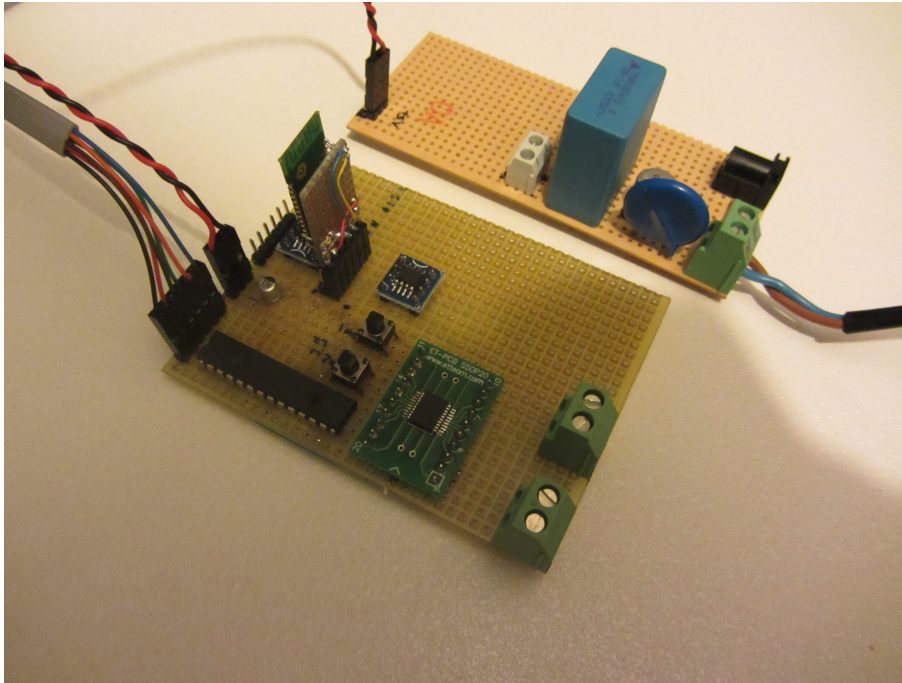


Figura 3.25: Otra vista del prototipo.

### 3.3. Software

El hardware sin control pierde toda su potencia. El control lo pone aquí el software que se encarga tanto de recuperar la información generada por el hardware como de presentarla en bruto al usuario o tratarla para que el usuario pueda obtener la información que requiera en cada momento.

#### 3.3.1. Introducción

El sistema está recopilando datos continuamente y los va almacenando en la EEPROM interna hasta que cuando aparezca un dispositivo autorizado la descargue.

Los datos en bruto son tratados en el microcontrolador del sistema de medición y a Android se le envían ya los datos procesados para que sólo se encargue de la presentación y generación de datos agregados.

El usuario interactúa con el medidor solicitando datos y modificando la configuración del equipo hardware.

El protocolo serie implementado para la transmisión de datos no contempla errores en el canal de transmisión ni encriptación de la información ya que no se considera un sistema crítico

y aunque un dato llegue erróneo o no llegue no implicará ninguna situación de peligro.

Como el sistema no se considera que sea crítico la contraseña de emparejamiento se guarda en claro en el fichero de preferencias pero como el alcance de bluetooth no es muy grande el robo de datos es difícil además de que los datos tampoco son sensibles.

### 3.3.2. Diagrama de bloques

En la Figura 3.26 podemos ver los elementos básicos que componen la parte de software que corre sobre Android.

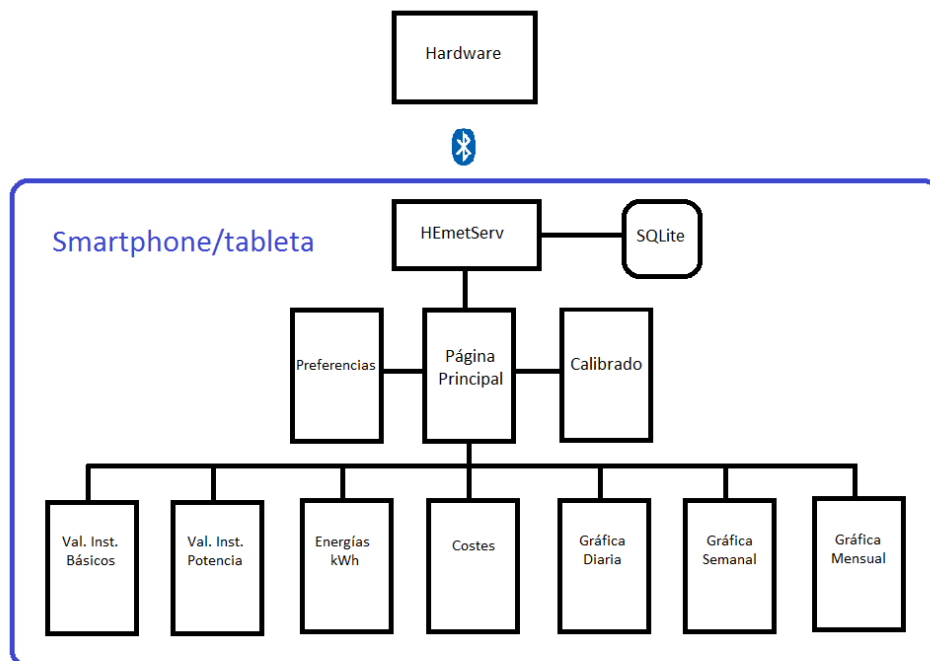


Figura 3.26: Diagrama de bloques de HEmet

A continuación vamos a describir, de forma somera, que hace cada uno de los elementos indicados en el diagrama anterior.

- HEmetServ. Es un servicio que vía Bluetooth se conecta con el hardware para realizar la transferencia de los datos.
- SQLite. Base de datos donde se almacenan los datos para generar gráficas históricas.
- Preferencias. Página de preferencias de la aplicación donde se almacenan datos como la contraseña de emparejamiento Bluetooth o el precio del kWh para calcular los costes.
- Página principal. Es la que arranca el servicio de recuperación de datos



- Pagina de calibrado. Permite la recalibración del equipo de medida.
- Valores instantáneos básicos. Presenta en pantalla y en tiempo real los valores de tensión, corriente y frecuencia de la señal medida.
- Valores instantáneos de potencia. Presenta en pantalla y en tiempo real los valores de potencia activa, potencia reactiva y potencia aparente.
- Valores energéticos. Presenta en pantalla los valores de energía activa, energía reactiva y energía aparente consumidas entre dos periodos dados.
- Costes. Presenta en pantalla los costes de los consumos.
- Gráfica diaria. Presenta graficas del último día de potencias y energías consumidas.
- Gráfica semanal. Presenta graficas de la última semana de potencias y energías consumidas.
- Gráfica mensual. Presenta graficas del último mes de potencias y energías consumidas.

Para desarrollar la aplicación

### 3.3.3. Manual de usuario

En esta sección vamos a distinguir dos partes:

1. Instalación del software en un dispositivo Android
2. Uso del software en un dispositivo Android

#### 3.3.3.1. Instalación

Antes de comenzar sólo indicar que las aplicaciones Android tienen extensión .apk y que están compiladas para una versión en particular de Android y posteriores, con lo que si nuestro dispositivo tiene una versión de Android anterior a la indicada en el software lo más posible es que no funcione.

Nuestra aplicación está compilada para la versión 2.3.3 de Android para que se pueda usar en teléfonos y en tabletas y debería funcionar en cualquier versión de Android superior a la indicada.

Para instalar el software realizaremos los siguientes pasos:

1. Comprobar que tenemos activado el check de “Instalar aplicaciones de origen desconocido” para poder hacer instalaciones de software que no estén en Google Play Store (Market de Android). Esta opción normalmente está en: Configuración > Aplicaciones.
2. Copiar el fichero HEmet.apk al dispositivo Android.
3. Con un navegador de ficheros ponerse en el directorio donde se haya copiado la aplicación.
4. Ejecutar el fichero HEmet.apk para que se instale.
5. El proceso ya está finalizado.

### 3.3.3.2. Uso

El uso de la aplicación HEmet es muy sencillo, solo hay que ir a la ventana de aplicaciones de Android y pinchar sobre ella. Debemos tener en cuenta que el dispositivo Android debe tener arrancado Bluetooth, pero, si no es así lo detectará y nos pedirá que lo arranquemos. En el menú de preferencias se configurará la conexión Bluetooth para emparejar el dispositivo con nuestro sistema hardware (con identificador linvor) antes de arrancar la aplicación. La contraseña de emparejamiento por defecto es 1234 aunque se podría cambiar en el módulo Bluetooth usando comandos AT (AT+PINxxxx) dónde xxxx es la nueva clave.

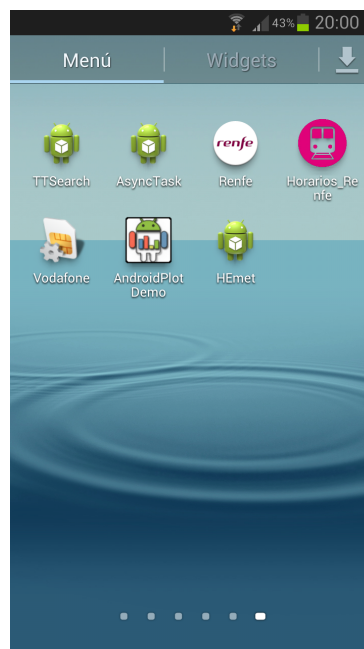


Figura 3.27: Arrancar la aplicación

Una vez arrancada la aplicación aparecerá la pantalla principal

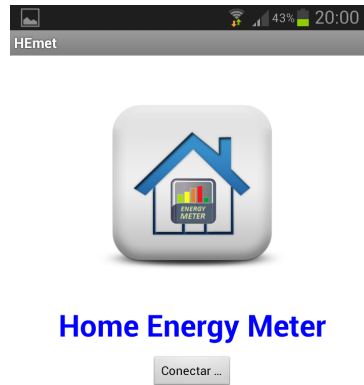


Figura 3.28: Pantalla de inicio de HEmet

Al pulsar Conectar se establecerá la conexión bluetooth con el hardware y comenzará a recibir datos que presentará en las pantallas de valores instantáneos: la básica con valores de tensión, corriente y frecuencia con un gráfico en tiempo real de variaciones del parámetro anterior que indiquemos; la de potencias, con potencias activas, reactivas y aparente y la de consumos de energías Activa, reactiva y aparente con los valores indicados entre dos fechas.

