

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN  
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



**Proyecto fin de carrera**

**Desarrollo e implementación de una plataforma  
hardware para el desarrollo de dispositivos de  
etiquetado electrónico.**



AUTOR: Raúl Sánchez Siles  
DIRECTOR(ES): Francisco M. Monzó Sánchez

agosto / 2005



<b>Autor</b>	Raúl Sánchez Siles
<b>E-mail del Autor</b>	rasasi78@gmail.com
<b>Director(es)</b>	Francisco M. Monzó Sánchez
<b>E-mail del Director</b>	franciso.monzo@upct.es
<b>Codirector(es)</b>	
<b>Título del PFC</b>	Desarrollo e implementación de una plataforma hardware para el desarrollo de dispositivos de etiquetado electrónico.
<b>Descriptores</b>	RFID plataforma inalámbrica inteligente activa

**Descripción:**

El sector logístico tiene una proyección muy extensa, donde se utilizan diversas tecnologías. El etiquetado electrónico de los productos tiene unas posibilidades cuya mejora se puede estudiar.

El objetivo es desarrollar una plataforma con la que se pueda contribuir a mejorar los sistemas existentes de etiquetado electrónico. Posteriormente realizar unos prototipos.

Se plantea una plataforma genérica basada en un microcontrolador con etapa de RF sobre el que hacer el desarrollo utilizando además puertos serie como base de la comunicación. Un teclado y un visualizador de cristal líquido proporcionan una interfaz para el usuario.

<b>Titulación</b>	Ingeniero de Telecomunicación
<b>Intensificación</b>	
<b>Departamento</b>	Tecnologías de la Información y de las Comunicaciones
<b>Fecha de Presentación</b>	septiembre / 2005

## Índice de contenido

1 Planteamiento.....	1
1.1 Tecnología aplicada a la logística.....	1
1.2 Puntos de control/supervisión.....	2
1.3 Entornos industriales / Equipos autónomos.....	3
1.4 Análisis de los datos.....	4
1.5 Aprovechamiento de avances tecnológicos.....	5
2 Estado del arte.....	7
2.1 Sistemas empleados en logística.....	7
2.1.1 Códigos de barras.....	7
2.1.2 Código electrónico de producto (EPC).....	9
2.1.3 Sistema de información EPC.....	9
2.2 Etiquetas RFID.....	10
2.2.1 Historia.....	11
2.2.2 Tipos.....	12
2.3 Comparación RFID / código de barras.....	14
2.4 Etiquetas activas.....	15
3 Estudio preliminar.....	17
3.1 Delimitación objetivo.....	17
3.2 Requisitos y arquitectura.....	17
3.3 Comunicaciones. Periféricos.....	18
3.3.1 Radiofrecuencia.....	18
3.3.2 Conexión cableada.....	20
3.3.3 Interfaz de programación.....	20
3.3.4 Interfaz de usuario.....	20
3.3.5 Sensor.....	20
3.3.6 Diagrama de bloques.....	20
3.4 Fases del desarrollo. Herramientas.....	21
3.5 Firmware / software asociado.....	22
4 Solución/implementación hardware.....	23
4.1 Selección plataforma de desarrollo.....	23
4.1.1 Microcontrolador.....	23
4.1.2 Plataforma.....	24
4.2 Otros componentes importantes. Consideraciones de diseño.....	25
4.2.1 Sensor de temperatura.....	25
4.2.2 Alimentación.....	25
4.2.3 Puertos serie.....	26
4.2.4 Teclado.....	27
4.2.5 Adaptadores de nivel.....	27
4.2.6 Consideraciones de diseño para el microcontrolador.....	28
4.3 Estudio de consumo.....	29
4.4 Diseño electrónico. Herramientas.....	30
4.4.1 Herramientas.....	30
4.4.2 Esquemático.....	31
4.4.3 PCB.....	31
4.5 Programación sistema.....	34
4.6 Construcción.....	35

5 Solución/implementación software.....	38
5.1 Herramientas.....	38
5.2 Programa de ejemplo.....	39
6 Pruebas.....	40
6.1 Pruebas eléctricas.....	40
6.2 Pruebas funcionales.....	40
6.3 Programación del microcontrolador. Pruebas del microcontrolador.....	41
6.4 Pruebas de prestaciones.....	41
7 Conclusiones. Líneas futuras.....	43
7.1 Conclusiones.....	43
7.2 Líneas futuras.....	43
7.2.1 Corto plazo.....	43
7.2.2 Medio plazo.....	43

**Índice de ilustraciones**

Figura 1: Diagrama genérico puntos de control.	3
Figura 2: Código de barras UPC	7
Figura 3: Código de barras EAN 13	7
Figura 4: Código 16K	8
Figura 5: Código maxicode	8
Figura 6: Código PDF417	8
Figura 7: Código QR	8
Figura 8: Lector de código de barras	8
Figura 9: Pila EPCIS.	10
Figura 10: Etiqueta RFID pasiva.	13
Figura 11: Etiqueta RFID activa.	13
Figura 12: Etiqueta inteligente sin contactos.	14
Figura 13: RFID activa y lector.	16
Figura 14: Diagrama de bloques de la plataforma.	21
Figura 15: Imagen del CC1010.	24
Figura 16: Diagrama del bloques de la plataforma.	25
Figura 17: Detalle etapa RF.	29
Figura 18: Software de programación del CC1010.	35
Figura 19: Capa PCB componentes.	36
Figura 20: Capa PCB soldadura.	36
Figura 21: Capa PCB serigrafía componentes.	36
Figura 22: Capa PCB serigrafía soldura.	37
Figura 23: Capa PCB taladros.	37

## Índice de tablas

Tabla 1: Historia y evolución tecnología RFID.	12
Tabla 2: Comparativa transmisión en RF.	19
Tabla 3: Estimación del consumo de los componentes.	29

## **1 Planteamiento.**

Los sistemas utilizados en logística han supuesto un gran avance en este sector. Éste destaca por su gran desarrollo, potencia y continua evolución. Su potencia es indiscutible: todos los productos tienen que llegar a un destino.

Pero para su total desarrollo han sido necesarias continuas evoluciones en los diversos Sistemas que acompañan a la logística. Estos sistemas como los electrónicos e informáticos que contribuyen a la logística serán objeto de este Proyecto.

La contribución de los Sistemas logísticos impacta directamente en campos tan importantes como el control de almacén, el seguimiento de la mercancía ó el estado de la mercancía. Esto tiene una importante repercusión en el coste y calidad del servicio. Mejorar estos sistemas redundará directamente en estos importantes aspectos. Como consecuencia la tecnología es un factor clave en el desarrollo del sector logístico.

### **1.1 Tecnología aplicada a la logística.**

La tecnología persigue mejorar los mecanismos con los que se desarrollan actividades cotidianas. En el caso de la logística, se trata de averiguar cuáles son los puntos en los que los procesos se pueden mejorar. Nos vamos a centrar en el seguimiento del ciclo del producto.

La *logística* es el arte y la ciencia de manejar y de controlar los flujos de mercancías, energía e información[WPO]<sup>1</sup>. En lo que a nosotros nos afecta, la logística hace que una gama de productos se distribuya de la mejor forma posible y llegue al destino adecuado para su uso o servicio. Para conseguir este objetivo existen dos enfoques:

1. Optimizar el flujo de material.
2. Coordinar los recursos.

El segundo punto corresponde a la fase de planificación de recursos, mientras que el primero aunque requiere de planificación, ésta debe tener en cuenta el tipo de recurso con el que se cuenta.

Para llevar a cabo el primer punto, es conveniente realizar un seguimiento exhaustivo del ciclo del producto a fin de conocerlo con detalle y poder actuar, mejorar o variar sus características. El modo en que se realiza este seguimiento determina, en gran medida, la efectividad y, por tanto, el éxito de la tarea. Es en este aspecto donde la tecnología puede ser un factor clave que establezca diferencias cualitativas entre dos realizaciones distintas de un objetivo logístico.

---

<sup>1</sup> <http://es.wikipedia.org/wiki/Logística>

Saber de forma precisa donde está la mercancía ó la cantidad de la misma que se tiene acopiada en algún almacén es básico para ejercer un control de la cadena logística. Esto proporciona ciertas prestaciones imprescindibles a la hora de acometer el objetivo. Además de esto, es posible mejorar la calidad de los procesos conociendo otros parámetros que pueden variar según el tipo de producto transportado.

Como ejemplo, sería interesante tener observaciones de la humedad y temperatura durante el transporte de alimentos perecederos. Los sistemas que dan soporte a los procesos logísticos deben estar preparados para contemplar estos parámetros. Existen varias posibilidades a la hora de hacer efectivo este control. En todos los casos es necesario obtener esta información con un dispositivo ligado más o menos estrechamente a la mercancía y por otro alguna plataforma de control que haga tareas de más alto nivel con los datos recibidos.

Con la tecnología existente hoy día, es posible dotar de inteligencia al dispositivo directamente relacionado con la mercancía de esta forma la obtención de datos se puede realizar de forma distribuida, autónoma y más flexible. Se trata de dotar a la mercancía de inteligencia.

Hasta ahora, se ha venido realizando el control de mercancías por bultos o lotes. Pero también es posible realizar un control más detallado llegando incluso a la unidad.

### **1.2 Puntos de control/supervisión.**

Una de las actividades necesarias para el seguimiento eficiente de las mercancías que se distribuyen es tener conocimiento de su posición. Además de este parámetros imprescindible también están otros como su estado o algunas de sus cualidades.

Por esto es por lo que es necesario desplegar estaciones donde se efectúe la tarea. Lo que llamamos punto de control y supervisión. La inclusión de estas estaciones de seguimiento añade unas cuantas variables a lo que es el problema de la distribución. Encontrar una solución óptima a este problema, no es sencillo y requiere de un estudio que incluya requisitos y otros datos fundamentales. En cualquier caso, este problema no es objeto de este proyecto, pero es posible consultar [este enlace](#)<sup>2</sup> donde es posible encontrar información al respecto.

Lo que sí es necesario conocer es un escenario genérico como el mostrado en la Figura 1 el ciclo del producto. Podemos observar una posible configuración de los puntos de control. La mercancía se fabrica y se almacena normalmente cerca del lugar de fabricación para posteriormente distribuirla. Para ello se va reubicando en distintos almacenes, cada vez más cerca del destino hasta llegar a él. Algunas cadenas de distribución terminan aquí. Alternativamente, la logística puede continuar recogiendo el producto una vez que cumple su función en el cliente para posteriormente reciclar la mercancía o proceder a su destrucción.

---

<sup>2</sup> [http://en.wikipedia.org/wiki/Supply\\_chain\\_management](http://en.wikipedia.org/wiki/Supply_chain_management)

En cada uno de estos pasos es posible incluir un punto de control y supervisión que permita realizar el seguimiento estrecho de la mercancía. Esto se representa en la Figura 1 con un recuadro con las letras PC.

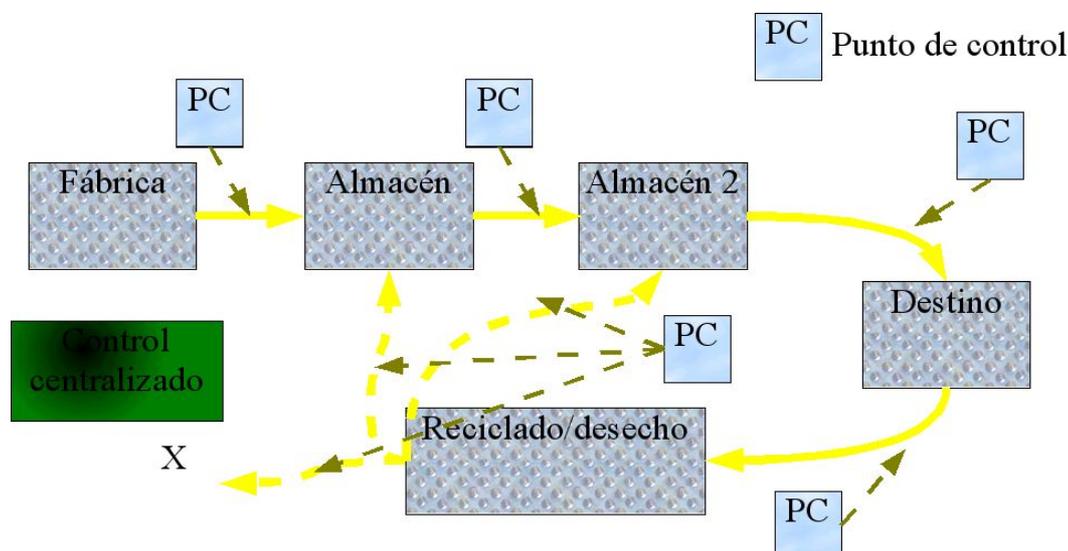


Figura 1: Diagrama genérico puntos de control.

Las flechas gruesas amarillas indican el flujo de la mercancía, mientras que las flechas más delgadas verdes señalan la posible posición de un punto de control.

Además de los puntos de control distribuidos a lo largo del ciclo del producto, para que el sistema tenga una finalidad útil, debe haber un sistema a más alto nivel que sea capaz de recibir la información que proporcionan los puntos de control y que gestione toda esta información. Esto implica desplegar una infraestructura que permita tales operaciones, y cuyo costo no suele ser ni mucho menos despreciable.

Sin embargo, es necesario recalcar que aunque el proceso de los datos se realice de forma centralizada, ahora las unidades a controlar pueden tener cierta independencia respecto de aquel.

En todo momento se debe tener constancia de la posición y estado de la mercancía. Esto se conoce como trazabilidad y es parte fundamental para el control de calidad.

### 1.3 Entornos industriales / Equipos autónomos.

El recorrido de las mercancías es variable según la naturaleza de la misma. También dependerá del uso que se le dé ó de la localización geográfica objetivo de la empresa que desarrolle la actividad.

Durante este recorrido, la mercancía puede atravesar todo tipo de entornos: desde los más severos hasta los más livianos. Es posible que la mercancía esté empaquetada convenientemente para superar estas circunstancias con éxito y de esta forma no se vea perjudicada en su trayecto. Esta preparación es en ocasiones muy costosa.

Los dispositivos de seguimiento en estudio estarán sujetos a las condiciones impuestas a las mercancías. Si requirieran de algún trato especial, su uso se desaprobaría inmediatamente en la mayoría de ocasiones por no ser compatible con los requisitos mínimos exigibles.

En esta situación, los sistemas de control logístico que se desean implantar estarán sujetos a todos estos factores ambientales, que deberán soportar de la mejor forma posible. Entre estos factores hay que tener en cuenta especialmente los siguientes:

- Entornos industriales. Lugares donde los sistemas están expuestos a interferencias, ruido y otros fenómenos radioeléctricos y electromagnéticos.
- Grandes variaciones de temperatura y humedad. El dispositivo puede estar adjunto a cualquier tipo de mercancía. Las hay que necesiten refrigeración o temperaturas constantes y otras a las que esto no les afecte. En este último caso, durante el transporte se pueden dar situaciones de temperaturas variables, muy altas o muy bajas con mayor o menos grado de humedad. Además de la temperatura y humedad no hay que olvidar entornos como los corrosivos o con partículas en suspensión.
- La distancia entre dos puntos intermedios de control puede ser muy grande y realizarse en un tiempo largo, p. ej.: días. Además como mínimo, el dispositivo de control debe tener la suficiente autonomía para ser funcional durante un ciclo de producto. Sin embargo, para que éste sea competitivo es conveniente que su duración se prolongue durante años.
- Los almacenes suelen ser lugares de dimensiones muy grandes. Puede haber distancias incluso de kilómetros. Los dispositivos deben ser funcionales a estas distancias.

#### **1.4 Análisis de los datos.**

Es evidente, que el objetivo de los dispositivos de control adjuntos a la mercancía tienen como finalidad la recogida de datos. Con estos datos, ya es posible llevar a cabo la gestión y el control de los bienes que se deben distribuir.

No hay que olvidar que los dispositivos en estudio precisan de un sistema a un nivel más elevado para poder extraer conclusiones de los datos acopiados. De la calidad y cantidad de estos datos depende el éxito de sistema de control logístico.

Un diseño y planificación adecuadas influye a la automatización del proceso que al fin y al cabo es uno de los objetivos principales junto con la fiabilidad para lograr una calidad ventajosa. Si se utilizan dispositivos inteligentes es más sencillo conseguir una automatización más lograda y fiable.

Los datos obtenidos permiten capacidades de seguimiento remoto de las unidades. Debe haber una capacidad para conseguir una vista unificada del estado de todas las unidades, algo así como un centro de control, al que no necesariamente se debe acceder desde una ubicación física exclusiva, sino que también debería ser posible obtener y visualizar la información desde lugares remotos. Todo esto es posible con las nuevas tecnologías de red.

En cuanto al procesamiento de datos, es posible hacer predicciones y sacar estadísticas con la información obtenida. En este sentido, la información recabada es muy valiosa pues de este modo se contribuye beneficiosamente en la calidad final. De esta forma se puede conocer los puntos débiles en la cadena de distribución, así como los lugares donde el proceso funciona correctamente.

Analizando la información se pueden determinar correcciones como ubicación de puntos intermedios adicionales ó rupturas de stock. Mientras que las estadísticas son útiles a la hora de evaluar la eficiencia del sistema o de los procesos como un todo.

### **1.5 Aprovechamiento de avances tecnológicos.**

Unos de los inconvenientes de las implantaciones dilatadas en el tiempo es que al final de su despliegue, o incluso durante éste, aparecen nuevas tecnologías que comparativamente son mejores que las recién implantadas.

En muchas ocasiones, esto es inevitable por la propia naturaleza del proceso de desarrollo e implantación. Sin embargo, es posible tratar de minimizar este efecto si el sistema implantado permite algún tipo de actualización.

Aún así, es necesario partir de desarrollos tan avanzados como permita la tecnología del momento. Por ejemplo, es muy importante obtener una tecnología de muy bajo consumo que permita cumplir con los requisitos impuestos. La eficiencia energética es fundamental en los sistemas autónomos y aún más si necesitan comunicarse de forma remota.

Este es uno de los avances clave que deben desarrollar estos sistemas. Otra de las tecnologías que proporcionan un beneficio inmediato son las inalámbricas. Este tipo de comunicaciones es, por su naturaleza, muy vulnerable a posibles ataques, bien activos o pasivos para obtener o falsear información. Hay que tener en cuenta las técnicas necesarias para evitar este tipo de ataques.

La capacidad de almacenar información también es importante en estos sistemas. Los datos generados en el proceso de transporte son relevantes para controlar la calidad del resultado final. El dispositivo debe ser capaz de salvar esta información de forma que sea posible recuperarla en el momento adecuado. Éste puede contener, entre otros, producto transportado, datos de posición, histórico de traslado, verificaciones intermedias, histórico de temperatura ó medidas de humedad.

## Capítulo 1:Planteamiento.

Con todas estas capacidades, si el sistema de gestión que actúa por encima del grupo de unidades es adecuado, se puede llegar incluso a que la mercancía no necesite una infraestructura densa para que la cadena de distribución cumpla con su objetivo. Sería necesario tan sólo con un grupo reducido de puntos de control. Esto complementa a la funcionalidad descrita anteriormente como es la de la gestión remota para reducir la infraestructura al mínimo y ejercer un control flexible.

## 2 Estado del arte.

La logística se ha ido desarrollando a lo largo del tiempo. En un principio era parte de la estrategia militar, mientras que ahora su uso se aplica a una disparidad de bienes, servicios o incluso energía.

### 2.1 Sistemas empleados en logística.

Cuando se quiere distribuir algo es indispensable evaluar la cantidad disponible en cada momento y lugar. Esto determina cuando se debe hacer el envío y es posible estimar cuando llegará a su lugar de destino. Para ello las mercancías están etiquetadas, de esta forma se las identifica.

La forma de etiquetado ha ido evolucionando a lo largo del tiempo: desde simple notas en papel, hasta etiquetas electrónicas pasando con pegatinas y códigos de barras. A continuación se detallan algunas de las características principales sobre los más importantes.

#### 2.1.1 Códigos de barras.

El código de barras es la representación de una determinada información mediante un conjunto de líneas paralelas verticales de diferente grosor y espaciado<sup>3</sup>. En este caso el código de barras es unidimensional.

Uno de los códigos de barras unidimensionales que presenta más densidad de información es el llamado código 128 (Code 128), en el que es posible representar una cantidad variable de caracteres ASCII según la longitud del código de barras. Éste además incluye una secuencia de inicio, fin y comprobación. Cada carácter se compone de una secuencia de barra y espacio repetido 3 veces, con anchura de 11 módulos en total y donde el módulo es la unidad de anchura primitiva. [BAR0]

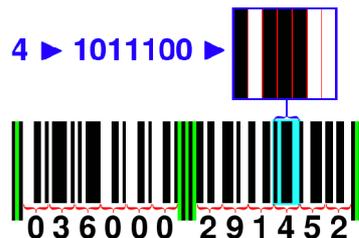


Figura 2: Código de barras UPC



Figura 3: Código de barras EAN 13

Otros ejemplos de códigos de barras unidimensionales son el UPC (Universal Product Code) utilizado en EE.UU y Canadá (Figura 2) ó el EAN-UCC (European Article Numbering-Unifom Code Council) (Figura 3) utilizado en todo el mundo. El UCC requirió a los fabricantes que utilizaban el UPC de 12 dígitos a sustituirlo por el EAN de 13 dígitos para el 1 de Enero de 2005.

3 [http://es.wikipedia.org/wiki/C%C3%B3digo\\_de\\_barras](http://es.wikipedia.org/wiki/C%C3%B3digo_de_barras)

Los códigos de barras unidimensionales pueden resultar insuficientes para representar cantidades de datos más grandes, es decir, para identificar mayor cantidad de elementos. Por esta razón se han implantado los códigos apilados y los bidimensionales, que permiten mayor densidad de información en menor espacio, con la contrapartida de requerir lectores más sofisticados.

El “código 16K” (Code 16K) es un ejemplo de código de barras apilado basado en el Código 128 en el que se pueden apilar desde 2 (Figura 4) a 16 códigos de este tipo. Otros tipos son el Matriz de datos (Data Matrix) que es un código de barras propietario en dos dimensiones capaz de empaquetar mucha información en muy poco espacio. Este código puede almacenar entre 1 y 500 caracteres por símbolo que a su vez puede entre una milésima de pulgada cuadrada hasta una pulgada cuadrada lo que da una densidad de información teórica de 500 millones de caracteres por pulgada cuadrada. Esta densidad en la práctica se ve limitada por la precisión de la tecnología utilizada para la impresión y lectura.



Figura 4: Código 16K



Figura 5: Código maxicode



Figura 6: Código PDF417



Figura 7: Código QR

Otros ejemplos de códigos bidimensionales son el Maxicode (propietario de UPS) en el que se utilizan puntos hexagonales en lugar de cuadrados (Figura 5) u otros de dominio público como el PDF417 (Figura 6) ó el código QR (Figura 7).

Para leer la información que proporcionan estos códigos de barras, es preciso contar con un lector de código de barras (Figura 8). Normalmente, estos lectores están basados en una fuente de luz, una lente y un fotoconductor. Con esto se obtiene una información que se analiza y se obtienen los datos del código. [WPO].



Figura 8: Lector de código de barras

Para más información sobre los códigos de barras consultar [BAR1] y [BAR2].

### **2.1.2 Código electrónico de producto (EPC).**

El código electrónico del producto (EPC: Electronic Product Code) es una etiqueta electrónica que nació para mejorar las características del código de barras UPC. El EPC es un número de 96 bits que identifica a un producto, el GTIN (Global Trade Identification Number) ó Número de Identificación Comercial Internacional.

Su principal diferencia respecto al número UPC es que además de identificar el producto en un grupo (como el UPC) proporciona un identificador al grupo, proporcionando mayor precisión en el seguimiento.

El GTIN es un término para describir la familia de estructuras de datos EAN/UCC en elementos comerciales, productos y servicios. Tienen una longitud de 14 caracteres y está compuesta por: ceros hasta completar los 14 dígitos; prefijos de país y compañía; elemento y dígito de comprobación.

El EPC se utiliza en los sistemas de identificación por radiofrecuencia (RFID) que se tratan en la sección 2.2.

### **2.1.3 Sistema de información EPC.**

El sistema de información de EPC ó EPCIS es un estándar para el intercambio de datos provenientes de EPCs a nivel de empresas, permitiendo integrar los datos obtenidos de los productos propios, a través de sus EPCs, con las bases de datos de otras empresas involucradas en la misma cadena de suministro. [UOC]

El diagrama de la Figura 9 muestra el esquema de un EPCIS. En él podemos observar como se relacionan las distintas partes de sistema de información. En primer lugar las etiquetas y sensores a nivel de campo proporcionan información de forma inalámbrica al lector de datos que a su vez los envía a una capa de middleware que se encarga de adecuarlos para su procesamiento por parte de otras aplicaciones.

El middleware es fundamental para el intercambio de datos entre los dispositivos y entidades como el EPCIS, externo a la compañía ó el WMS<sup>4</sup> y el ERP<sup>5</sup>, estos dos últimos muy ligados a la empresa.

Posteriormente, la aplicación almacena la información en una base de datos para poder hacer consultas por parte de otro u otros agentes. Uno de ellos es el EPCIS, que proporciona un canal de datos para consultas a esta base de datos. Las consultas pueden resolver la localización de un determinado elemento enviado por una capa superior, como la de enumeración.

---

4 WMS: Warehouse Management System ó Sistema de gestión de almacén.

5 ERP: Enterprise Resource Management o Gestor de recurso de empresa.

La capa de enumeración es capaz de comprender peticiones de nombre dando como resultado un localizador o para otros propósitos es posible responder con un puntero con el que modificar datos de la base de datos.