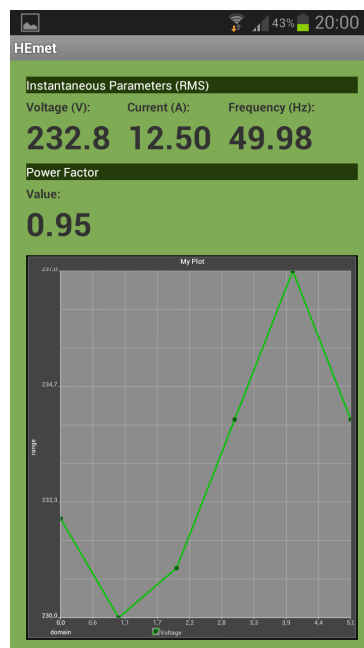


Figura 3.29: Pantalla de datos básicos de HEmet

Otra pantalla nos da información del coste del consumo eléctrico entre dos fechas obteniendo el precio del kWh de la pantalla de preferencias.

Y en otras pantallas visualizaremos gráficas diaria, semanal, mensual y a medida de potencias para poder detectar picos y valles de consumos.

## 3.4. Entorno de desarrollo

Para realizar el presente trabajo se necesitaros los siguientes elementos:

1. Java
2. Entorno de desarrollo Eclipse y Android SDK ADT Bundle
3. Git. Como sistema de control de versiones
4. github. Como forja de almacenamiento público de la aplicación
5. Otros

A continuación pasaremos a describir cada uno de ellos.

### 3.4.1. Java

Para el entorno de desarrollo funcione correctamente necesitamos tener Java instalado. para lo cual realizaremos los siguientes pasos:

1.- Descargar Java de <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

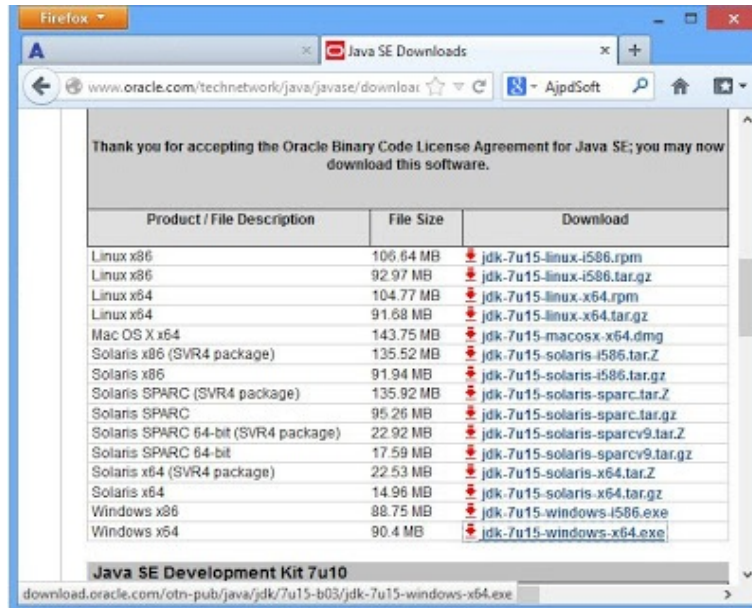


Figura 3.30: Instalación de Java 1

2.- Ejecutar el fichero descargado y (en Windows) contestar Sí cuando nos pida permiso para realizar cambios en el equipo.

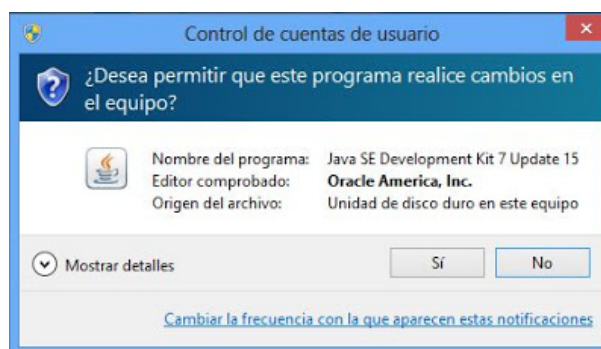


Figura 3.31: Instalación de Java 2

3.- Arrancará el asistente de instalación. Pulsar Next.



Figura 3.32: Instalación de Java 3

4.- Elegir los componentes a instalar (Development Tools, Source Code, Public JRE), la ubicación de los ficheros de instalación y pulsar Next.



Figura 3.33: Instalación de Java 4

5.- Elegir la carpeta donde se instalará JRE y pulsar Next.

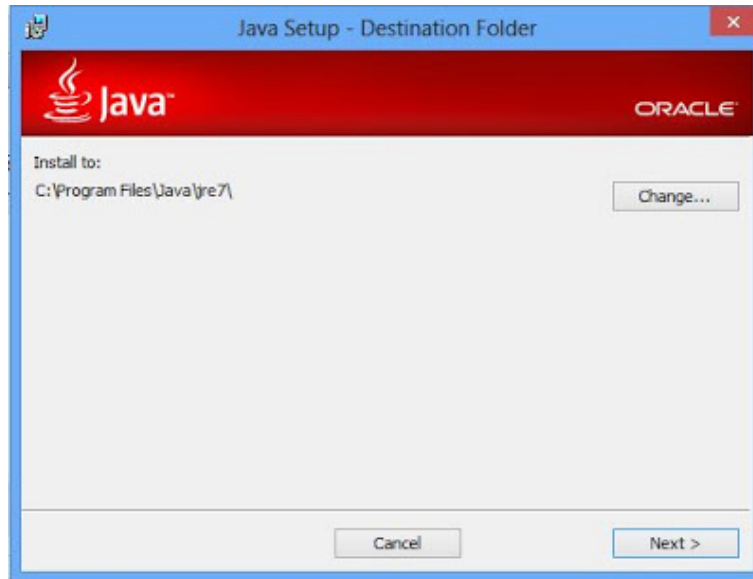


Figura 3.34: Instalación de Java 5

6.- Una vez instalado JDK y JRE pulsar Close en el asistente de instalación



Figura 3.35: Instalación de Java 6

Ya tenemos el primer elemento instalado.

### 3.4.2. Entorno de desarrollo Eclipse y Android SDK ADT Bundle

Ahora realizaremos la instalación del bundle de Eclipse + Android SDK ADT. para lo cual realizaremos los siguientes pasos:

1.- Descargar el bundle de <http://developer.android.com/sdk/index.html>.

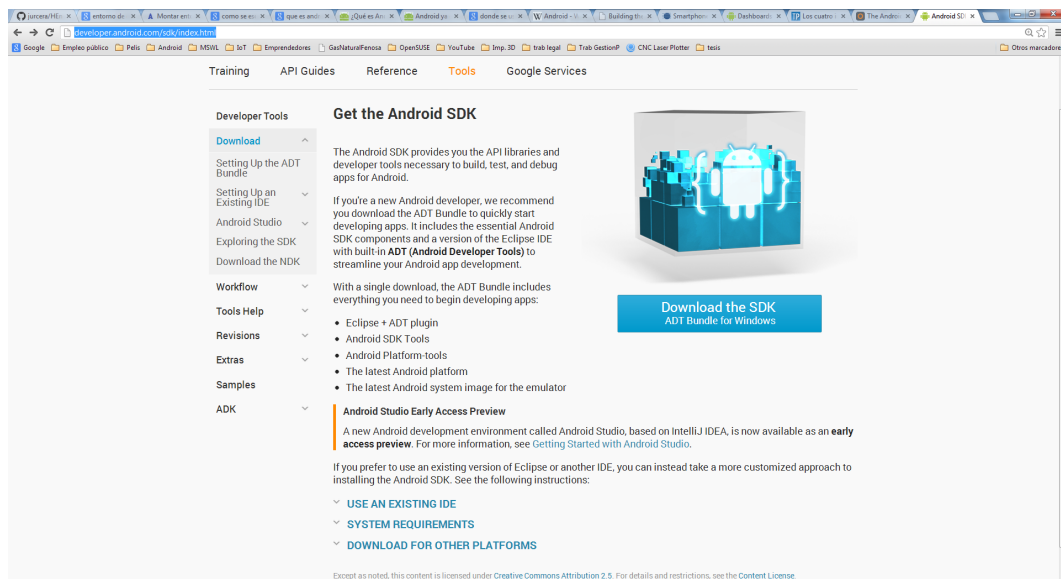


Figura 3.36: Instalación de Eclipse 1

2.- Una vez descargado el fichero, que ocupa aproximadamente unos 425 MB, debemos descomprimirlo en el directorio que tengamos designado para ello. La instalación es así de sencilla.

3.- Arrancaremos Eclipse para realizar la primera configuración y nos aparecerá la siguiente ventana:



Figura 3.37: Instalación de Eclipse 2

3.- En el primer inicio nos solicitará la carpeta para el espacio de trabajo (Workspace). Indicaremos la carpeta y pulsaremos OK



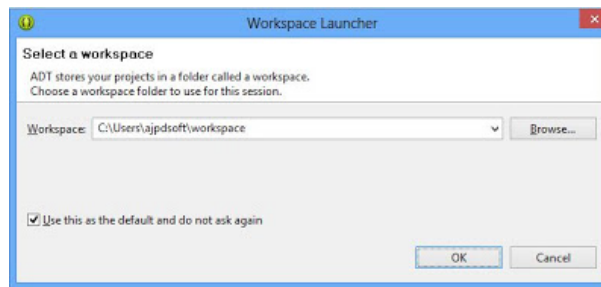


Figura 3.38: Instalación de Eclipse 3

4.- Si queremos enviar estadísticas de uso anónimas a Google pulsaremos Yes, en caso contrario pulsaremos No

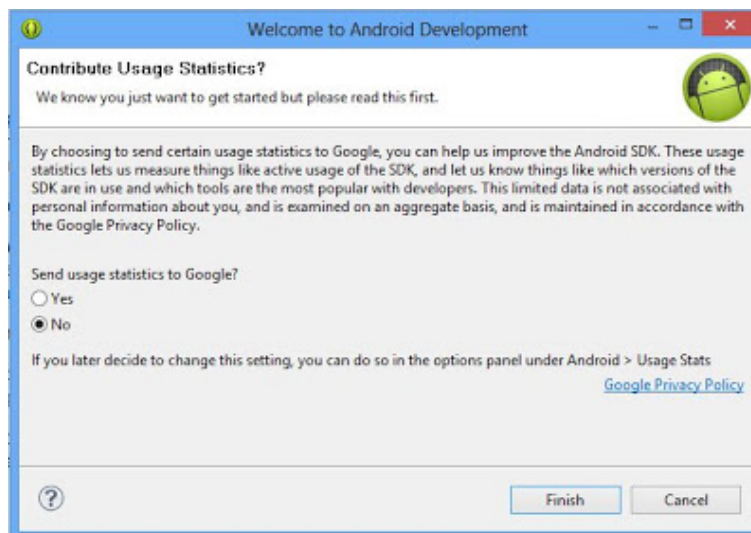


Figura 3.39: Instalación de Eclipse 4

5.- Es recomendable actualizar el IDE Eclipse a la última versión, para ello pulsaremos en el menú Help > Check for Updates

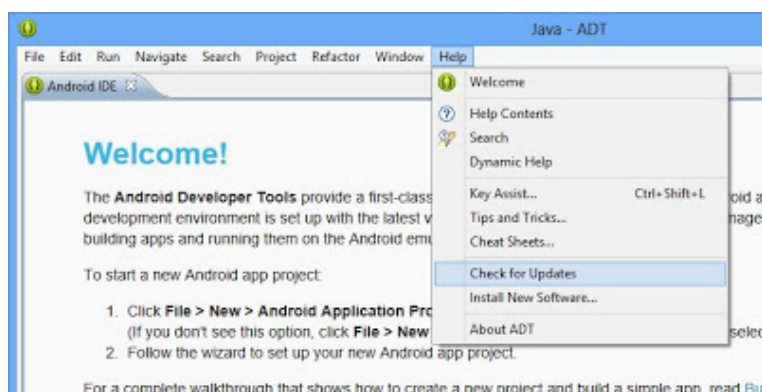


Figura 3.40: Instalación de Eclipse 5

6.- Para actualizar Android SDK y para instalar nuevos componentes pulsaremos en el menú Windows > Android SDK Manager

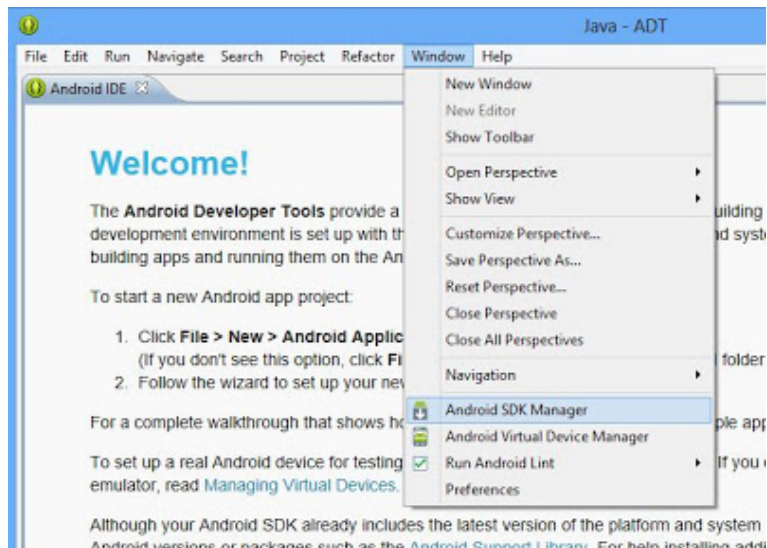


Figura 3.41: Instalación de Eclipse 6

7.- Elegiremos los componentes a instalar y los que queramos actualizar y pulsaremos Install packages

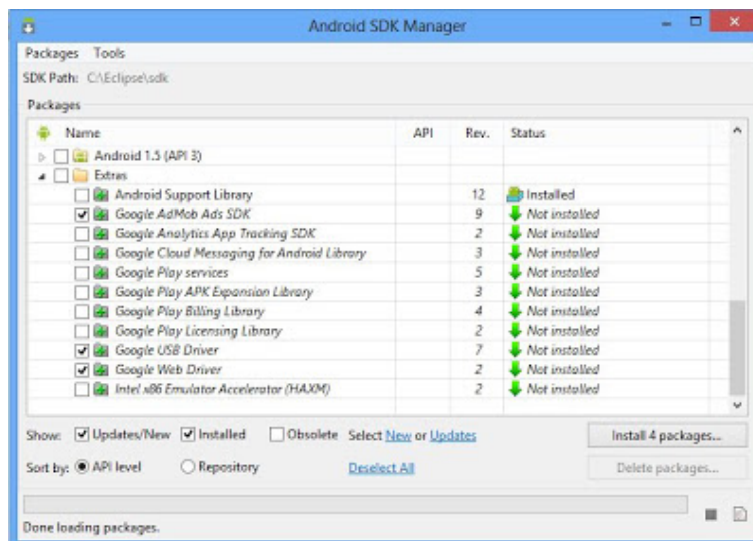


Figura 3.42: Instalación de Eclipse 7

8.- Si algún paquete requiere de aceptación de términos de licencia nos lo indicará, en nuestro caso hemos instalado: Samples for SDK API 17, revision 1, Google AdMob Ads SDK, revision 9, Google USB Driver, revision 7 y Google Web Driver, revision 2. Marcaremos Accept License y pulsaremos Install.

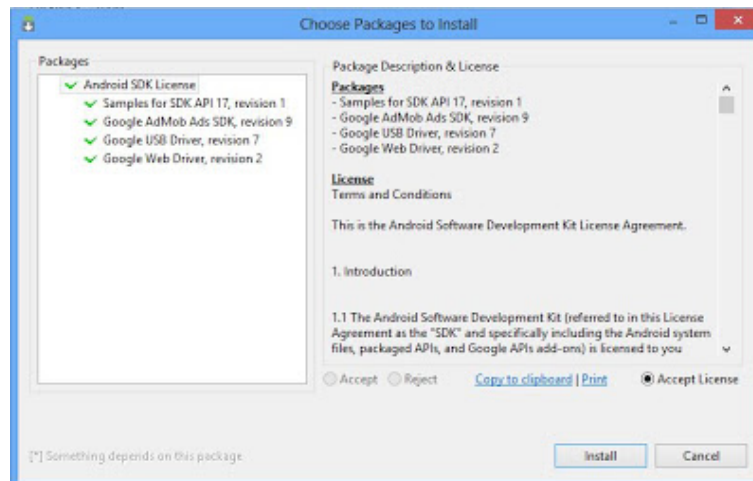


Figura 3.43: Instalación de Eclipse 8

Ya está listo eclipse con el SDK de Android, ya sólo nos falta configurar un dispositivo virtual AVD (Android Virtual Device) donde probar nuestras aplicaciones. Para ello seguiremos los siguientes pasos:

1.- Para agregar un nuevo dispositivo virtual pulsaremos en el menú Window > Android Virtual Device Manager

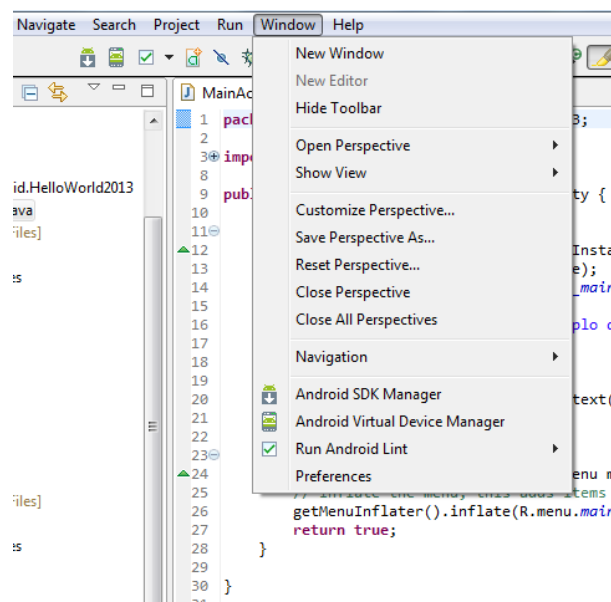


Figura 3.44: Configuración de un AVD 1

2.- En la pestaña Android Virtual Device pulsaremos en New. En este caso ya salen cuatro creados porque son los que uso habitualmente.

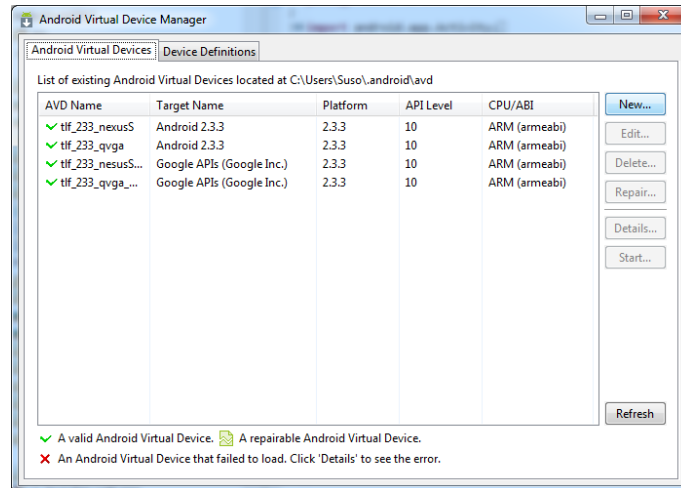


Figura 3.45: Configuración de un AVD 2

3.- En esta ventana salen una serie de datos a rellenar: AVD Name, es el nombre que le queramos dar, aconsejo que de la mayor cantidad posible de información como la versión de android y el tamaño de la pantalla; Device, aquí definimos el tamaño de la pantalla, una QVGA es manejable y se va bastante bien; Target, es la versión de Android que corre, ojo si vamos a usar mapas tenemos que seleccionar la que ponga Google APIs y, lo demás, son configuraciones que en estos momentos no necesitamos.