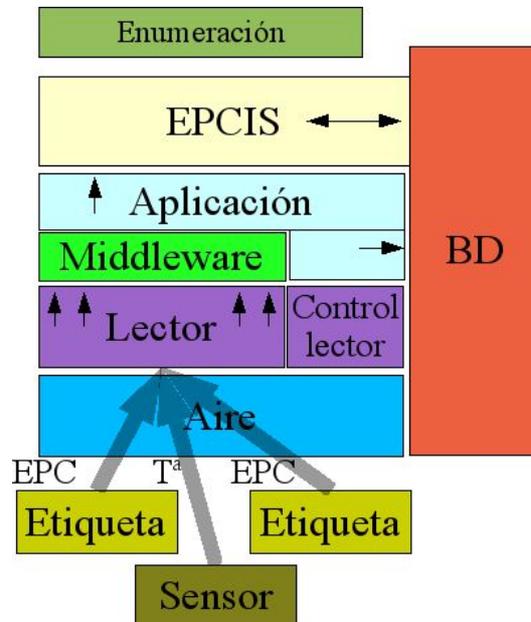


Figura 9: Pila EPCIS.

La base de datos sobre la que se sustenta este tipo de sistemas contiene información sobre los elementos a controlar. En ella se almacenan los datos recolectados y otros extraídos a partir de los mismos.

## 2.2 Etiquetas RFID.

Las siglas RFID significan RadioFrequency Identification ó Identificación de radiofrecuencia., que es un método de almacenar y obtener información remotamente utilizando unos dispositivos llamados etiquetas de radiofrecuencia, RFID tags o transpondedores.

Una etiqueta de radiofrecuencia es un objeto reducido que se puede solapar o incorporar a un elemento a controlar: móvil, estático, inerte o vivo. Las etiquetas de radiofrecuencia incluyen antenas que los habilitan para recibir y responder a peticiones por radiofrecuencia de un transceptor. También incluyen transductor inalámbrico y un encapsulado.

El transceptor ó lector de etiquetas consta de una antena, un transceptor y un decodificador que envía señales periódicas para que las etiquetas próximas respondan. Una vez que obtiene respuesta la debe trasladar a la capa superior para que ésta las gestione.

Su comodidad a la hora de la lectura y su capacidad para intercambiar más y mejor información hacen que las etiquetas de radiofrecuencia sean una de las opciones para el control de mercancías.

### ***Middleware.***

Existe un componente importante que se ubica entre los dispositivos de etiquetamiento y las aplicaciones de nivel superior. Éste es el middleware que realiza una primera gestión de la información recibida.

Las funciones que cumple esta capa de software son el suavizado de datos, la deduplicación de los datos y el envío de datos.

El suavizado de datos gestiona las lecturas o lecturas incorrectas de dispositivos por parte del lector. La deduplicación evita las lecturas duplicadas en zonas de solapamiento por parte de dos lectores distintos. Por último, el envío de datos ayuda a reducir el tráfico en la red transmitiendo datos sólo cuando hay algún cambio en los lectores seleccionados en la aplicación.

#### ***2.2.1 Historia.***

La identificación por radiofrecuencia empezó como la combinación de dos tecnologías: radar y radiodifusión(radio broadcast). El radar era ya un hecho consumado en la 2ª Guerra Mundial y la identificación mediante técnicas radar fueron utilizadas por primera vez por los Ingleses en la 2ª Guerra Mundial para distinguir sus propios aviones del resto.

Un estudio temprano de lo que podría llegar a ser el RFID se escribió en 1948 ("Communication by Means of Reflected Power", por Harry Stockman). En él se concluía que todavía era necesario realizar muchos esfuerzos para que esta tecnología fuera aplicable en la práctica. Después de 30 años, la llegada de los dispositivos electrónicos supuso un importante avance en este sentido. Este fue un intervalo de exploración de las tecnologías radar y radio.

En los años 60, se obtuvieron realizaciones de diversos dispositivos que supusieron la antesala a los dispositivos comerciales en los años 70. Eran dispositivos primarios, pero a su vez útiles que solían transmitir la información equivalente a 1bit para indicar presencia o ausencia con técnicas inductivas o de microondas. Éstos eran dispositivos baratos y de ahí su rápida extensión.

A mediados de los 70 ya había algunas instituciones y laboratorios de investigación tratando de mejorar los dispositivos conocidos hasta que a finales de los 70 algunas empresas lanzaron dispositivos más avanzados como sistemas de identificación electrónica, transpondedores ó codificadores por microondas.

Durante los 80, se habían puesto en marcha la implementación total de la tecnología, la prueba es que empezaron a surgir empresas que trabajaban estos sistemas. Los intereses se centraban en dispositivos para animales, industriales, aplicaciones para empresas y peajes en autopistas en Europa. Pero también surgió interés para aplicaciones como transporte, o puertos de mar.

En los 90, se popularizaron aplicaciones como el telepeaje, control de acceso. Pero el logro más significativo de esta época fue lograr una mayor integración, con lo que se obtenían dispositivos completos integrados en un chip, esto permitía ser el candidato ideal del código de barras.[LANDT01]

*Tabla 1: Historia y evolución tecnología RFID.*

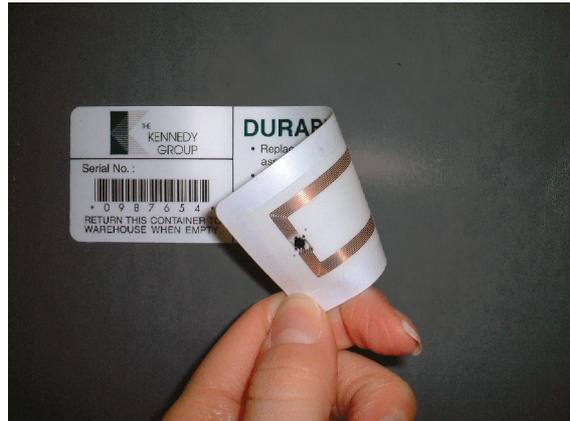
<i><b>Década</b></i>	<i><b>Hecho</b></i>
1940-1950	Perfeccionamiento y uso del radar. Principal desarrollo en la 2º Guerra Mundial. Invención del RFID en 1948.
1950-1960	Primeras exploraciones de la tecnología RFID. Experimentos de laboratorio.
1960-1970	Desarrollo de la teoría de RFID. Primeras pruebas de campo con aplicaciones.
1970-1980	Eclosión del desarrollo RFID. La pruebas sobre RFID se multiplican. Aplicaciones tempranas de RFID.
1980-1990	Aplicaciones comerciales de RFID.
1990-2000	Aparece la normalización. Uso RFID muy extendido. Utilización cotidiana de RFID.

### **2.2.2 Tipos.**

Dentro de los dispositivos RFID nos encontramos varios tipos[CH05]. Si los clasificamos según la fuente de energía utilizada en pasivos, semi-pasivos y activos. También es posible su clasificación según la frecuencia utilizada para la comunicación.

***Según la fuente de energía.***

Los **pasivos** no poseen ninguna fuente de energía interna, por lo que su funcionamiento se ha de basar en la energía transmitida desde fuentes externos. La pequeña corriente eléctrica inducida en la antena por las señales de radiofrecuencia emitidas por el lector suministran energía suficiente para que el dispositivo transmita una respuesta. Pero este funcionamiento limita la longitud de la respuesta de la etiqueta a sólo un número de identificación. Por otro lado, esto también permite que el dispositivo sea muy pequeño, ya que no necesita una batería para operar. Para leer estos dispositivos hace falta un equipo lector cuyo rango abarca desde los 10mm hasta los 6 metros.



*Figura 10: Etiqueta RFID pasiva.*

Las **etiquetas activas** por contra, presentan una batería interna lo que les permite un mayor rango de funcionamiento y posibilitan la transmisión de cantidades más grandes de datos. Estos dispositivos son algo más grandes que los pasivos, su batería tiene una duración que ronda los 10 años y permiten distancias de decenas de metros.



*Figura 11: Etiqueta RFID activa.*

Las **etiquetas semi-pasivas** si bien utilizan las mismas técnicas de transmisión y recepción de radiofrecuencia que las etiquetas activas, incluyen una batería interna y otra circuitería para efectuar otras tareas relacionadas con la aplicación que tiene, por ejemplo medición de temperatura. En resumen, estos dispositivos son una combinación de las características de comunicación de las etiquetas pasivas utilizando el procesamiento de una etiqueta activa.

Aunque no están dentro de la categoría RFID, ,mención especial merecen las tarjetas inteligentes sin contacto ó “*contactless smartcard*”. Estos dispositivos incluyen un microcontrolador ó una memoria embutido dentro de un plástico plano. La forma de comunicación es inalámbrica a partir de tecnología de autoalimentación por inducción. El alcance es muy limitado, de tan sólo 10cm. (Figura 12).

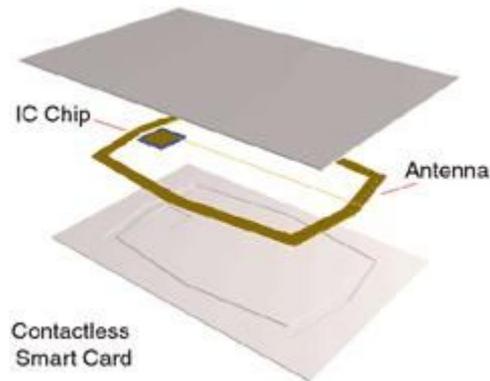


Figura 12: Etiqueta inteligente sin contactos.

### **Según banda de frecuencia de funcionamiento.**

Los dispositivos de etiquetado de radiofrecuencia funcionan principalmente en las siguientes bandas[RFJO]:

- ◆ Baja frecuencia. 134,2KHz según estándar internacional y 125KHz para EE.UU. Su alcance es muy corto y su tasas de transferencia también.
- ◆ Alta frecuencia. 13,54MHz. Alcance mayor que en el caso anterior con tasa de transferencia algo mayores.
- ◆ Ultra alta frecuencia. Desde 850MHz hasta 950MHz. Tasa de transferencia media y alcance mayor que los anteriores. También utilizan más potencia que los anteriores, sin embargo también algo más caros.
- ◆ Microondas. 5,8GHz. Tasas de transferencia muy altas. Alcance de metros, pero son caros y consumen mucha energía.

### **2.3 Comparación RFID / código de barras.**

Parece claro que las etiquetas de radiofrecuencia RFID son competencia directa del muy extendido y bien conocido código de barras. En este punto, es interesante detenerse para analizar las ventajas y los inconvenientes de cada uno de ellos de forma que podamos deducir las aplicaciones más adecuadas para cada método.

Hoy por hoy, el código de barras está presente en la practica totalidad de los sistemas logísticos y en general todo tipo de artículos. Sin embargo la tendencia está cambiando en favor de la etiqueta electrónica.

### ***Precisión en la lectura.***

Empezamos observando la precisión de la lectura en cada caso. Mientras que el código de barras de lectura óptica, la etiqueta sigue mecanismos electrónicos para cumplir su función. Esto implica que la lectura del código de barras siempre es posible dentro del rango de funcionamiento, mientras que en el caso de la etiqueta RFID esto no siempre es posible si los objetos tienen grandes cantidades de agua o metal.

Por contra, el RFID permite una mejor automatización de los procesos. Esto se debe a que para leer el código, el elemento no tiene que estar en una posición concreta, mientras que con el código de barras esta posición era fija e invariable. Esto confiere a la etiqueta de radiofrecuencia una tasa de lecturas por unidad de tiempo mayor que el código de barras, y no sólo eso, si no que ahora es posible leer elementos que antes no podían ser leídos, por ejemplo, aquellos cuya envoltura no lo permitía.

### ***Latencia en la lectura.***

La lectura de el código de barras precisa en la mayoría de los casos de la intervención humana, salvo excepciones poco habituales. Esto hace que el proceso de lectura incluya una latencia irreductible. Por contra, parece claro que esta latencia no se presenta de la misma forma en las lecturas de RFID.

### ***Especificidad de los datos.***

La RFID puede incluir datos con gran riqueza de detalles, por ejemplo, instancia de un objeto, temperatura, posición, etc. Esta granularidad no es posible alcanzarla con el código de barras. Por este motivo, un elemento que se lee dos veces con el código de barras computa como dos elementos, mientras que con la RFID, sólo computa una.

Esto se debe a que con el código de barras la cantidad de información que se puede leer está limitada, mientras que con la etiqueta electrónica la cantidad de información suministrada permite la distinción hasta el nivel de unidades dentro de una misma clase, algo imposible con el código de barras.

## **2.4 Etiquetas activas.**

Las tecnología RFID en su versión activa como se ha visto tiene unas propiedades y unas ventajas a las que se le puede sacar provecho. Son estos sistemas los que se estudian por su potencial. Con ellas se puede conseguir mejorar la calidad de los procesos logísticos.