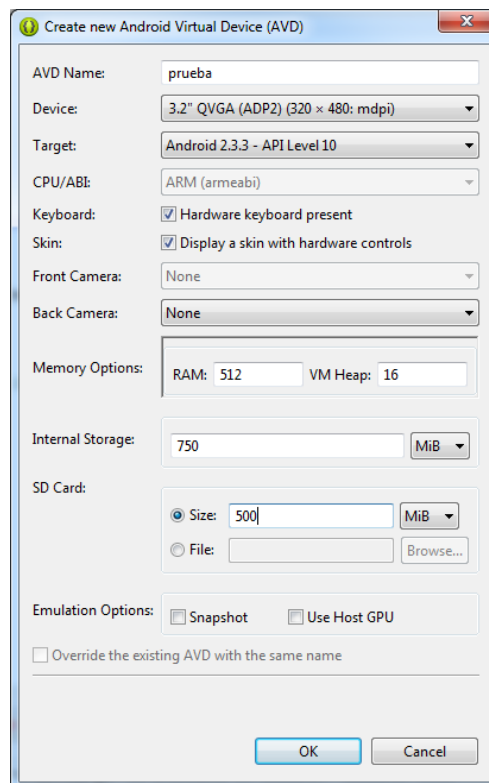


Figura 3.46: Configuración de un AVD 3

### 3.4.3. Forja: GitHub

Todo proyecto de Software Libre necesita una forja donde poner a disposición de los usuarios de los colaboradores el código fuente de la aplicación. En este caso y después de realizar varias comparativas durante el máster he decidido colgar en GitHub el proyecto completo.

Mediante el conector de GitHub que tiene Eclipse se pueden subir directamente las modificaciones a GitHub desde Eclipse sin necesidad de usar otro software.

GitHub además de repositorio de código nos ofrece otros recursos básicos para el proyecto como la wiki, para realizar la documentación del mismo y la gestión de errores con su issue tracker.

A continuación indicaremos los pasos a seguir para crear un proyecto en GitHub:

- 1.- Primero hay que darse de alta como usuario en GitHub.
- 2.- Una vez dentro de GitHub creamos un nuevo repositorio pulsando sobre el botón New repository.

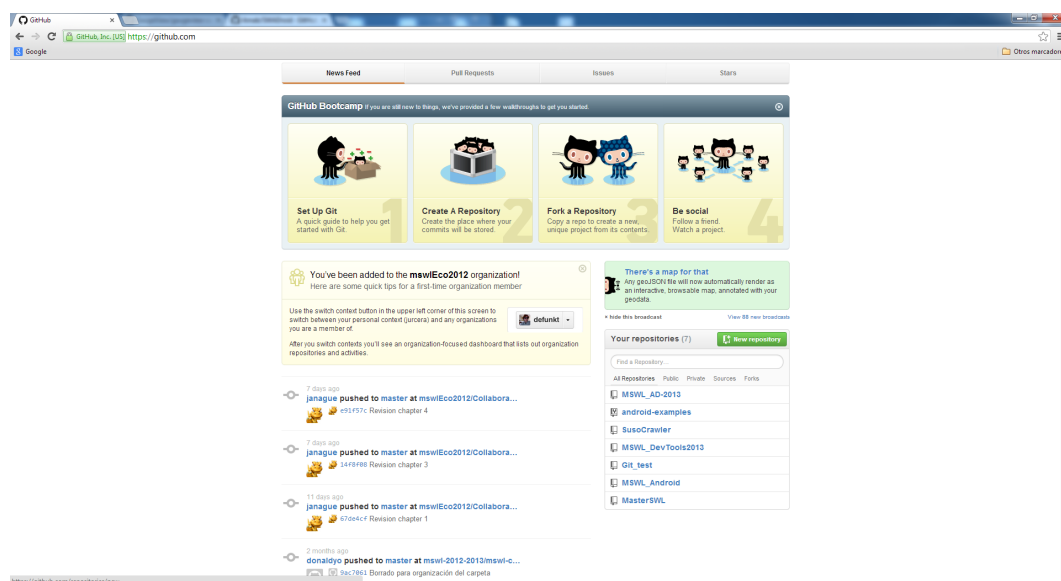


Figura 3.47: Creación de un repositorio en GitHub 1

3.- En la siguiente pantalla debemos indicar el nombre del repositorio, si es público o privado, en nuestro caso será público y además marcaremos la opción de inicializar el repositorio con un README. Una vez hechas estas tres cosas sólo es necesario pulsar el botón Create repository, como podemos ver en la figura 3.49.

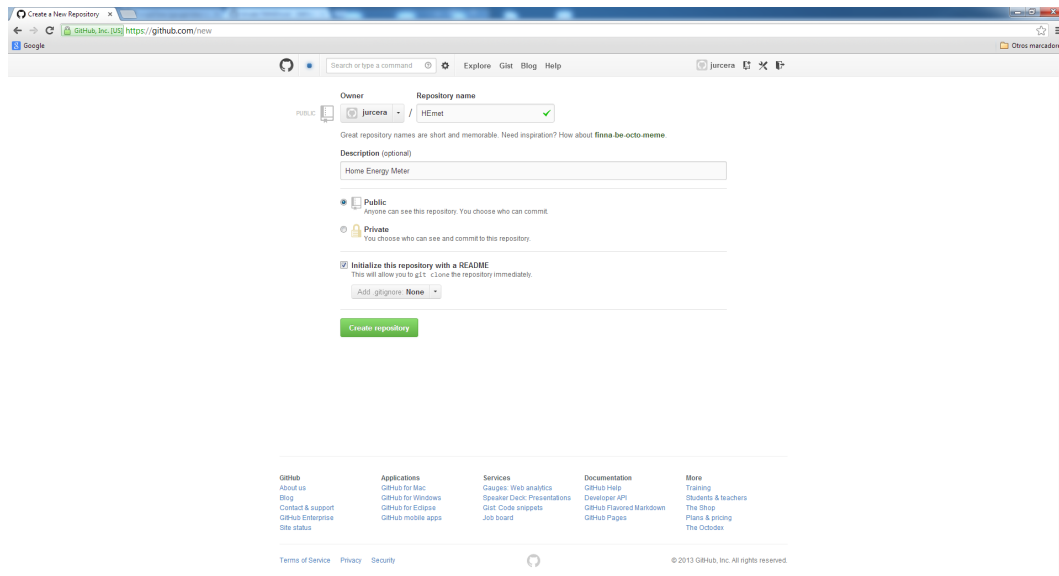


Figura 3.48: Creación de un repositorio en GitHub 2

4.- Y, para finalizar, nos aparecerá la página principal del repositorio, como podemos ver en la figura 3.49.

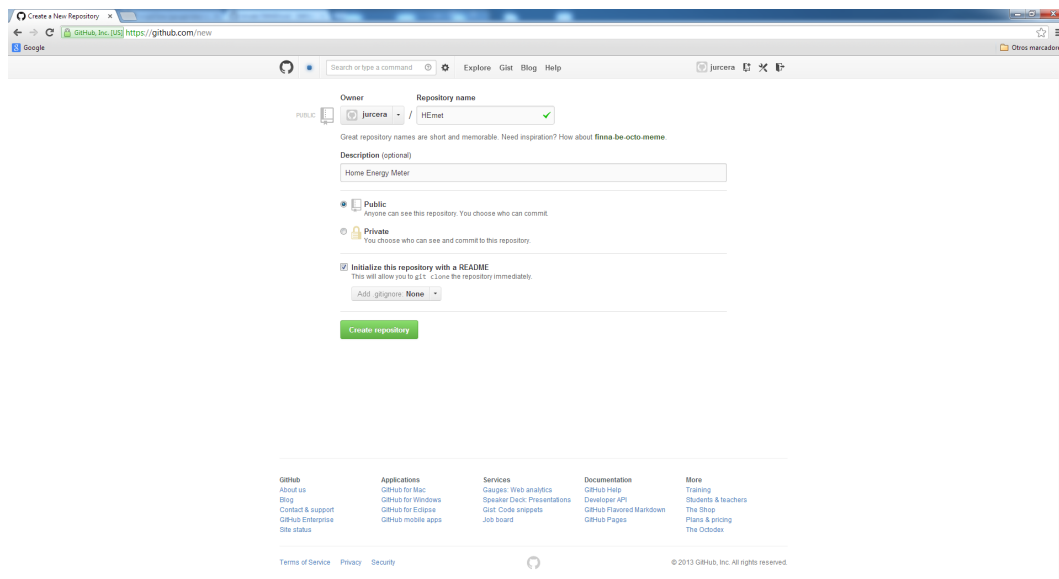


Figura 3.49: Creación de un repositorio en GitHub 3

Con esto ya tenemos el repositorio en GitHub listo para subir código.

### 3.4.4. Otras herramientas

Con la intención de fomentar el conocimiento del proyecto y obtener colaboración de otros miembros de la comunidad se pretende poner a disposición de la misma los siguientes elemen-

tos:

- Code: `git clone git://github.com/jurcera/HEmet.git`
- Documentation: <https://github.com/jurcera/HEmet/wiki>
- Issue tracker: <https://github.com/jurcera/HEmet/issues>
- Mailing list: <https://groups.google.com/forum/#!forum/hemet>
- Twitter: <http://twitter.com/HEmet>
- Facebook: <https://www.facebook.com/HEmet>
- Blog: <http://blog.urcera.com/wordpress>





# Capítulo 4

## Presupuesto

En este capítulo se detallan todos los elementos necesarios para la realización del presente trabajo y con sus correspondientes valoraciones. Se han dividido en los siguientes conceptos:

1. Ejecución material (materiales y mano de obra)
2. Gastos generales y beneficio industrial
3. Honorarios de dirección y redacción
4. Coste de ejecución por contrata
5. Presupuesto total

### 4.1. Costes ejecución material

Los costes de ejecución material incluyen todos los costes asociados al material, tanto hardware como software y el tiempo empleado en el desarrollo del trabajo.

#### 4.1.1. Coste del hardware

Vamos a distinguir entre el equipamiento empleado y los componentes y materiales electrónicos usados para el montaje del prototipo. Consideramos equipamiento a todo útil necesario en cualquiera de las fases del desarrollo del trabajo y consideramos componentes y materiales electrónicos a todo elemento empleado en el montaje del prototipo, incluyendo los fungibles.

##### 4.1.1.1. Componentes y materiales electrónicos

En este apartado se valora en coste de los componentes y materiales electrónicos usados en el montaje del prototipo. No se ha incluido el coste de la realización de los PCB de los prototipos ya que al final, por cuestiones de tiempo, no se han llegado a solicitar.

Cantidad	Dispositivo)	Valor	Precio unitario	Totales (€)
2	10-XX	Pulsador rojo/negro	1,2900	2,58
2	LEDCHIP-LED0603	APT1608EC	0,0848	0,17
1	C-EU225-087X268	1.5uF 305Vac	0,7390	0,74
3	R-EU-R0603	1K	0,0126	0,04
4	R-EU-R0805	1K	0,0250	0,10
2	R-EU-R0603	1k2	0,0122	0,02
3	R-EU-R0603	4K7	0,0126	0,04
1	R-EU-R0603	10K	0,0126	0,01
2	R-EU-R0603	10R	0,0126	0,03
1	C-EUC0603	10uF	0,1680	0,17
2	CPOL-EUA/3216-18R	10uF	0,4480	0,90
1	CPOL-EUB	10uF	0,4200	0,42
1	25LC256-I/SN	25LC256-I/SN	1,5400	1,54
1	R-EUR2512	100R/2W	0,8500	0,85
15	C-EUC0603	100nF	0,0295	0,44
1	C-EUC0603	100nF	0,0068	0,01
2	C-EUC1206	100nF	0,1792	0,36
3	LI0805H151R-10	150 ohm	0,0524	0,16
2	LI1806C151R-10	150 ohm	0,1220	0,24
2	R-EU-R1206	332K	0,0126	0,03
1	CPOL-EUG	470uF	0,9500	0,95
1	ADUM1201	ADUM1201	5,7300	5,73
1	BZG03C15	BZG03C15	0,3544	0,35
1	PINHD-1X4	Bluetooth (Serial)	0,2400	0,24
1	JP3Q	Bluetooth/Wire	0,2500	0,25
1	PINHD-1X5	ICSP	0,2400	0,24
1	DIODE-SMB	MBRS140T3G	0,2488	0,25
1	MCP1700T-3302E/TT	MCP1700T-3302E/TT	0,5380	0,54
1	MCP1703-5002E/DB	MCP1703-5002E/DB	0,3420	0,34
1	MCP3901	MCP3901	1,9600	1,96
2	MKDSN1,5/2-5,08	MKDS 3/2-5,08	1,0200	2,04
1	PIC18F25K80-I/SS	PIC18LF26K80-I/SS	1,9500	1,95
1	10-XX	Reset	1,2900	1,29
1	S20K275	S20K275	1,5400	1,54
1	RCA4070	SHUNT 0R005	1,4800	1,48
1	PINHD-1X6	WIRE (Serial)	0,2400	0,24
1	Caja	CNMB/2V/KIT	11,1500	11,15
1	PCB	PCB	10,2700	10,27
1	Estaño	Estaño/Pasta de soldar	3,0000	3,00
1	Módulo Bluetooth	Módulo Bluetooth	7,0200	7,02
1	Cable FTDI	Cable FTDI	12,8700	12,87
<b>Subtotal</b>				<b>72,54 €</b>

Cuadro 4.1: Costes de hardware. Componentes y materiales electrónicos.

#### 4.1.1.2. Equipamiento

En la tabla 4.2 aparece el listado de todo el equipamiento usado. Hay un caso especial, la UPS, que se usó par aislar el osciloscopio de la red eléctrica debido al problema de masa caliente (hot or live ground) comentado en el apartado 3.2.3.7 .

Concepto	Precio (€)	Amortización (años)	Uso (meses)	Totales (€)
Ordenador portátil HP core i7 y 6 GB RAM	1029	5	4	68,60
Impresora laser HP 1300	300	5	4	20,00
Osciloscopio Tektronix TDS2014	1800	5	4	120,00
Polímetro Fluke 289	680	5	4	45,33
Estación soldadura OKI HCT-900-21	500	5	4	33,33
Soldador JBC 14N	30	5	4	2,00
Kit desarrollo Microchip MCP3901	120	5	4	8,00
UPS online 3KVA PHASAK PH9230	650	5	4	43,33
Fuente A. triple salida DC 0-30V / 0-5A	280	5	4	18,67
PicKit 3	30	5	4	2,00
Tablet Asus NEXUS 7	250	5	4	16,67
Adaptador Bluetooth Manhattan	12	5	4	0,80
<b>Subtotal</b>				<b>378,73 €</b>

Cuadro 4.2: Costes de hardware. Equipamiento.

#### 4.1.2. Coste del software

La mayor parte es Software Libre, con coste cero pero. El resto de paquetes aunque disponen de versiones de evaluación o limitadas en prestaciones que se podrían usar en entornos educativos o personales sin ánimo de lucro, se han incluido las versiones profesionales para obtener datos lo más reales posible. El SO usado viene impuesto por ciertas aplicaciones privativas.

Concepto	Precio (€)	Amortización (años)	Uso (meses)	Totales (€)
Windows 7 professional	300	5	4	20,00
TeXworks con MiKTeX (L <sup>A</sup> T <sub>E</sub> X)	0	5	4	0,00
Mplab X	0	5	4	0,00
Eclipse	0	5	4	0,00
Android Development Kit	0	5	4	0,00
Java	0	5	4	0,00
Compilador PicC18	370	5	4	24,67
Eagle PCB software Professional 6.2.0	1385	5	4	92,33
<b>Subtotal</b>				<b>137,00 €</b>

Cuadro 4.3: Costes de Software.

### 4.1.3. Coste por tiempo empleado

En este apartado se incluyen las horas de ingeniería empleadas para el desarrollo del trabajo y las horas de gestión documental.

Concepto	Horas empleadas	Precio/hora (€)	Totales (€)
Ingeniería	400	50	20.000,00
Gestión documental	80	15	1.200,00
<b>Subtotal</b>			<b>21.200,00 €</b>

Cuadro 4.4: Coste por tiempo empleado.

### 4.1.4. Coste total de presupuesto de ejecución material

En este apartado se reunifican los tres conceptos anteriores para obtener el coste total de ejecución material.

Concepto	Totales (€)
Coste del hardware	451,27
Coste del software	137,00
Coste del tiempo empleado	21.200,00
<b>Subtotal</b>	<b>21.788,27 €</b>

Cuadro 4.5: Coste total del presupuesto de ejecución material.

## 4.2. Gastos generales y beneficio industrial

Los gastos generales y beneficio industrial son los gastos obligados que se derivan de la utilización de las instalaciones de trabajo, amortizaciones, etc., sumándose a ellos el beneficio industrial. El porcentaje establecido en estos conceptos es del 22 % sobre el presupuesto de ejecución material.

Concepto	Valor	Porcentaje	Totales (€)
Total del presupuesto de ejecución material	21.788,27	22 %	4.575,54
<b>Subtotal</b>			<b>4.575,54 €</b>

Cuadro 4.6: Gastos generales y beneficio industrial.

### 4.3. Honorarios de redacción

Se calcula como el 7 % del presupuesto de ejecución material.

Concepto	Valor	Porcentaje	Totales (€)
Total del presupuesto de ejecución material	21.788,27	7 %	1.525,18
<b>Subtotal</b>			<b>1.525,18 €</b>

Cuadro 4.7: Honorarios de redacción.

### 4.4. Coste de ejecución por contrata

Es la suma del presupuesto de ejecución material y gastos generales y beneficio industrial.

Concepto	Totales (€)
Presupuesto de ejecución material	21.788,27
Gastos generales y beneficio industrial	4.575,54
<b>Subtotal</b>	<b>26.363,81 €</b>

Cuadro 4.8: Coste de ejecución por contrata.

### 4.5. Importe total del presupuesto

El total es la suma del presupuesto de ejecución por contrata y los honorarios por redacción.

Concepto	Totales (€)
Coste de ejecución por contrata	26.363,81
Honorarios de redacción	1.525,18
<b>Subtotal</b>	<b>27.888,99 €</b>
IVA (21 %)	5.856,69
<b>Total</b>	<b>33.745,68 €</b>

Cuadro 4.9: Importe total del presupuesto.

**El importe total del proyecto suma la cantidad de: “TREINTA Y TRES MIL SETECIENTOS CUARENTA Y CINCO EUROS , CON SESENTA Y OCHO CÉNTIMOS”**



# Capítulo 5

## Conclusiones

### 5.1. Lecciones aprendidas

Como lecciones aprendidas podemos mencionar que con poco dinero podemos construir un sistema que nos permita tener a raya los consumos del hogar, detectar horas pico y horas valle o como organizarnos para que nuestras tareas no hagan saltar el limitador. A veces, encendemos aparatos sin tener en cuenta otros que ya hay encendidos y podemos llegar al límite máximo contratado, sin embargo en otros momento el consumo es mínimo. Si nos organizamos para realizar tareas en otros horarios podríamos incluso, llegar a reducir la potencia contratada en nuestro hogar ya que podemos ajustarnos a la que realmente necesitamos, bueno, no lo sabíamos hasta ahora, a partir de este momento y con nuestro sistema ya podremos llegar a saberlo.

### 5.2. Trabajos futuros

Trabajo futuro, hay mucho, por ejemplo: se podría seguir mejorando el desarrollo Android para poner alertas o avisos al llegar a unos determinados valores prefijados; se podrían controlar elementos como la lavadora o el lavavajillas para que se enciendan automáticamente cuando el sistema considere que es el más idóneo; se podría hacer un sistema reducido para conectar en cada enchufe de nuestra casa y nos envíe información del equipo que tiene conectado y centralizando todos los datos; se podría ir corrigiendo el factor de potencia en tiempo real, si supera unos umbrales prefijados, para consumir menos energía reactiva; se podría ...





# Apéndice A

## Microcontrolador PIC18LF26K80

El PIC18LF26K80 es un microcontrolador de 8 Bits de muy bajo consumo, altas prestaciones y fabricado por la empresa Microchip Technology Inc.