Pero no todo son ventajas, de hecho son sistemas menos popularizados principalmente por su precio, aún teniendo muchas más prestaciones que sus análogos pasivos. Al no ser sistemas con tanto nivel de utilización, los procesos de fabricación no gozan las ventajas de economía de escala y esto cierra un círculo que retrae su expansión.

La única forma de que su uso se extienda es que se muestren lo suficientemente ventajosos como para que su implantación reduzca costes o mejore la calidad de todo el proceso.

En principio sus ventajas son claras. Las etiquetas activas son capaces de almacenar más datos. Al poder contar con una fuente de energía el posible contar con dispositivos de retención de datos. Su capacidad dependerá del espacio disponible para este fin.

La distancia para la que la comunicación es posible es mayor, además las comunicaciones son más fiables. Si no se cuenta con limitaciones de energía es posible utilizar circuitería de comunicaciones más compleja, con amplificadores y filtros. Las comunicaciones son más fiables porque la sensibilidad puede ser mejor que en el caso de las etiquetas pasivas y pueden incluir datos para detección y corrección de errores.

Es posible añadir algún tipo de circuitería de control y procesamiento con lo que las características pueden sufrir un cambio cualitativo. Se puede mejorar la independencia del dispositivos y darle usos más amplios.

Pero como se ha dicho no todo son ventajas. El incremento de precio y de tamaño son inconvenientes que pueden imposibilitar que esta tecnología se acabe imponiendo.



Figura 13: RFID activa y lector.

### 3 Estudio preliminar.

Hasta ahora se ha analizado el problema que se quiere resolver. Dando a conocer sus causas y el estado actual de las soluciones que en la actualidad se han adoptado. En este capítulo se plantea como objetivo del Proyecto el camino para mejorar las soluciones que hasta ahora existen.

#### 3.1 Delimitación objetivo.

Se ha observado que el logístico es un sector con mucho potencial, se pone el interés en los sistemas que rodean al control de mercancías. En la actualidad, existen unos pocos sistemas que se encargan de esta tarea y han surgido recientemente la tecnología RFID con etiquetas activas. Éstas pueden tener grandes prestaciones, pero no existe en el mercado ninguna solución genérica que explote las posibilidades de estos sistemas.

El objetivo de este Proyecto es el diseño y realización de una plataforma hardware que permita la simulación y prueba de protocolos de comunicación para dispositivos de etiquetado electrónico. De esta forma es posible conocer y ensayar las características más importantes para cada tipo de aplicación.

La plataforma incluirá los medios de comunicación más convenientes para esta tarea, así como interfaces que permitan la interacción con el usuario y sensores para obtener datos del entorno de la plataforma.

#### 3.2 Requisitos y arquitectura.

En este apartado se abordan los requisitos necesarios para el diseño de la plataforma. Los requisitos son las características más importantes que debe presentar la plataforma e incluyen la arquitectura como base fundamental de toda la plataforma, pero no hay que olvidar otros como los dispositivos de comunicación o los periféricos.

Para que el abanico de aplicaciones sea lo más amplio posible se establece que esta debe ser una plataforma **abierta y genérica**. Esto significa que no los requisitos para comenzar el desarrollo serán los mínimos posibles. No debe depender de otras tecnologías, especialmente de aquellas con patentes que dificulten el desarrollo con las mismas. Sin embargo, la plataforma no debe impedirlo en este tipo de tecnologías.

La plataforma deberá utilizar preferentemente **componentes bien conocidos**. Si un componente tiene un recorrido largo y conocido, será más fácil aprovechar experiencias anteriores para llevar a cabo el desarrollo además evitar posibles errores de desarrollo y de tener la posibilidad de resolver problemas potenciales con más celeridad. Todo esto lleva a sistemas con experiencias satisfactorias para otras aplicaciones.

Por otro lado una vez que un componente está extendido su precio también se ve reducido. Esta es una forma de que la fabricación de un prototipo tenga un **precio contenido**.

Por las características de la aplicación, es necesario que el dispositivo tenga un autonomía significativa, lo que lleva a **reducir el consumo de energía**, de esta forma se alargará la vida útil del dispositivo y la aplicación estará menos restringida.

El consumo de energía es directamente proporcional a la carga computacional que se efectúa. Si se quiere reducir el consumo energético es necesario limitar la cantidad de procesamiento. Afortunadamente, la aplicación a la que se desea llegar **no requiere un procesamiento intensivo**. Es más, si fuera necesario llevar a cabo tal cantidad de cálculo, se podría posponer de forma que fuera realizado externamente por un sistema de más alto nivel.

Se han mencionado algunos de los entornos a los que deben enfrentarse estos sistemas: ruidosos, con interferencias, etc... Ante esto la plataforma debe ser **robusta y fiable**. El sistema debe ser capaz de restablecerse en caso de funcionamiento anómalo, pero tampoco es posible permitirse la pérdida de datos.

La proyectada debe ser una plataforma que soporte el desarrollo en multitud de escenarios. Esto implica **versatilidad**: se deben proporcionar diversos periféricos y mecanismo con los que se puedan cubrir la mayoría de situaciones a probar y simular. Es necesario contar con un dispositivo reprogramable.

Otro de los requisitos que se imponen es la posibilidad de utilizar **seguridad en las comunicaciones**. Esto es necesario, ya que algunas aplicaciones requerirán confidencialidad en las comunicaciones.

Teniendo todas estos requisitos en cuenta se opta por una plataforma basada en un **microcontrolador**.

### **3.3 Comunicaciones. Periféricos.**

Como es lógico para este tipo de aplicaciones, la plataforma debe comunicarse con otros dispositivos. También es posible la comunicación entre dos plataformas. Además, debe haber una forma de comunicación fiable y sencilla, por ejemplo con el entorno de desarrollo.

Otras interfaces necesarias son las de programación y la del usuario. Los sensores completan el apartado de periféricos.

#### **3.3.1 Radiofrecuencia.**

Para conseguir un sistema que se comunique con otros de forma inalámbricas es necesario que ésta implemente alguna de las siguientes tecnologías: radiofrecuencia, infrarrojos, microondas.

Las *microondas* son ondas electromagnéticas con frecuencias superiores al gigahercio. Dentro de este espacio radioeléctrico, existen dos zonas de uso sin licencia, el resto está regulado, con lo que habría que obtener una licencia para utilizarlo. Por otro lado, los dispositivos que funcionan a estas frecuencias suelen estar enfocados a aplicaciones concretas, con lo que sería necesario desarrollar una etapa de propósito específico. Esto sería costoso, difícil de probar y entrañaría algunos riesgos, más aún si no se tiene experiencia previa.

Las frecuencia para las que no es necesario licencia son 2,45GHz y 5,8GHz. Sería viable utilizar estas frecuencias como hacen la tecnología Wi-Fi, Bluetooth ó Wi-Max, pues existen dispositivos comerciales que trabajan en estas bandas. El uso de éstas se descarta por dos razones. Una de ellas es el coste de estos dispositivos, ya que estas bandas se utilizan para comunicaciones de más altas prestaciones que la requeridas, donde los precios están en otro orden. Además existen inconvenientes técnicos, como que la potencia requerida para estas transmisiones va en contra del requisito de máxima eficiencia energética.

La *tecnología infrarroja* está más presente en dispositivos de propósito general, con lo que sería más barata. Sin embargo el alcance de este tipo de comunicación está muy limitado, además debe haber visión directa. Por esta razón su uso está muy limitado.

Por último tenemos la *radiofrecuencia*, una parte del espectro que se extiende desde los 300KHz hasta los 3GHz. Esta parte del espectro, es la más interesante por contar varios tramos de uso libre, sin licencia, si bien con alguna restricciones en cuanto a potencia transmitida. Estas bandas son las siguientes:

Tabla 2: Comparativa transmisión en RF.

	<i>Rango de frecuencias</i>	<i>Alcance</i>	<i>Tasa de transferencia</i>	<i>Coste</i>
Baja frecuencia	134,2 KHz	Corto	Muy baja	Bajo
Alta frecuencia	13'56 MHz	Medio	Baja	Medio
Ultra alta frecuencia (UHF)	433MHz 860–950 MHz	Largo	Media	Medio

Se busca un compromiso entre precio y factores como la velocidad de transferencia, teniendo en cuenta que cuanto mayor sea ésta, mayor es la potencia necesario para transmitir y menor será el alcance. Se decide por tanto utilizar la banda de las ultra altas frecuencias dentro de la **radiofrecuencia** y dentro de éstas la banda sin licencia en Europa de 868MHZ.

Esta selección tiene dos componentes: el primero es que la banda situada entorno a los 433MHz. se viene utilizando desde hace más tiempo y, por tanto está más saturada (mandos a distancia); la segunda razón es que la banda situada en torno a los 930MHz. tiene restricciones de uso para uso en países como Francia, por esta razón sólo es utilizable en EE.UU. sin licencia.

### ***3.3.2 Conexión cableada.***

Aunque con una etapa de radiofrecuencia se resuelve el problema de la comunicación con otros dispositivos, los requisitos sobre los que se basa la plataforma, aún no. Al ser esta una plataforma para desarrollo, es necesario que cuente con otras formas de comunicación que la hagan más versátil.

Este requisito se cumple con la inclusión de algún tipo de conexión cableada y sencilla, de forma que sea más inmediata y robusta la comunicación externa con la plataforma. Este es el caso de la conexión al entorno de desarrollo.

### ***3.3.3 Interfaz de programación.***

Está claro que para efectuar un desarrollo fluido y continuo es necesario hacer modificaciones sobre el código que ejecuta la plataforma. Se llega a la conclusión de que debe ser posible modificar el código una vez que se tiene el sistema terminado. Para conseguir esto existen dos vías:

- ◆ Modificar el código sobre el propio microcontrolador.
- ◆ Modificar el código almacenado en un dispositivo externo al microcontrolador.

En el segundo caso, sería posible utilizar algún tipo de dispositivo extraíble que contuviera el código. Esto sería flexible, pero encarecería el resultado final y aumentaría la complejidad del diseño.

Por esta razón hay que prever que, contando con un dispositivo reprogramable embebido en la plataforma, se debe incluir un puerto para su programación.

### ***3.3.4 Interfaz de usuario.***

En algunas aplicaciones el usuario debe interactuar con la plataforma directamente. Para estos casos la plataforma dispondrá de una interfaz adecuada.

### ***3.3.5 Sensor.***

Como ya se ha mencionado, para dar un valor añadido a la plataforma es interesante que la misma pueda proporcionar datos sobre su entorno. Para estos casos es conveniente incluir algún tipo de sensor que los obtenga y los traslade al bloque de procesamiento.

### ***3.3.6 Diagrama de bloques.***

En la Figura 14 se puede encontrar una representación del diagrama de bloques de la plataforma. En ella se puede contemplar las principales características tratadas hasta ahora: microcontrolador, etapa de radiofrecuencia, comunicación serie, interfaz de programación, de usuario y sensor(es).

## Plataforma

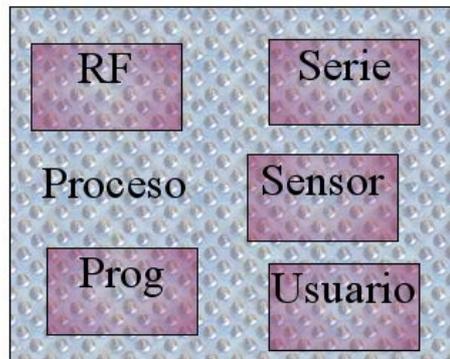


Figura 14: Diagrama de bloques de la plataforma.

### 3.4 Fases del desarrollo. Herramientas.

Como todo proyecto, Éste debe atravesar ciertas fases hasta llegar a un resultado. Para completar cada una de la fases menudo es necesario utilizar una herramienta. Una vez definidas la directrices generales y los requisitos, el Proyecto debe atravesar las fases indicadas en la enumeración siguiente, donde se incluye, en su caso, la herramienta con la que se debe trabajar:

1. Diagrama eléctrico(esquemático): en esta fase se determinan los componentes electrónicos a utilizar así como la conexión entre ellos, es decir, el esquemático. Los componentes y las conexiones se trata como unidades lógicas, no se tratan sus medidas reales. Utiliza una herramienta llamada editor de esquemáticos.
2. Diseño PCB: se utilizan las medidas reales de los componentes y se trata de hacer una conexión física entre ellos siguiendo ciertas normas eléctricas y electrónicas. Como resultado se obtienen los planos de una placa de circuito impreso (PCB, Printed Circuit Board). Su necesita una herramienta de *layout*<sup>6</sup> y *routing*<sup>7</sup>.
3. Desarrollo del software/firmware: Para comprobar el correcto funcionamiento de la PCB diseñada se utiliza un firmware que luego se incluirá en la plataforma y opcionalmente un software que se incluye en algún ordenador externo con el fin de probar la plataforma. Para esto es necesario un ensamblador<sup>8</sup> ó compilador<sup>9</sup> y un enlazador<sup>10</sup> para microcontrolador y para la el ordenador de prueba. Se utilizarán herramientas diseñadas con software libre en la medida de lo posible.