### Características básicas:

- Memoria: De programa 64 Kbytes, de datos 3648 Bytes y EEPROM 1024 Bytes.
- Patillas: 28 y de ellas 24 de Entrada/Salida.
- Periféricos:
  - 8 canales ADC de 12 bits,
  - 1 unidad de medida de tiempo de carga (CTMU),
  - 4 CCP (Capture/Compare/PWM) y 1 ECCP,
  - 2 timer de 8 bits y 3 de 16 bits,
  - 2 USART mejoradas (EUSART),
  - 2 comparadores,
  - 1 módulo ECAN y
  - 1 MSPP (Master Synchronous Serial Port) con I2C y SPI.
- Alimentación: 3,3 Vdc
- Modos de bajo consumo:
  - Run mode: 3.8 uA, valor típico.
  - Idle mode: 880 nA, valor típico.
  - Sleep mode: 13 nA, valor típico.

Ya que las hojas de características son demasiado extensas sólo se han incluido en este apéndice las secciones relativas a la introducción general.

En la figura A.1 podemos ver el patillaje del micro con todas las funciones asociadas a cada patilla. Claro está, que si estamos usando una de ellas, el resto de las funciones asociadas a esa patilla no se pueden usar al mismo tiempo.

28-Pin SSOP/SPDIP/SOIC

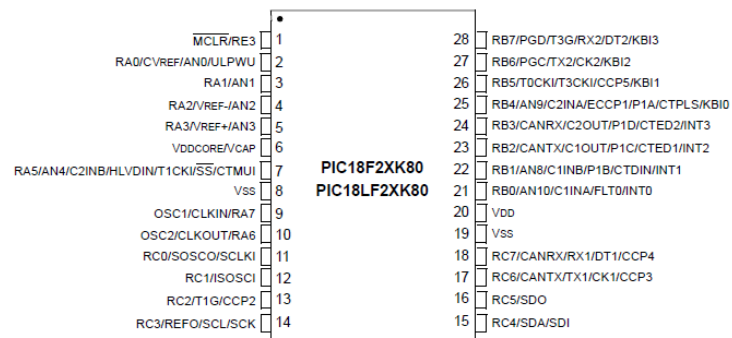


Figura A.1: Patillaje del PIC18LF26K80



# PIC18F66K80 FAMILY

## 28/40/44/64-Pin, Enhanced Flash Microcontrollers with ECAN™ and nanoWatt XLP Technology

### Power-Managed Modes:

- Run: CPU on, Peripherals on
- Idle: CPU off, Peripherals on
- Sleep: CPU off, Peripherals off
- Two-Speed Oscillator Start-up
- Fail-Safe Clock Monitor (FSCM)
- Power-Saving Peripheral Module Disable (PMD)
- Ultra Low-Power Wake-up
- Fast Wake-up, 1  $\mu$ s, Typical
- Low-Power WDT, 300 nA, Typical
- Run mode Currents Down to Very Low 3.8  $\mu$ A, Typical
- Idle mode Currents Down to Very Low 880 nA, Typical
- Sleep mode Current Down to Very Low 13 nA, Typical

### ECAN Bus Module Features:

- Conforms to CAN 2.0B Active Specification
- Three Operating modes:
  - Legacy mode (full backward compatibility with existing PIC18CXX8/FXX8 CAN modules)
  - Enhanced mode
  - FIFO mode or programmable TX/RX buffers
- Message Bit Rates up to 1 Mbps
- DeviceNet™ Data Byte Filter Support
- Six Programmable Receive/Transmit Buffers
- Three Dedicated Transmit Buffers with Prioritization
- Two Dedicated Receive Buffers

### ECAN Bus Module Features (Continued):

- 16 Full, 29-Bit Acceptance Filters with Dynamic Association
- Three Full, 29-Bit Acceptance Masks
- Automatic Remote Frame Handling
- Advanced Error Management Features

### Special Microcontroller Features:

- Operating Voltage Range: 1.8V to 5.5V
- On-Chip 3.3V Regulator
- Operating Speed up to 64 MHz
- Up to 64 Kbytes On-Chip Flash Program Memory:
  - 10,000 erase/write cycle, typical
  - 20 years minimum retention, typical
- 1,024 Bytes of Data EEPROM:
  - 100,000 Erase/write cycle data EEPROM memory, typical
- 3.6 Kbytes of General Purpose Registers (SRAM)
- Three Internal Oscillators: LF-INTOSC (31 KHz), MF-INTOSC (500 kHz) and HF-INTOSC (16 MHz)
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 4 ms to 4,194s
- In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug via Two Pins
- Programmable BOR
- Programmable LVD

TABLE 1: DEVICE COMPARISON

Device	Program Memory	Data Memory (Bytes)	Data EE (Bytes)	Pins	I/O	CTMU	12-Bit A/D Channels	CCP/ ECCP	Timers 8-Bit/16-Bit	EUSART	Comparators	ECAN™	MSSP	BORM/LVD	DSM
PIC18F25K80	32 Kbytes	3,648	1,024	28	24	1	8-ch	4/1	2/3	2	2	1	1	Yes	No
PIC18LF25K80	32 Kbytes	3,648	1,024	28	24	1	8-ch	4/1	2/3	2	2	1	1	Yes	No
PIC18F26K80	64 Kbytes	3,648	1,024	28	24	1	8-ch	4/1	2/3	2	2	1	1	Yes	No
PIC18LF26K80	64 Kbytes	3,648	1,024	28	24	1	8-ch	4/1	2/3	2	2	1	1	Yes	No
PIC18F45K80	32 Kbytes	3,648	1,024	40/44	35	1	11-ch	4/1	2/3	2	2	1	1	Yes	No
PIC18LF45K80	32 Kbytes	3,648	1,024	40/44	35	1	11-ch	4/1	2/3	2	2	1	1	Yes	No
PIC18F46K80	64 Kbytes	3,648	1,024	40/44	35	1	11-ch	4/1	2/3	2	2	1	1	Yes	No
PIC18LF46K80	64 Kbytes	3,648	1,024	40/44	35	1	11-ch	4/1	2/3	2	2	1	1	Yes	No
PIC18F65K80	32 Kbytes	3,648	1,024	64	54	1	11-ch	4/1	2/3	2	2	1	1	Yes	Yes
PIC18LF65K80	32 Kbytes	3,648	1,024	64	54	1	11-ch	4/1	2/3	2	2	1	1	Yes	Yes
PIC18F66K80	64 Kbytes	3,648	1,024	64	54	1	11-ch	4/1	2/3	2	2	1	1	Yes	Yes
PIC18LF66K80	64 Kbytes	3,648	1,024	64	54	1	11-ch	4/1	2/3	2	2	1	1	Yes	Yes

# PIC18F66K80 FAMILY

---

## Peripheral Highlights:

- Five CCP/ECCP modules:
  - Four Capture/Compare/PWM (CCP) modules
  - One Enhanced Capture/Compare/PWM (ECCP) module
- Five 8/16-Bit Timer/Counter modules:
  - Timer0: 8/16-bit timer/counter with 8-bit programmable prescaler
  - Timer1, Timer3: 16-bit timer/counter
  - Timer2, Timer4: 8-bit timer/counter
- Two Analog Comparators
- Configurable Reference Clock Output
- Charge Time Measurement Unit (CTMU):
  - Capacitance measurement
  - Time measurement with 1 ns typical resolution
  - Integrated voltage reference
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to Four External Interrupts
- One Master Synchronous Serial Port (MSSP) module:
  - 3/4-wire SPI (supports all four SPI modes)
  - I<sup>2</sup>C™ Master and Slave modes
- Two Enhanced Addressable USART modules:
  - LIN/J2602 support
  - Auto-Baud Detect (ABD)
- 12-Bit A/D Converter with up to 11 Channels:
  - Auto-acquisition and Sleep operation
  - Differential Input mode of operation
- Data Signal Modulator module:
  - Select modulator and carrier sources from various module outputs
- Integrated Voltage Reference

# PIC18F66K80 FAMILY

## 1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC18F25K80
- PIC18F26K80
- PIC18F45K80
- PIC18F46K80
- PIC18F65K80
- PIC18F66K80
- PIC18LF25K80
- PIC18LF26K80
- PIC18LF45K80
- PIC18LF46K80
- PIC18LF65K80
- PIC18LF66K80

This family combines the traditional advantages of all PIC18 microcontrollers – namely, high computational performance and a rich feature set – with an extremely competitive price point. These features make the PIC18F66K80 family a logical choice for many high-performance applications where price is a primary consideration.

### 1.1 Core Features

#### 1.1.1 nanoWatt TECHNOLOGY

All of the devices in the PIC18F66K80 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- **Alternate Run Modes:** By clocking the controller from the Timer1 source or the Internal RC oscillator, power consumption during code execution can be reduced.
- **Multiple Idle Modes:** The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further.
- **On-the-Fly Mode Switching:** The power-managed modes are invoked by user code during operation, allowing the user to incorporate power-saving ideas into their application's software design.
- **nanoWatt XLP:** An extra low-power BOR and low-power Watchdog timer

#### 1.1.2 OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18F66K80 family offer different oscillator options, allowing users a range of choices in developing application hardware. These include:

- External Resistor/Capacitor (RC); RA6 available
- External Resistor/Capacitor with Clock Out (RCIO)
- Three External Clock modes:
  - External Clock (EC); RA6 available
  - External Clock with Clock Out (ECIO)
  - External Crystal (XT, HS, LP)

- A Phase Lock Loop (PLL) frequency multiplier, available to the external oscillator modes which allows clock speeds of up to 64 MHz. PLL can also be used with the internal oscillator.
- An internal oscillator block that provides a 16 MHz clock ( $\pm 2\%$  accuracy) and an INTOSC source (approximately 31 kHz, stable over temperature and VDD)
  - Operates as HF-INTOSC or MF-INTOSC when block is selected for 16 MHz or 500 kHz
  - Frees the two oscillator pins for use as additional general purpose I/O

The internal oscillator block provides a stable reference source that gives the family additional features for robust operation:

- **Fail-Safe Clock Monitor:** This option constantly monitors the main clock source against a reference signal provided by the internal oscillator. If a clock failure occurs, the controller is switched to the internal oscillator, allowing for continued low-speed operation or a safe application shutdown.
- **Two-Speed Start-up:** This option allows the internal oscillator to serve as the clock source from Power-on Reset, or wake-up from Sleep mode, until the primary clock source is available.

#### 1.1.3 MEMORY OPTIONS

The PIC18F66K80 family provides ample room for application code, from 32 Kbytes to 64 Kbytes of code space. The Flash cells for program memory are rated to last up to 10,000 erase/write cycles. Data retention without refresh is conservatively estimated to be greater than 20 years.