6 Layout: disposición de los componentes electrónico en los planos de montaje de una placa de circuito impreso(PCB).

7 Routing: encaminamiento de las señales eléctricas para conectar componentes en los planos habilitados para ello.

8 Ensamblador: aplicación software que convierte código en ensamblador a objetos de código binario para la arquitectura destino.

9 Compilador: aplicación software que convierte el códigos de medio o alto nivel (C, Java) a objetos código binario para la plataforma destino.

10 Enlazador: aplicación software que une los objetos binarios obtenidos por un ensamblador ó un compilador y crea un objeto binario que se puede ejecutar en la plataforma destino.

4. Fabricación PCB: una vez se tiene el diseño físico del PCB, se procede a su fabricación. Normalmente este paso se encarga a alguna entidad especializada en ello con la infraestructura necesaria para llevarlo a cabo, por esta razón este proceso sólo se explicará someramente en este Proyecto fin de carrera.
5. Montaje PCB: la placa resultante del proceso anterior no tiene los componentes montados para que sea útil. En esta fase se montan con técnicas de soldadura. Esta fase también incluye el acopio de materiales. Para llevarla a cabo son necesarias soldador y máquinas especializadas para el montaje de circuitos integrados.
6. Programador de microcontrolador. Cuando se tiene la realización física de la PCB y también el firmware que se va a utilizar en el microcontrolador de la plataforma, es necesario que éste lo reciba por algún método. Para ello se utiliza un dispositivo llamado programador que se conecta a alguno de los puertos del microcontrolador y procede a la descarga del firmware.
7. Herramientas adicionales. Con todos los pasos anteriores completados ya se puede utilizar la plataforma diseñada. Son necesarias una serie de herramientas adicionales como fuente de alimentación, osciloscopio, polímetro, cables y para este caso en el que se utiliza una puerto de radiofrecuencia, una antena recomendada por el fabricante:

### **3.5 Firmware / software asociado.**

Ante todo vamos a distinguir entre firmware y software:

- ◆ El firmware tiene una relación estrecha con el hardware, normalmente va incluido en el mismo. Muy utilizado en microcontroladores.
- ◆ El software se suele ejecutar sobre un sistema operativo, pues no gestiona el hardware directamente. Por este motivo, el software se ejecuta en sistemas más potentes donde las prestaciones son mayores.

Parece obvio que el microcontrolador debe incluir firmware, mientras que el ordenador utilizado para comunicarse con la plataforma debe utilizar un software apropiado para ello.

En este Proyecto sólo se desarrollará o utilizará aquel firmware ó software necesario para comprobar el correcto funcionamiento de la plataforma.

## **4 Solución/implementación hardware.**

En este apartado se va a dar cuenta de como se materializan los requisitos formulados en el capítulo 3.

### **4.1 Selección plataforma de desarrollo.**

Dados los requisitos especificados, nos centramos en los distintos aspectos en los que hay que tomar una decisión.

#### ***4.1.1 Microcontrolador.***

La necesidad de una plataforma fiable, con bajo consumo y sin necesidad de procesamiento intensivo hace considerar un arquitectura de microcontrolador de 8 bit que es lo suficiente simple y potente como para satisfacer estos requisitos. En concreto, nos centramos en la arquitectura 8051 bien conocida y experimentada.

Existen una gran variedad de fabricantes en el mercado que utilizan arquitecturas de 8 bit para múltiples finalidades. Esta amplia oferta facilita conseguir algún modelo que cumpla con las especificaciones requeridas. La oferta varía desde las familias más simples hasta otras con multitud de periféricos o memoria.

Es fundamental que el dispositivo disponga de una etapa de radiofrecuencia, de hecho es uno de los requisitos más importantes. Además de esto también sería deseable que el microcontrolador disponga de algún puerto serie. Su ausencia no sería determinante, pero haría falta en este caso conseguir este funcionamiento por firmware, una solución posible pero algo engorrosa.

Como se ha establecido, el microcontrolador ejecutará el código desde la memoria de programa, interna en el mismo. Además esta memoria es regrabable, lo más habitual es que sea de tipo flash. La cantidad necesaria depende de la aplicación que se quiere programar, pero lo más correcto, para cubrir la mayor cantidad de casos, es que sea lo mayor posible. Es aconsejable que sea de por lo menos 16KB. La ejecución de cualquier firmware requiere también de memoria RAM, en la practica totalidad de microcontroladores comerciales, ésta está integrada en el mismo en cantidad suficiente, por lo que esto no debe ser problemático.

Los datos analógicos recogidos por sensores externos pueden ser adquiridos por el microcontrolador directamente gracias a una entrada analógicas o bien por un convertidor analógico/digital externo. También puede ocurrir que el sensor proporcione directamente una señal digital.

En la actualidad, estos microcontroladores tienen funciones que propician el bajo consumo, deshabilitando funcionalidades mientras no se necesitan o incluso durmiendo al dispositivo completo.

Explorando el mercado, hallamos el microcontrolador CC1010 de Chipcon (Figura 15), con las siguientes características:

- ◆ Arquitectura 8051. Núcleo a 2'5 veces su velocidad.
- ◆ Etapa de radiofrecuencia utilizable en las frecuencias 300-1000MHz. Tasa máxima de transmisión de 76,8Kbps. Bajo consumo. Posibilidad de transmisión con saltos en frecuencia.
- ◆ 32KB de memoria flash. 2KB de memoria RAM.
- ◆ 2 puertos series.
- ◆ 3 entradas analógicas.
- ◆ Mecanismos de cifrado DES internos.
- ◆ Tensión de funcionamiento: 2,7V a 3,6 V.



Figura 15: Imagen del CC1010.

#### 4.1.2 Plataforma.

El microcontrolador es el “cerebro” de la plataforma. Controla todos los dispositivos a su alrededor e interactúa con otros sistemas externos. Una vez se ha seleccionado el mismo continuamos con el resto de componentes de la plataforma.

Para la interfaz humana se han seleccionado un visualizador LCD estándar y un teclado numérico de membrana. Los LCD estándar tienen un bus de 8 líneas para datos y datos de control y 3 líneas de control exclusivamente. El teclado es una matriz de líneas (4x3) de la que se extraen 7 líneas con las que se pueden leer el estado de las 12 teclas.

La alimentación de la plataforma será externa por medio de una entrada de alimentación. Utilizará corriente continua donde una etapa de alimentación la convertirá en las tensiones adecuadas para el funcionamiento de la plataforma.

La programación se efectuará a través de un puerto paralelo incluido en la plataforma y que se conectará a un PC que será el que efectúe la descarga del firmware.

Todo esto se puede observar de forma esquemática en la Figura 16.

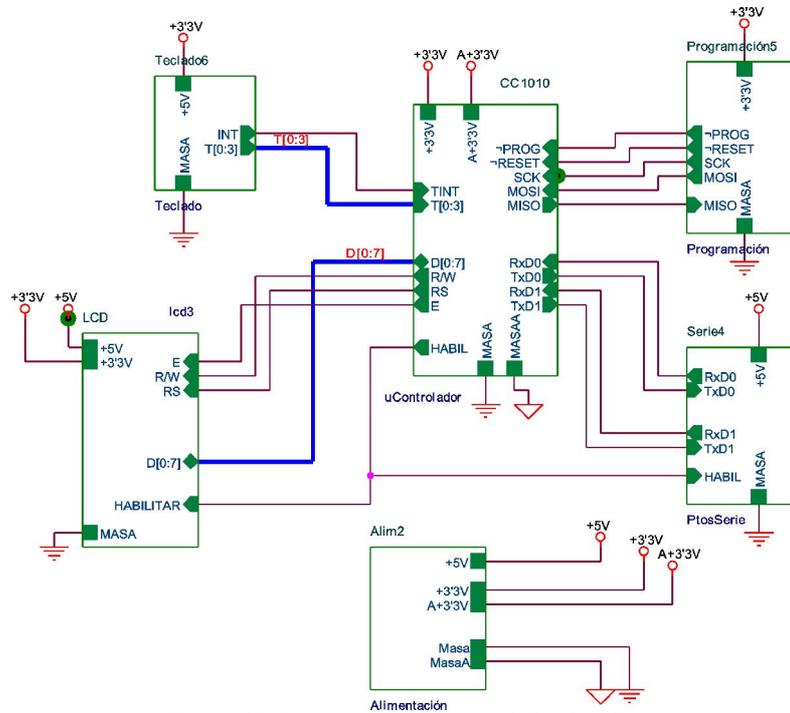


Figura 16: Diagrama del bloques de la plataforma.

## 4.2 Otros componentes importantes. Consideraciones de diseño.

Se dan a continuación los detalles más importantes a la hora de realizar el diseño. Todos ellos se pueden observar con detalle en el apéndice A.

### 4.2.1 Sensor de temperatura.

El sensor de temperatura es el LM61 de National Instruments. Es analógico y funciona en el rango  $-30^{\circ}\text{C}$  hasta  $100^{\circ}\text{C}$  con una precisión de  $\pm 3^{\circ}$  a temperatura ambiente. Se conecta a una de las entradas analógicas del microcontrolador.

### 4.2.2 Alimentación.

La alimentación llega a la placa a través de un jack al que se conecta una fuente de alimentación que tiene como salida una tensión continua mayor de 7 voltios y menor de 14 voltios.

La placa utiliza circuitería con dos tensiones de alimentación distintas. Una a 5 voltios y otra a 3,3 voltios. Para conseguir estas tensiones en primer lugar se hace una regulación lineal para conseguir una tensión estable a 5V. Se ha decidido utilizar un regulador lineal porque en esta placa la eficiencia de la etapa de alimentación no es crítico y además los reguladores conmutados, aún siendo más eficientes energéticamente, producen más ruido y pueden contribuir a la interferencia electromagnética.

Así se ha optado por el regulador L4940V5 de ST, de baja caída de tensión y una corriente máxima de salida de 1,5A más que suficiente para esta aplicación (ver sección 4.3). El encapsulado es el conocido como D<sup>2</sup>PAK que permite una buena disipación del calor para un dispositivo SMD<sup>11</sup>. La salida se permanecerá en un rango de  $5V \pm 0,2V$  con un rizado máximo de 30mV.

A continuación se añaden dos reguladores lineales para salida de 3,3V. En este caso además hay que tener en cuenta que los reguladores lineales son la única opción, ya que los dispositivos conectados a esta tensión requieren niveles precisos y constantes, con pocas fluctuaciones. Para este caso se ha elegido el modelo MC33269 de Motorola que proporciona una corriente máxima de salida de 800mA (ver sección 4.3) con una caída muy baja de tensión en el dispositivo. Para la versión de 3,3V la tensión se mantiene en  $3,3V \pm 70mV$  con un rizado máximo del 0,5%.

La duplicidad de reguladores a 3,3V es indispensable al utilizarse en la placa alimentaciones analógicas y otras digitales. Si se hubiera utilizado un sólo regulador, se hubieran producido ruidos indeseables en las líneas de alimentación con el consiguiente riesgo de mal funcionamiento de los dispositivos.

No hay que olvidar que el correcto funcionamiento de los reguladores requiere de condensadores de estabilización tanto a la entrada como a la salida del dispositivo.

Los dispositivos mencionados en este apartado se han protegido convenientemente con la ayuda de diodos para evitar retornos de corriente. Además de otro diodo a la entrada para evitar polarizaciones equivocadas.

### ***4.2.3 Puertos serie.***

El microcontrolador se alimenta a una tensión de 3,3 V y todas sus patillas de entrada y salida coherentemente presentan como máximo una tensión de este valor o algo inferior. Esto no es excepcional para aquellos patillas que componen los puertos serie.

El puerto serie estándar RS-232C, su versión más actualizada, presenta niveles de tensión normalizada que oscila entre los -12 para el símbolo '1' y los +12 para el símbolo '0'. Con estos datos a la vista es evidente que tiene que haber algún tipo de conversión de niveles entre ambas interfaces, la del microcontrolador y la del puerto serie.

---

<sup>11</sup> SMD: Surface Mount Device. Dispositivo de montaje superficial. Permite la colocación del dispositivo sin necesidad de practicar taladros en la placa y ahorrando por consiguiente más espacio.

## Capítulo 4: Solución/Implementación hardware.

Esta conversión es muy común y para resolverla se utilizan dispositivos llamados transceptores. El que se ha elegido ha sido el ICL3222 de Intersil. Este circuito integrado presenta dos entradas y dos salidas, suficiente para manejar las líneas básicas de dos puertos serie. Además el dispositivo elegido incluye dos patillas, una de habilitación y otra de control de apagado con lo que es posible reducir su consumo habilitando o inhabilitando por separado las líneas de transmisión y de recepción..

Estos circuitos integrados se sirven de algunos componentes como unos condensadores para llevar a cabo su función.

### **4.2.4 Teclado.**

El teclado es una matriz de 4 filas por 4 columnas lo que resulta en un total de 16 teclas. Cuando una tecla se pulsa, se cierra un circuito de forma que aplicando tensión en la fila y observando la tensión en la columna se puede determinar cual es la tecla que ha sido pulsada.