The Flash program memory is readable and writable. During normal operation, the PIC18F66K80 family also provides plenty of room for dynamic application data with up to 3.6 Kbytes of data RAM.

#### 1.1.4 EXTENDED INSTRUCTION SET

The PIC18F66K80 family implements the optional extension to the PIC18 instruction set, adding eight new instructions and an Indexed Addressing mode. Enabled as a device configuration option, the extension has been specifically designed to optimize re-entrant application code originally developed in high-level languages, such as 'C'.

# PIC18F66K80 FAMILY

---

## 1.1.5 EASY MIGRATION

Regardless of the memory size, all devices share the same rich set of peripherals, allowing for a smooth migration path as applications grow and evolve.

The consistent pinout scheme used throughout the entire family also aids in migrating to the next larger device. This is true when moving between the 28-pin, 40-pin, 44-pin and 64-pin members, or even jumping from smaller to larger memory devices.

The PIC18F66K80 family is also largely pin compatible with other PIC18 families, such as the PIC18F4580, PIC18F4680 and PIC18F8680 families of microcontrollers with an ECAN module. This allows a new dimension to the evolution of applications, allowing developers to select different price points within Microchip's PIC18 portfolio, while maintaining a similar feature set.

## 1.2 Other Special Features

- **Communications:** The PIC18F66K80 family incorporates a range of serial communication peripherals, including two Enhanced USARTs that support LIN/J2602, one Master SSP module capable of both SPI and I<sup>2</sup>C™ (Master and Slave) modes of operation and an Enhanced CAN module.
- **CCP Modules:** PIC18F66K80 family devices incorporate four Capture/Compare/PWM (CCP) modules. Up to four different time bases can be used to perform several different operations at once.
- **ECCP Modules:** The PIC18F66K80 family has one Enhanced CCP (ECCP) module to maximize flexibility in control applications:
  - Up to four different time bases for performing several different operations at once
  - Up to four PWM outputs
  - Other beneficial features, such as polarity selection, programmable dead time, auto-shutdown and restart, and Half-Bridge and Full-Bridge Output modes
- **12-Bit A/D Converter:** The PIC18F66K80 family has a differential A/D. It incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period, and thus, reducing code overhead.

- **Charge Time Measurement Unit (CTMU):** The CTMU is a flexible analog module that provides accurate differential time measurement between pulse sources, as well as asynchronous pulse generation.

Together with other on-chip analog modules, the CTMU can precisely measure time, measure capacitance or relative changes in capacitance, or generate output pulses that are independent of the system clock.

- **LP Watchdog Timer (WDT):** This enhanced version incorporates a 22-bit prescaler, allowing an extended time-out range that is stable across operating voltage and temperature. See [Section 31.0 "Electrical Characteristics"](#) for time-out periods.

## 1.3 Details on Individual Family Members

Devices in the PIC18F66K80 family are available in 28-pin, 40/44-pin and 64-pin packages. Block diagrams for each package are shown in [Figure 1-1](#), [Figure 1-2](#) and [Figure 1-3](#), respectively.

The devices are differentiated from each other in these ways:

- Flash Program Memory:
  - PIC18FX5K80 (PIC18F25K80, PIC18F45K80 and PIC18F45K80) – 32 Kbytes
  - PIC18FX6K80 (PIC18F26K80, PIC18F46K80 and PIC18F66K80) – 64 Kbytes
- I/O Ports:
  - PIC18F2XK80 (28-pin devices) – Three bidirectional ports
  - PIC18F4XK80 (40/44-pin devices) – Five bidirectional ports
  - PIC18F6XK80 (64-pin devices) – Seven bidirectional ports

All other features for devices in this family are identical. These are summarized in [Table 1-1](#), [Table 1-2](#) and [Table 1-3](#).

The pinouts for all devices are listed in [Table 1-4](#), [Table 1-5](#) and [Table 1-6](#).

# PIC18F66K80 FAMILY

**TABLE 1-1: DEVICE FEATURES FOR THE PIC18F2XK80 (28-PIN DEVICES)**

Features	PIC18F25K80	PIC18F26K80
Operating Frequency	DC – 64 MHz	
Program Memory (Bytes)	32K	64K
Program Memory (Instructions)	16,384	32,768
Data Memory (Bytes)	3.6K	
Interrupt Sources	31	
I/O Ports	Ports A, B, C	
Parallel Communications	Parallel Slave Port (PSP)	
Timers	Five	
Comparators	Two	
CTMU	Yes	
Capture/Compare/PWM (CCP) Modules	Four	
Enhanced CCP (ECCP) Modules	One	
Serial Communications	One MSSP and Two Enhanced USARTs (EUSART)	
12-Bit Analog-to-Digital Module	Eight Input Channels	
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow, MCLR, WDT (PWRT, OST)	
Instruction Set	75 Instructions, 83 with Extended Instruction Set Enabled	
Packages	28-Pin QFN-S, SOIC, SPDIP and SSOP	

**TABLE 1-2: DEVICE FEATURES FOR THE PIC18F4XK80 (40/44-PIN DEVICES)**

Features	PIC18F45K80	PIC18F46K80
Operating Frequency	DC – 64 MHz	
Program Memory (Bytes)	32K	64K
Program Memory (Instructions)	16,384	32,768
Data Memory (Bytes)	3.6K	
Interrupt Sources	32	
I/O Ports	Ports A, B, C, D, E	
Parallel Communications	Parallel Slave Port (PSP)	
Timers	Five	
Comparators	Two	
CTMU	Yes	
Capture/Compare/PWM (CCP) Modules	Four	
Enhanced CCP (ECCP) Modules	One	
Serial Communications	One MSSP and Two Enhanced USARTs (EUSART)	
12-Bit Analog-to-Digital Module	Eleven Input Channels	
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow, MCLR, WDT (PWRT, OST)	
Instruction Set	75 Instructions, 83 with Extended Instruction Set Enabled	
Packages	40-Pin PDIP and 44-Pin QFN and TQFP	

# PIC18F66K80 FAMILY

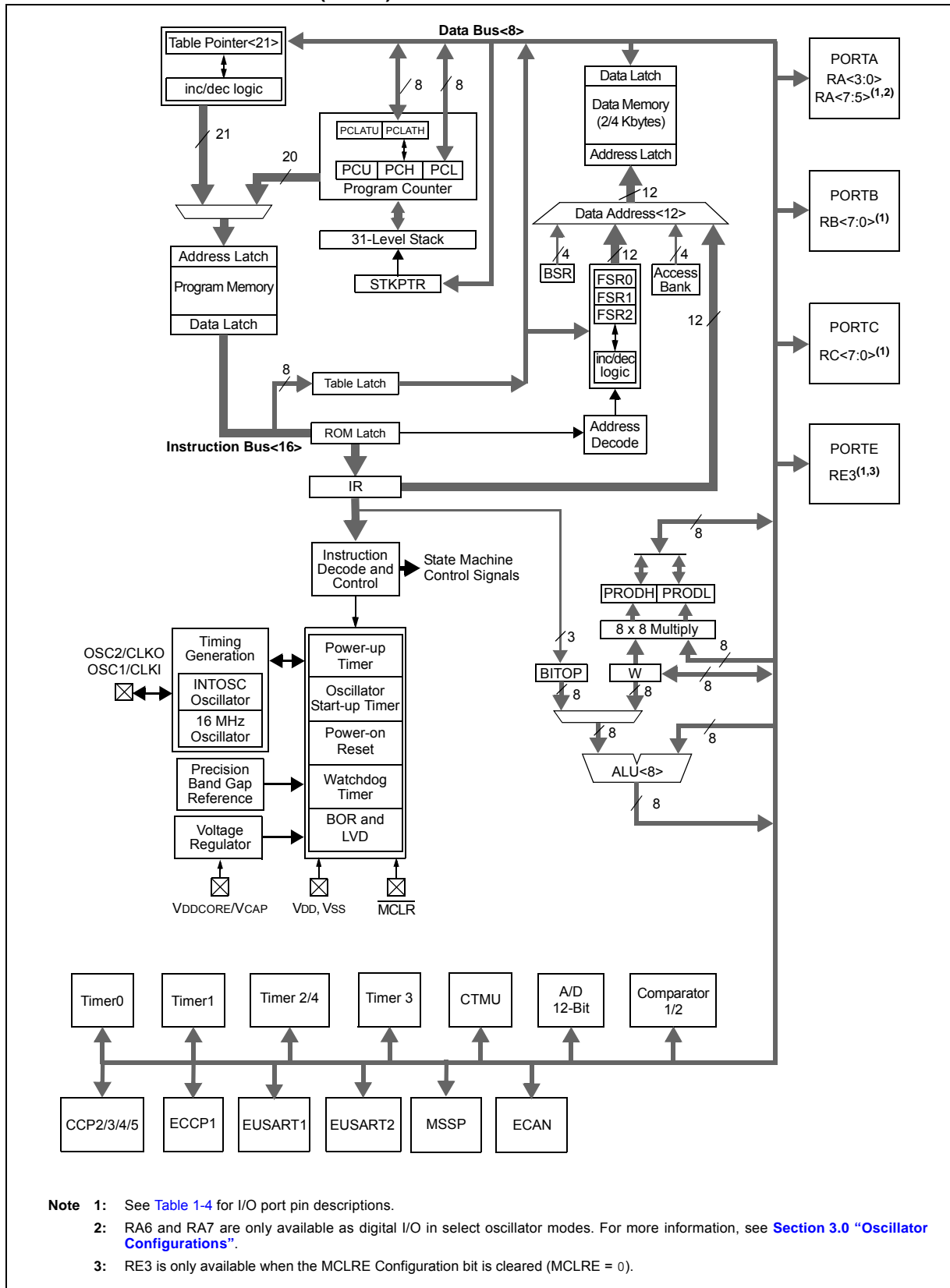
**TABLE 1-3: DEVICE FEATURES FOR THE PIC18F6XK80 (64-PIN DEVICES)**

Features	PIC18F65K80	PIC18F66K80
Operating Frequency	DC – 64 MHz	
Program Memory (Bytes)	32K	64K
Program Memory (Instructions)	16,384	32,768
Data Memory (Bytes)	3.6K	
Interrupt Sources	32	
I/O Ports	Ports A, B, C, D, E, F, G	
Parallel Communications	Parallel Slave Port (PSP)	
Timers	Five	
Comparators	Two	
CTMU	Yes	
Capture/Compare/PWM (CCP) Modules	Four	
Enhanced CCP (ECCP) Modules	One	
DSM	Yes	Yes
Serial Communications	One MSSP and Two Enhanced USARTs (EUSART)	
12-Bit Analog-to-Digital Module	Eleven Input Channels	
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow, $\overline{\text{MCLR}}$ , WDT (PWRT, OST)	
Instruction Set	75 Instructions, 83 with Extended Instruction Set Enabled	
Packages	64-Pin QFN and TQFP	



# PIC18F66K80 FAMILY

FIGURE 1-1: PIC18F2XK80 (28-PIN) BLOCK DIAGRAM





# Apéndice B

## Módulo Bluetooth

Módulo transceptor RS232TTL sin hilos de RF Bluetooth de 3,3 V y 10 m. de alcance con antena incluida.