Si se quisiera realizar esta función con el microcontrolador serían necesarias 8 patillas y una rutina que se encargara del rastreo de la tecla. En lugar de esto se ha optado por un circuito integrado que realizará esta función de forma auxiliar. En realidad es un codificador que transforma la lectura del teclado a tan sólo cuatro líneas codificadas en binario con un total de 16 posibles combinaciones, es decir, el número total de teclas disponibles.

El codificador de teclas es el MM54C922 de National Semiconductor que además de realizar la función ya descrita posee un mecanismo para evitar los llamados rebotes o notificación de pulsaciones falsas de teclas.

### **4.2.5 Adaptadores de nivel.**

En cuestión de niveles de tensión como ya se ha visto ha sido necesario realizar la conversión de los niveles del microcontrolador a los niveles particulares del puerto serie. Esta operación es necesario que se repita en casos en los que los niveles de tensión de funcionamiento de un dispositivo no coincida con los suministrados por el microcontrolador.

Tal es el caso del visualizador LCD y del puerto paralelo que servirá para programar el microcontrolador. Estos elementos utilizan niveles TTL, es decir, valores de 0 y 5V, por lo tanto es necesario buscar una solución para la adaptación de los niveles.

Afortunadamente existen circuitos comerciales capaces de realizar esta función, como por ejemplo el 74LVC4245 de Philips Semiconductors. Estos circuitos integrados presentan dos patillas de alimentación que pueden ser de distinto valor además de dos conjuntos de patillas que se corresponden. Si alimentamos a 3,3V una de las patillas de alimentación y la otra 5V, tenemos que las patillas de cada grupo del integrado funcionan en estos niveles, haciendo corresponder el valor lógico del grupo de 3,3V con el valor lógico en la correspondiente del grupo de 5V y viceversa.

El 74LVC4245 dispone además de una patilla de habilitación para poner al circuito en modo de bajo consumo y otra de dirección que establece el sentido en el que se deben propagar las señales. Bien desde el grupo de 3,3V al de 5V o al revés.

### **4.2.6 Consideraciones de diseño para el microcontrolador.**

El microcontrolador es la pieza fundamental de la plataforma, por tanto hay que tener cuidado en su diseño. El fabricante proporciona un diseño de referencia y unas directrices en su kit de evaluación. Estas consideraciones sirven como guía para realizar luego ajustes y modificaciones sobre ellas.

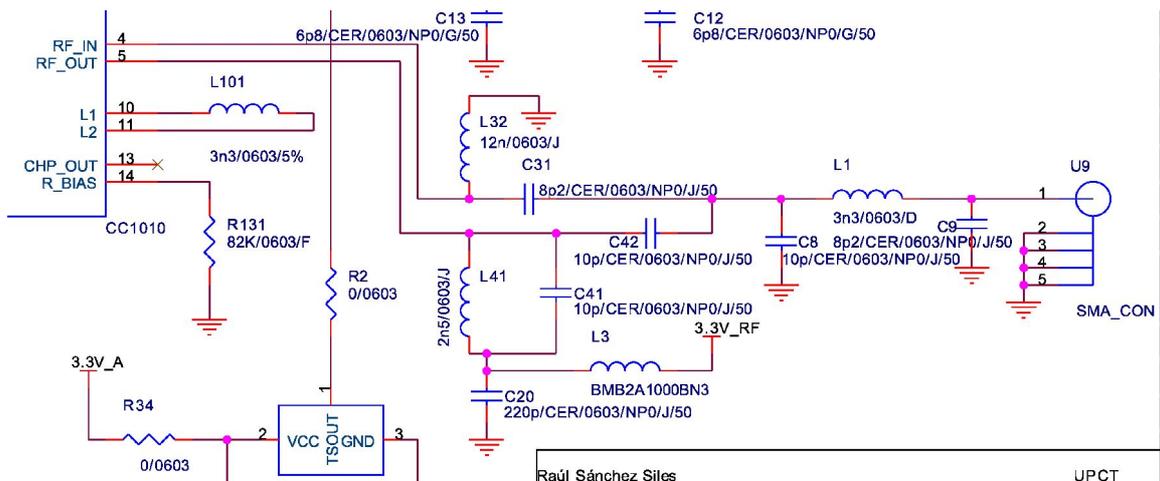
En primer lugar consideraremos la alimentación. Este componente tiene numerosas patillas de alimentación. Cada una de ellas debe incluir un condensador de desacoplo, normalmente cerámico de baja capacidad. Su misión es filtrar el ruido de alta frecuencia.. Todos estos condensadores al final se unen en otro condensador algo más grande, de tántalo que eliminará el ruido de más baja frecuencia.

En los diseños en los que están involucrados señales de alta frecuencia, para evitar precisamente el ruido de alta frecuencia se tiene a incluir ferritas de protección, también conocidas como *chokes*.

En cuanto a la entrada y salida, se ha dotado a cada patilla destinada a este fin de una resistencia a alimentación ó *pull-up* con la que la tensión en cada una de ellas esté siempre en valores conocidos. También se han incluido en dos de las patillas de propósito general, transistores para hacer uso de unos LEDs también de propósito general.

Los cristales por su parte, se han acoplado convenientemente con sus condensadores, imprescindibles para hacerlos oscilar. Este microcontrolador dispone de dos entradas de oscilador, una de alta frecuencia que el la que genera las señales para el funcionamiento de la etapa de radiofrecuencia y otro de baja frecuencia para el funcionamiento del microcontrolador en modo de bajo consumo.

Especial importancia tiene la etapa de radiofrecuencia. Las patillas de salida y entrada de radiofrecuencia se unen a través de un circuito adaptador de impedancia de forma que van a dar a la antena después de atravesar un filtro paso banda (Figura 17).



Raúl Sánchez Sites  
Figura 17: Detalle etapa RF.

En la figura anterior también se aprecia la bobina L101 utilizada para el funcionamiento adecuado del PLL interno y la resistencia R131 para la polarización del mismo.

Existen dos patillas interesantes a la hora de su acceso de forma cómoda. Una es la de reset y otra es la de programación. Mientras que para la primera se ha incluido un pulsador para poder efectuar un reset, en el segundo caso se incluye un jumper para habilitar o inhabilitar la programación del dispositivo.

### 4.3 Estudio de consumo.

El dimensionamiento de ciertos componentes se hace importante a la hora del diseño. En éste uno de los más importantes ocurre con la etapa de alimentación. Es necesario conocer el consumo que tendrá la PCB para dimensionar los dispositivos a utilizar de forma que sean los más adecuados para la aplicación que se les va a dar.

En la Tabla 3 podemos ver una estimación de los consumos de cada dispositivo. Siempre se tiene en cuenta el peor caso.

Tabla 3: Estimación del consumo de los componentes.

<b>Dispositivo (unidades)</b>	<b>Tensión de funcionamiento (V).</b>	<b>Corriente máxima (mA) por unidad.</b>
LM61 (1)	3,3	1
CC1010 (1)	3,3	41
MM74C922 (1)	5	1
ICL3222 (1)	5	1
74LVC4245 (3)	5 / 3,3	4 / 4
LCD (1)	5	2,5
LED5 (1)	5	13

<i>Dispositivo (unidades)</i>	<i>Tensión de funcionamiento (V).</i>	<i>Corriente máxima (mA) por unidad.</i>
LED3,3 (4)	3,3	6

Con esto obtenemos una corriente total de 29,5mA a 5V y 78 mA a 3,3V que se reparten entre el regulador para alimentación analógica y el de alimentación digital. La corriente que se extrae de los reguladores de 3,3V se extrae a su vez del de 5V con lo que a la corriente requerida al regulador de 5V es la de la circuitería que funciona a 3,3V más la de los propios circuitos que funcionan a 5V.

Este funcionamiento es el habitual para reguladores lineales, por lo que la corriente requerida por la PCB será la suma de todas las corrientes a todas las tensiones que los componentes requieren. Esto puede ser delicado en el primer regulador que es el que tiene que disipar la energía sobrante en forma de calor. Esta es una de las limitaciones que tienen este tipo de reguladores.

#### **4.4 Diseño electrónico. Herramientas.**

El diseño de una PCB consta de dos fases principales. La primera es la creación del esquemático, la segunda la delineación del PCB.

En la primera fase, el esquemático indica los componentes utilizados y la conexión que hay entre ellos. En esta fase hay que crear las librerías de símbolos que representan cada componente electrónico. Mientras que los componentes más comunes suelen estar creados, los componentes más particulares como el microcontrolador no suelen estar incluidos. Pero esto depende mucho del software que se utilice para crear el esquemático.

En la segunda fase, la delineación del PCB consiste en hacer corresponder a cada componente lógico su componente físico, la huella que define la forma del componente real. Después se colocan todos los componentes convenientemente en las caras del PCB y por último se trazan las pistas que unan los componentes según lo establecido previamente en el esquemático.

##### **4.4.1 Herramientas.**

Para la creación del esquemático existen varias alternativas en cuanto a software. Las más significativas por su reconocimiento son las siguientes:

- ◆ PADS: Software propietario de la empresa Mentor Graphics que incluye editor de esquemáticos y herramientas de PCB y simulación. Sólo disponible para S.O. Windows.
- ◆ CadStar: Otra solución propietaria de la empresa Zucken. También incluye editor de esquemáticos, herramientas para el diseño de PCB y análisis térmicos y de compatibilidad electromagnética. Sólo disponible para S.O. Windows.
- ◆ Orcad: familia de herramientas para desarrollo electrónico. Dispone de herramientas de simulación, editor de esquemáticos, editor de PCB y herramientas de autorroteo. Sólo disponible para S.O. Windows.

- ◆ Eagle: Herramienta de desarrollo electrónico disponible para varias plataformas (Linux, Windows ó Mac). Dispone de un editor de esquemáticos y una herramienta de delineación de PCB. Existe una versión para uso no comercial gratuito con algunas restricciones. Dispone de multitud de recursos en su página web.
- ◆ gEDA: conjunto de utilidades GNU para el desarrollo electrónico. Incluye librerías, editor de esquemáticos, simulador de circuitos, delineación de PCB. Aún se encuentra en estado alfa de desarrollo, pero presenta varias utilidades y sus formatos son abiertos. También se puede ejecutar sobre varios S.O. incluyendo Linux y Windows.

De entre estas herramientas se ha seleccionado la familia Orcad al ser con la que se tenía experiencia anterior. De este modo se evitaba la curva de aprendizaje necesaria para utilizar el resto. Otro motivo para esta elección ha sido que ya se disponía de algo de trabajo realizado en el formato de Orcad, los cambios de formato en este tipo de herramientas no son nada sencillos, casi imposibles, esta era otra de las razones para mantener la misma dinámica.

Sin embargo, para nuevos diseños, merecería la pena utilizar bien “gEDA” ó bien “Eagle”. Estos programas son útiles para hacer diseños que no sean muy extensos, el desarrollo avanza rápidamente y se evita el pago de licencias.

#### **4.4.2 Esquemático.**

Como ya se ha mencionado, Orcad es un conjunto de herramientas para realizar el diseño de placas de circuito impreso. Consta de varias herramientas. La primera herramienta a considerar es el editor de esquemáticos, llamado Orcad capture.

Con él crearemos los símbolos de los componentes utilizados además de las conexiones entre ellos. Posteriormente, se realiza una conversión con la que se consigue un archivo en el que figuran estas conexiones. Este archivo se denomina *netlist*, y no es más que una enumeración de cada una de las líneas por las que van a circular las distintas señales y las líneas de alimentación y masa. Cada línea tiene uno o varios orígenes y uno o varios destinos. En este archivo se especifican los mismos para cada una de las señales y para cada uno de los componentes utilizados.

En el archivo *netlist* es imprescindible que aparezcan también la referencia de los componentes utilizados, con las que después se hará la correspondencia con la representación del componente físico.

Esto último aspecto es muy importante para la obtención de la llamada *lista de materiales*, una relación de todos los componentes utilizados en la que se detallan referencia, cantidad y valores.

#### **4.4.3 PCB.**

La herramienta con la que diseñará el PCB es el Orcad Layout. Esta herramienta extrae la información del archivo *netlist* generado por la herramienta anterior con lo que crea un boceto de la PCB, de forma que queda constancia de las conexiones entre elementos.

### ***Posicionamiento.***

Posteriormente se hace corresponder cada componente lógico con sus dimensiones físicas reales. Si éstas no se encontraran en las librerías de la herramientas, tendríamos que crear la *huella*, es decir, la forma y tamaño que tendría el componente real.

Finalizado esto, obtenemos un boceto de la placa que nos sirve para hacernos una primera aproximación de la placa. Sin embargo, los componentes no están posicionados en su lugar definitivo, ni siquiera se han definido las dimensiones para la PCB.

Creamos un borde para la placa y colocamos dentro de este área todos los componentes. Observamos una gran cantidad de líneas amarillas que indican las uniones lógicas en este momento y que luego se harán físicas entre componentes. Estas líneas se convertirán al final del proceso en pistas.