Este pequeño módulo transceptor permite a un dispositivo enviar y recibir datos serie TTL, mediante tecnología Bluetooth, sin conectar ningún cable al ordenador. Es de uso muy fácil y compacto.

### Características:

- Transferencia transparente de datos serie TTL entre un Maestro y un dispositivo.
- Incluido chip Bluetooth BC417143.
- Especificación Bluetooth v2.0 + EDR.
- Trabaja con cualquier adaptador USB Bluetooth.
- Antena incluida.
- Tamaño reducido.

### Especificaciones:

- Tensión de alimentación: +3,3 Vdc.
- Consumo observado: 45 mA.
- Configuración de transmisión por defecto: 9600,8,n,1
- Alcance de la señal: 10 m.
- Tamaño: 26.9mm \* 13mm \* 2.2 mm

- Peso: 1g

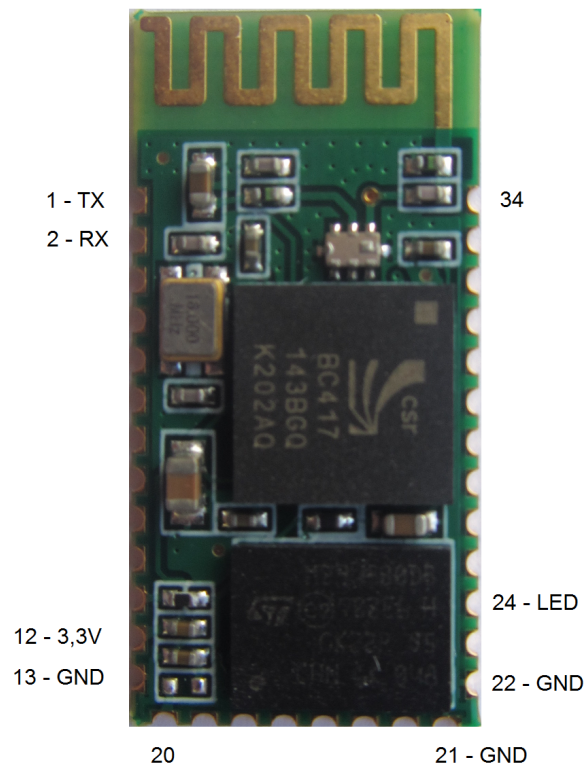
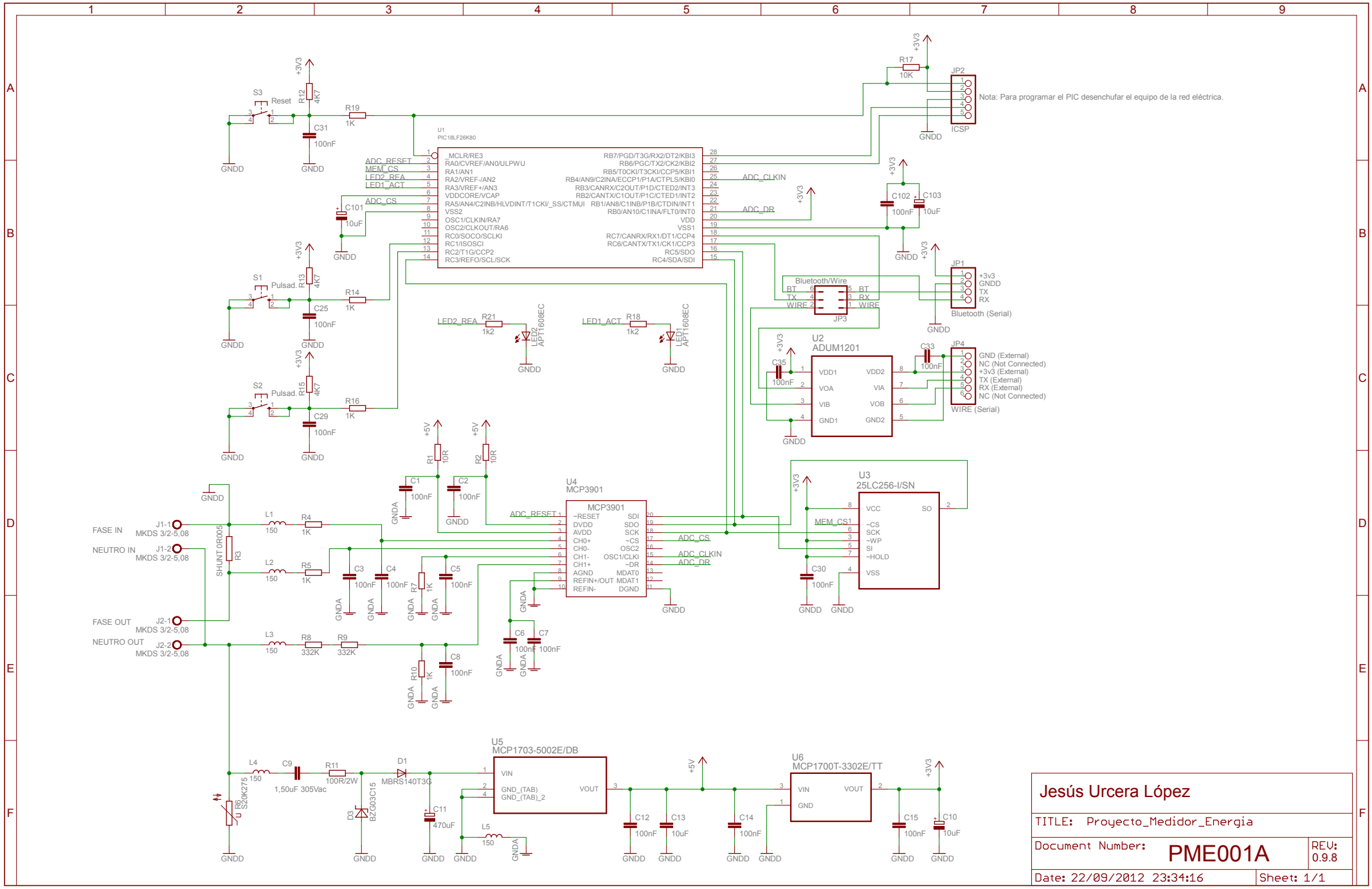


Figura B.1: Módulo Bluetooth

# Apéndice C

## Esquema electrónico

En la página siguiente podemos ver el esquema electrónico del circuito completo, a gran tamaño, para tener una mejor visión del mismo.



Nota: Para programar el PIC desenchufar el equipo de la red eléctrica.

<b>Jesús Urcera López</b>	
TITLE: Proyecto_Medidor_Energia	
Document Number:	<b>PME001A</b>
Date: 22/09/2012 23:34:16	REU: 0.9.8
Sheet: 1/1	

# Apéndice D

## Contenido del CD/DVD

En el CD/DVD se incluyen además de la presente documento y sus fuentes, los esquemas, la documentación de los anexos completa, los programas, tanto los que corren en el microcontrolador como los que corren en el Android y la bibliografía que sea de acceso público.

**La estructura de directorios es la siguiente:**

- Memoria.
- Anexos.
- Esquemas electrónicos.
- Software.
  - PIC.
  - Android.
- Bibliografía pública.





# Bibliografía

- [BDS13] Inc. Black Duck Software. Android project. Website, 2013. <http://www.ohloh.net/p/android>.
- [Con04] Reston Condit. Transformerless power supplies: Resistive and capacitive. Website, 2004. <http://ww1.microchip.com/downloads/en/AppNotes/00954A.pdf>.
- [Gar13] Inc. Gartner. Gartner says asia/pacific led worldwide mobile phone sales to growth in first quarter of 2013. Website, 2013. <http://www.gartner.com/newsroom/id/2482816>.
- [Inc10a] Microchip Technology Inc. Mcp3901 adc evaluation board for 16-bit mcus user's guide. Website, 2010. <http://ww1.microchip.com/downloads/en/DeviceDoc/51845B.pdf>.
- [Inc10b] Microchip Technology Inc. Mcp3901 low-cost power monitor reference design user's guide. Website, 2010. <http://ww1.microchip.com/downloads/en/DeviceDoc/51915A.pdf>.
- [Inc12a] Microchip Technology Inc. Mcp3901 and pic18f65j90, energy meter reference design user's guide. Website, 2012. <http://ww1.microchip.com/downloads/en/DeviceDoc/51968A.pdf>.
- [Inc12b] Microchip Technology Inc. Mcp3901 data sheet, two-channel analog front end. Website, 2012. <http://ww1.microchip.com/downloads/en/DeviceDoc/22192d.pdf>.
- [Inc12c] Microchip Technology Inc. Pic18lf26k80 data sheet. Website, 2012. <http://ww1.microchip.com/downloads/en/DeviceDoc/39977f.pdf>.
- [Inc13] Microchip Technology Inc. Energy metering. Website, 2013.
- [ItOHA13] Google Inc. and the Open Handset Alliance. Android dashboards. Website, 2013. <http://developer.android.com/about/dashboards/index.html>.

- [Kin09] Craig L. King. Iec compliant active-energy meter designusing the mcp3905a/06a. Website, 2009. <http://ww1.microchip.com/downloads/en/AppNotes/00994b.pdf>.
- [KQ09] Craig L. King and Vincent Quiquempoix. Designing with the mcp3901 dual channel analog-to-digital converters. Website, 2009. <http://ww1.microchip.com/downloads/en/AppNotes/01300a.pdf>.