Ahora que tenemos los componentes colocados dentro de los límites de la PCB, intentamos acomodarlos de forma que los procesos mecánicos se puedan llevar a cabo satisfactoriamente. Por ejemplo, los conectores deben estar al borde de la PCB, y los LED deben ser visibles. Otras consideraciones afectan a la colocación del conector de la antena: debe estar próximo al borde de la placa; el visualizador y el teclado deben posicionarse de forma que el usuario pueda acceder a ellos cómodamente.

Otra observación importante deducida del estudio del esquemático, es que para que las pistas sean lo más cortas y homogéneas posibles, el microcontrolador debe colocarse de modo que esté centrado respecto a sus periféricos. Para ello y para aprovechar al máximo el espacio disponible colocaremos componentes por las dos caras, siempre que sea conveniente.

Atendiendo a condicionamientos eléctricos, posicionaremos los llamados condensadores de desacoplo lo más cerca que se pueda de las patillas de alimentación de los componentes a los que prestan servicio. Como norma general, los componentes asociados como los que se acaban de mencionar, deben permanecer cercanos, evitando así longitudes de línea innecesarias beneficiando a la integridad de las señales que se transportan. Con esta misma finalidad intentamos alejar dentro de las posibilidades grupos potencialmente ruidosos como las fuentes de alimentación de grupos más sensibles a ruido como la etapa de radiofrecuencia.

### ***Delineación.***

Con estas observaciones ya tenemos colocados todos los componentes y procederemos a su interconexión mediante pistas. Para empezar el proceso es necesario decidir la anchura de las pistas y el diámetro de *vía*<sup>12</sup> que se puede utilizar. Estos dos parámetros son muy dependientes de la tecnología que se utilizará para la fabricación. En este caso, los valores en los que oscila la anchura de pista serán, en última instancia, mayores que 0,5mm siendo conveniente que sean mayores que 0,7mm. En cuanto al diámetro de vía utilizaremos dos medidas, un diámetro de 1,3mm para señales de alta corriente como alimentaciones y masa; y otra de 1mm para señales con baja corriente.

La anchura de las pistas también dependerá de el uso de cada una de ellas, así las pistas de alimentación normalmente trasladan corrientes más altas por lo que interesa que sean cortas y anchas, minimizando así la inductancia y la resistencia. Por el contrario, las pistas que transporten señales digitales puede tener anchuras menores al no transportar grandes corrientes, intentando acortarlas todo lo que se pueda y evitando caminos revirados que pueden padecer interferencias ó diafonía de otras señales. En todo caso se evitarán líneas de distintos dispositivos que discurran de forma paralela.

También es la tecnología de fabricación la que establece el número de caras y capas utilizables para el uso en una PCB. En este caso utilizamos dos capas tanto para la colocación de componentes como para el trazado de pistas, además como se ha mencionado se permite el uso de vías conductoras entre capas.

Para finalizar con las consideraciones para el trazado de pistas, se observa la señal fundamental en todo circuito electrónico: la masa o referencia. En este caso, aún existiendo diversos valores para las tensiones de alimentación sólo existirá un único plano de masa para evitar bucles de masa, contrarios al buen funcionamiento de la circuitería. Por lo tanto la masa se dispersará en forma de plano ó planos (1 por cada cara).

### **Autorouting.**

En este momento el objetivo es lograr trazar todas las pistas que transportan las señales de la forma más óptima sin que haya cruces entre ellas. Un cruce supondría la unión indeseada de dos señales con consecuencias impredecibles, comúnmente el fallo total del sistema diseñado. Otra incorrección común ocurre cuando una vía intercepta a una pista.

El trazado de pistas no es tarea sencilla, más aún cuando el número de señales de la PCB se incrementa. Realizar esta tarea de forma manual es muy tedioso y propensa a numerosos errores con sus correspondientes correcciones. Por ello se hace necesario la utilización de herramientas que automaticen el proceso.

---

12 via: taladro conductor que se practica en la PCB para conducir una señal de una cara a la opuesta.

## Capítulo 4: Solución/ implementación hardware.

Lo más común es trazar en primer lugar las pistas más críticas, de forma que se encuentre un camino óptimo para ellas en un inicio, cuando no hay obstáculos. Este podría ser el caso de las pistas de alimentación, un bus de datos o las pistas que contengan señales de radiofrecuencia. En este caso es viable una solución manual.

Para el resto de señales se puede recurrir a herramientas como el Orcad Spectra, utilizado en este Proyecto. Con ella se definen los parámetros más importantes de la PCB como las anchuras de pista y vía para cada caso y otros como cuáles son las señales de alimentación y masa.

La herramienta realizará sus cálculos, intentando llegar a una solución óptima, la cual no es siempre posible. De hecho, es posible que la herramienta encuentre imposible completar el proceso. Por este motivo y porque los trazados resultantes no siempre son aconsejables, se hace necesario repetir el proceso con otros valores, igualmente compatibles con las especificaciones de fabricación esperando que en sucesivas ocasiones haya más “suerte”.

Experimentadas varias realizaciones del proceso, seleccionamos la que obtenga mejor resultado para retocarla manualmente. Se trata de mejorar el trabajo hecho por la herramienta. Es fácil observar pistas con trazados sinuosos o excesivamente largos. Con esto se intentará retrazar las pistas lo que puede llevar a consecuencias ligeramente más graves como por ejemplo, la necesidad de recolocar un componente o retrazar otras pistas de forma que el resultado global mejore.

Después de varios ensayos se obtiene un resultado aceptable.

### **4.5 Programación sistema.**

Según lo establecido, el microcontrolador dispone de 32KB de memoria flash que se deben programar. El mecanismo que proporciona el microcontrolador seleccionado para realizar tal tarea es a través de un puerto SPI.

El bus SPI es un bus estandarizado por Motorola que consiste en una línea serie de entrada (SI) una línea serie de salida (SO) y una de reloj (SCK). Adicionalmente, para poner el microcontrolador en modo programación es necesario controlar las líneas PROG y RESET. Todo esto es posible implementarlo y conducirlo a través del puerto paralelo de un PC.

Por este motivo se ha dispuesto de un conector DB25 para la conexión directa de un PC y así realizar la programación. En definitiva el PC crea las señales con la temporización adecuada para enviar los datos a la memoria flash del microcontrolador.

Un software propietario y gratuito suministrado por el fabricante (Chipcon) es el que realiza esta función. Este software se ejecuta bajo Windows (Figura 18), si bien ha sido posible obtener el código fuente por parte del fabricante, no pudiendo reproducirse aquí por restricciones de licencia.



Figura 18: Software de programación del CC1010.

#### 4.6 Construcción.

En esta sección se explica someramente los pasos a seguir para fabricar la PCB.

Como resultado de la delineación, se obtienen los planos de las distintas capas que componen la PCB. Para este caso, una PCB con componentes y pistas por ambas caras, se tienen las siguientes capas:

- ♦ Cara de componentes: define las pistas en la cara de componentes ó superior. Los componentes de la cara de componentes se sueldan en esta capa.

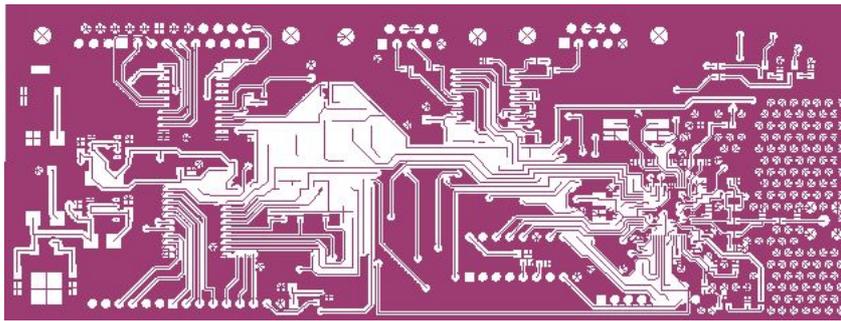


Figura 19: Capa PCB componentes.

- ◆ Cara de soldadura: define las pistas en la cara de componentes ó inferior. Los componentes de la cara de soldadura se sueldan en esta capa.

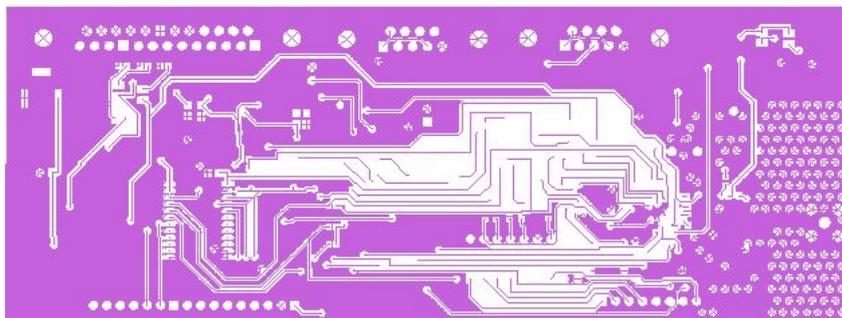


Figura 20: Capa PCB soldadura.

- ◆ Serigrafía componentes: dibuja la posición de los componentes en la cara de componentes. Se usa para localizar el componente en esta capa.

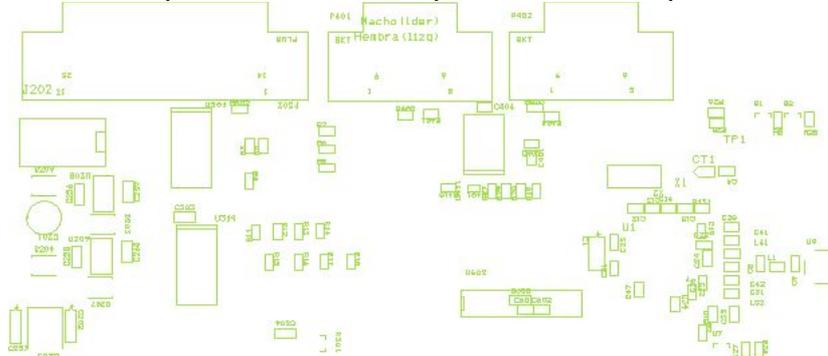


Figura 21: Capa PCB serigrafía componentes.

- ◆ Serigrafía soldadura: dibuja la posición de los componentes en la cara de soldadura. Se usa para localizar el componente en esta capa.

## Capítulo 4: Solución/Implementación hardware.

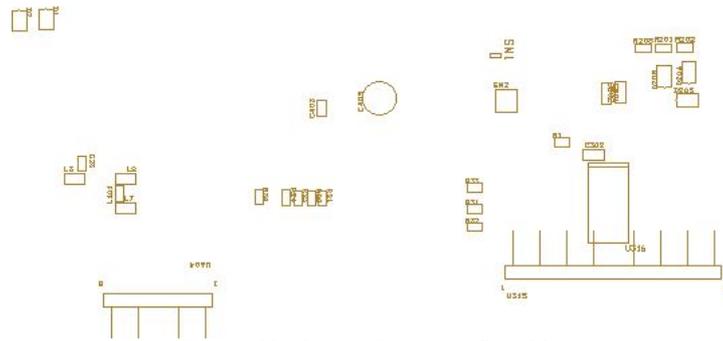


Figura 22: Capa PCB serigrafía soldura.

- ◆ Taladros: define dónde se deben practicar los taladros y las vías.

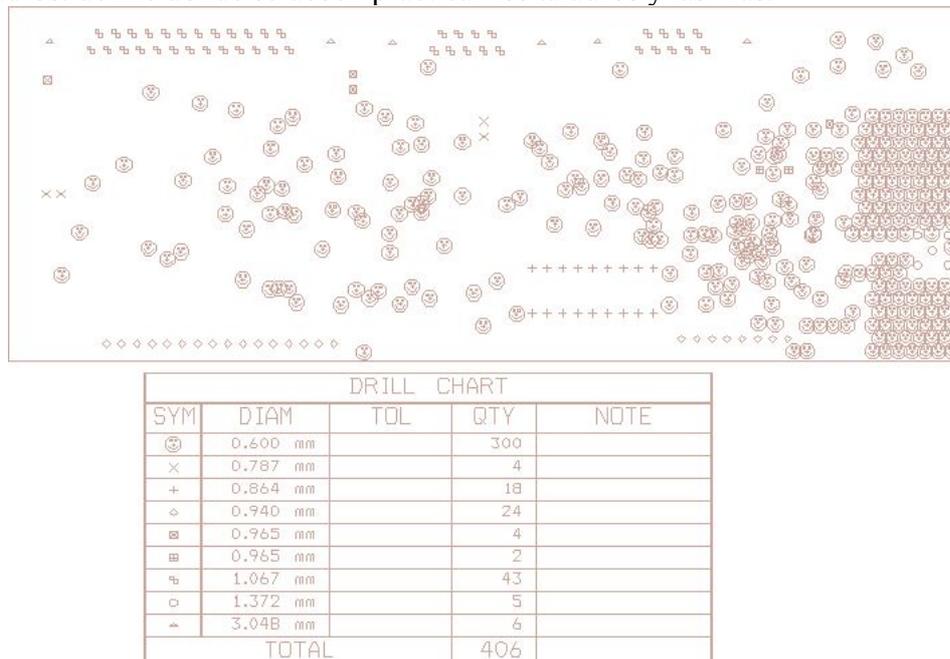


Figura 23: Capa PCB taladros.

Con estos planos se puede comenzar el mecanizado de la PCB, se metalizan las vías de tal forma que haya continuidad entre capas y por último se termina el mecanizado.

En este momento se tiene una placa en la que quedan definidas las vías, las pistas y las huellas de cada componente. Con lo que se puede proceder al montaje de los componentes.

La fase de montaje consiste en posicionar y soldar los componentes a la PCB. Para ello se suele utilizar un soldador si el montaje se hace por métodos manuales, pero existen otras técnicas más avanzadas para realizar este proceso por medios mecánicos.

Esta placa ha sido montada manualmente, soldado cada componente uno a uno. La única excepción en este caso ha sido el montaje del microcontrolador. Este componente suponía una dificultad y riesgo a la hora de soldar, por lo que su montaje se ha efectuado por medios mecánicos, mediante soldadura por infrarrojos.

## 5 Solución/Implementación software.

Para aprovechar al máximo las funcionalidades de la plataforma es necesario diseñar un firmware que se adapte a la aplicación y que cuente con las funcionalidades de la plataforma.

El software asociado a las aplicaciones que se le pueden dar a la plataforma también es necesario diseñarlo cuidadosamente. Todo dependerá de la aplicación, pero el servicio que un usuario final vea en la plataforma está condicionado a la creación de estas dos piezas fundamentales. Sin embargo, tanto el desarrollo del firmware como el del software quedan fuera del alcance de este Proyecto.

En este caso, nos limitamos a indicar el firmware y el software necesario para hacer algunas pruebas a la plataforma y así poder comprobar su correcto funcionamiento. Para esto se realizará la descarga de un firmware adecuado en la plataforma, utilizando un software apropiado en un ordenador externo y realizando algunas comprobaciones en el hardware.

### 5.1 Herramientas.

Para el desarrollo de firmware para la plataforma se sigue el proceso habitual de desarrollo de cualquier firmware, sólo hay que tener en cuenta los periféricos con los que se cuenta y que los recursos son muy limitados, por lo que la programación debe ser muy eficiente.

Las fases para el desarrollo son:

- ◆ Creación y edición de código. Para esto es necesario un editor de texto. Cualquiera puede valer, pero es aconsejable y más cómodo que es editor disponga resaltado de sintaxis y que sea capaz de trabajar con varios archivos.
- ◆ Ensamblador/Compilador. El código se escribirá en ensamblador ó en lenguaje de más alto nivel como el C. El ensamblador ó el compilador se encargará de traducir el código a lenguaje máquina y crear un archivo objeto con ese código.
- ◆ Enlazado. Aunque los archivos objeto contienen código comprensible para el microcontrolador, no son suficientes por sí mismo de ejecutarse. Para esto necesitan enlazarse unos con otros, pues pueden existir funciones que se utilicen en un archivo objeto y estén localizadas en otro. En enlazador suele incluir también un código de inicio que ajusta los registros internos del microcontrolador de forma que el programa se pueda ejecutar. El resultado del enlazador es un archivo que se puede descargar directamente en la memoria de programa del microcontrolador.
- ◆ Herramienta “*GNU make*”. Si bien es posible realizar el ensamblado/compilado y enlazado de forma manual, ejecutando los comandos sucesivamente, es más cómodo utilizar una herramienta que automatiza el proceso llamada “*make*”. Esta herramienta comprueba si los archivos han cambiado desde la última ejecución y sólo procesa los archivos imprescindibles.

Como editor se recomienda “*kate*” un editor sencillo de utilizar con resaltado de sintaxis, gestión de múltiples archivos y más funcionalidades. Si bien como ya se ha dicho es posible utilizar cualquier otro.

Como ensamblador/compilador se recomienda el uso de “*sdcc*” que también es un enlazador. “*sdcc*” es un compilador para “dispositivos pequeños”, que compila código C, ensamblador y está diseñada para microcontroladores como el 8051 ó el Z80. software libre por lo que se puede participar en su desarrollo. Es estable, está bien documentada y es totalmente compatible con el estándar ANSI C.

El paquete “*sdcc*” también incluye un simulador que se puede utilizar con front-ends compatibles con gdb, como por ejemplo “*DDD*”.

La salida que se obtiene de “*sdcc*” tiene formato *intel hex* un formato que define el contenido de la memoria bastante utilizado.

## **5.2 Programa de ejemplo.**

Se incluye un programa de ejemplo en el apéndice B, extraído de los ejemplos suministrados por Chipcon. Para compilarlo es necesario las librerías y los archivos de cabecera también suministrados por Chipcon.

El listado en el apéndice B es un ejemplo puede utilizar a su vez para comprobar el correcto funcionamiento de la placa. Su utilidad es transmitir y recibir paquetes por vía inalámbrica con otro dispositivo de tal forma que todo la información se refleja en el puerto serie. En pocas palabras, es un módem nulo de radiofrecuencia.

## 6 Pruebas.

La realización de pruebas es imprescindible para comprobar el correcto funcionamiento de la Plataforma. Estas pruebas deben incluir comprobaciones eléctricas y comprobaciones funcionales.

Las primeras verifican la integridad de los componentes, de la fabricación de la PCB y del posterior montaje. Mientras que las segundas demuestran que el funcionamiento es el esperado.

### 6.1 Pruebas eléctricas.

Las pruebas eléctricas son el primer paso para comprobar que la fabricación y el montaje se ha llevado a cabo correctamente. En primer lugar se describen las inmediatamente posteriores al proceso de fabricación:

- ◆ Comprobación visual del proceso de fabricación. Observar que no hay pistas cortadas ni taladradas. Las vías se han practicado en el lugar oportuno.
- ◆ Comprobación eléctrica del proceso de fabricación. Hay continuidad en las pistas y las vías conducen corriente. No hay cortocircuitos.
- ◆ Comprobación de las huellas. Aunque esta comprobación ya se ha realizado durante la fase de diseño y de prefabricación no viene demás volver a realizarla sobre la PCB física. Consiste en presentar los componentes sobre su huella. Comprobar que encajan y que las pistas salen desde el lugar correspondiente.

Una vez comprobado esto se procede con el montaje de los componentes. A continuación, con los componentes ya montados se hacen las siguientes pruebas:

- ◆ Los componentes están en la posición correcta y con la orientación correcta.
- ◆ Comprobación visual de la corrección de la soldadura. No debe haber soldaduras irregular, frías. El estaño se ha aportado en la cantidad justa y no hay rebosamientos que provoquen cortocircuitos entre patillas.
- ◆ Comprobación eléctrica de la soldadura. Las patillas están eléctricamente conectadas a las pistas y a las de los componentes cuya señal se desea transmitir.

### 6.2 Pruebas funcionales.

Cuando se tiene la seguridad de que lo realizado hasta ahora es correcto, se procede con las pruebas funcionales que tratan de averiguar que cada componente cumple con su función.

Sin darle alimentación al circuito se comprueba la etapa de alimentación, sobre todo lo que concierne a los diodos de protección. Se comprueba con un polímetros que están funcionan correctamente. A continuación se comprueban los condensadores de desacoplo asegurándose de que dan en circuito abierto. Si alguno de ellos estuviera estropeado es posible que creara un cortocircuito entre las pistas de alimentación, pudiendo estropearla o incluso estropeando la propia etapa de alimentación.

Se alimenta la PCB y se comprueba que los diodos lucen correctamente, síntoma de que hay tensión y de que están correctamente polarizados. A continuación se comprueba que las patillas que suministran la alimentación a la PCB dan la tensión esperada.

Por último, se comprueba que la tensión en las patillas del resto de componentes estén en los rangos adecuados. Esto incluye comprobar con un osciloscopio que los cristales oscilan de la forma esperada.

Hasta aquí ha sido posible comprobar, puesto que en estas circunstancias el microcontrolador no ha sido programado y no es posible conocer si funciona correctamente.

### **6.3 Programación del microcontrolador. Pruebas del microcontrolador.**

Una vez se tiene el firmware de prueba preparado para su descarga en el microcontrolador, se conecta la interfaz de programación, se carga el software de programación y se efectúa la programación.

Si la descarga es exitosa se demuestra que parte de la funcionalidad del microcontrolador es correcta y por lo tanto es un buen síntoma de que el microcontrolador va a funcionar correctamente.

Según la funcionalidad del microcontrolador que se quiera probar se utilizará un firmware u otro distinto. Se proponen las siguientes pruebas para verificar el completo funcionamiento de la Plataforma:

- ◆ Comprobación de los puertos serie y de los leds. Firmware que recoja los caracteres recibidos por cada puerto serie y los transmita uno a uno por los dos puertos serie simultáneamente. Cada vez que se reciba un carácter se encenderá uno de los leds.
- ◆ Comprobación de la etapa de radiofrecuencia. Firmware que recoja los caracteres recibidos por el puerto serie y los transmita por la etapa de radiofrecuencia. Otra plataforma con la configuración opuesta los debe recibir y transmitirlos posteriormente por uno de los puertos serie.
- ◆ Comprobación del visualizador. Firmware que muestre por el LCD un mensaje de prueba.
- ◆ Comprobación del teclado. Cada pulsación de una tecla se mostrará en el display.
- ◆ Comprobación del sensor. Mostrar por el display el valor leído del sensor de temperatura.

Con esto quedan cubiertos todos los funcionamientos básicos de la plataforma.

### **6.4 Pruebas de prestaciones.**

Una vez que se ha comprobado que los dispositivos funcionan perfectamente, se hace necesario conocer los límites de funcionamiento de la Plataforma. Entre ellos destacamos el consumo o el alcance de la etapa de radiofrecuencia.

- ◆ La comprobación del consumo eléctrico de la placa se efectúa con un polímetro y una fuente de alimentación, colocando el polímetro en serie entre la fuente de alimentación y la PCB.
- ◆ La comprobación del alcance de la etapa de radiofrecuencia se puede efectuar con la ayuda del ejemplo rx\_tx\_switch proporcionado por Chipcon. Una vez ejecutado el firmware, se van alejando las plataformas en pruebas mientras se observa la señal recibida.

## **7 Conclusiones. Líneas futuras.**

### **7.1 Conclusiones.**

Se ha analizado brevemente la complejidad de los procesos logísticos, sus puntos fuertes y sus puntos débiles donde es posible conseguir mejores resultados a través de la investigación y el desarrollo.

Se ha completado el diseño de la Plataforma planteada y se han fabricado unos prototipos para comprobar su funcionamiento. Este Proyecto sirve como punto de partida para el desarrollo de un sistema completo de gestión logística.

Se han propuesto unos métodos de prueba para la plataforma que una vez superados supondrán el inicio de la etapa de desarrollo del firmware y aprovechamiento de las posibilidades de la Plataforma.

Se establecen una líneas futuras por las que debe seguir el desarrollo hasta llegar a un sistema funcional y aplicable habiendo concluido que la aplicación de esta tecnología en la empresa posibilita la innovación de procesos, la reducción de costes, la completa trazabilidad de productos, así como una mejor gestión de la información del producto.

### **7.2 Líneas futuras.**

A continuación se establecen las líneas futuras que se deben seguir para completar el desarrollo de un sistema de control y gestión logístico a partir de la Plataforma desarrollada.

#### **7.2.1 Corto plazo.**

- ◆ Adaptación del software de programación para otros sistemas operativos/ plataformas.
- ◆ Completar las pruebas para demostrar que la plataforma es totalmente funcional.
- ◆ Diseño de un protocolo de comunicaciones óptimo para el tipo de aplicaciones estudiado.
- ◆ Implementación del protocolo de comunicaciones en la plataforma.
- ◆ Optimizar diseño eléctrico. Utilizar herramientas de diseño electrónico basadas en software libre.

#### **7.2.2 Medio plazo.**

- ◆ Incluir más sensores.
- ◆ Utilizar tecnología de localización (GPS, Galileo).
- ◆ Desarrollo de un software de gestión y control logístico a alto nivel.
- ◆ Una vez conseguida la solución estable deseada es necesario optimizar el resultado para su implantación a nivel comercial.
- ◆ Adaptar la plataforma para comercializarlo.

## Referencias

[WPO] : Varios, Wikipedia, 2005, [www.wikipedia.org](http://www.wikipedia.org)

[BAR0] : Adams Communications , 2-Dimensional Bar Code Page, 1995,  
<http://www.adams1.com/pub/russadam/stack.html>

[BAR1] : Vault Information Services LLC, Barcode symbologies, 2004,  
<http://www.barcodeisland.com/symbolgy.phtml>

[BAR2] : Azalea Software, Inc., White Papers, 2005, <http://www.azalea.com/WhitePapers/>

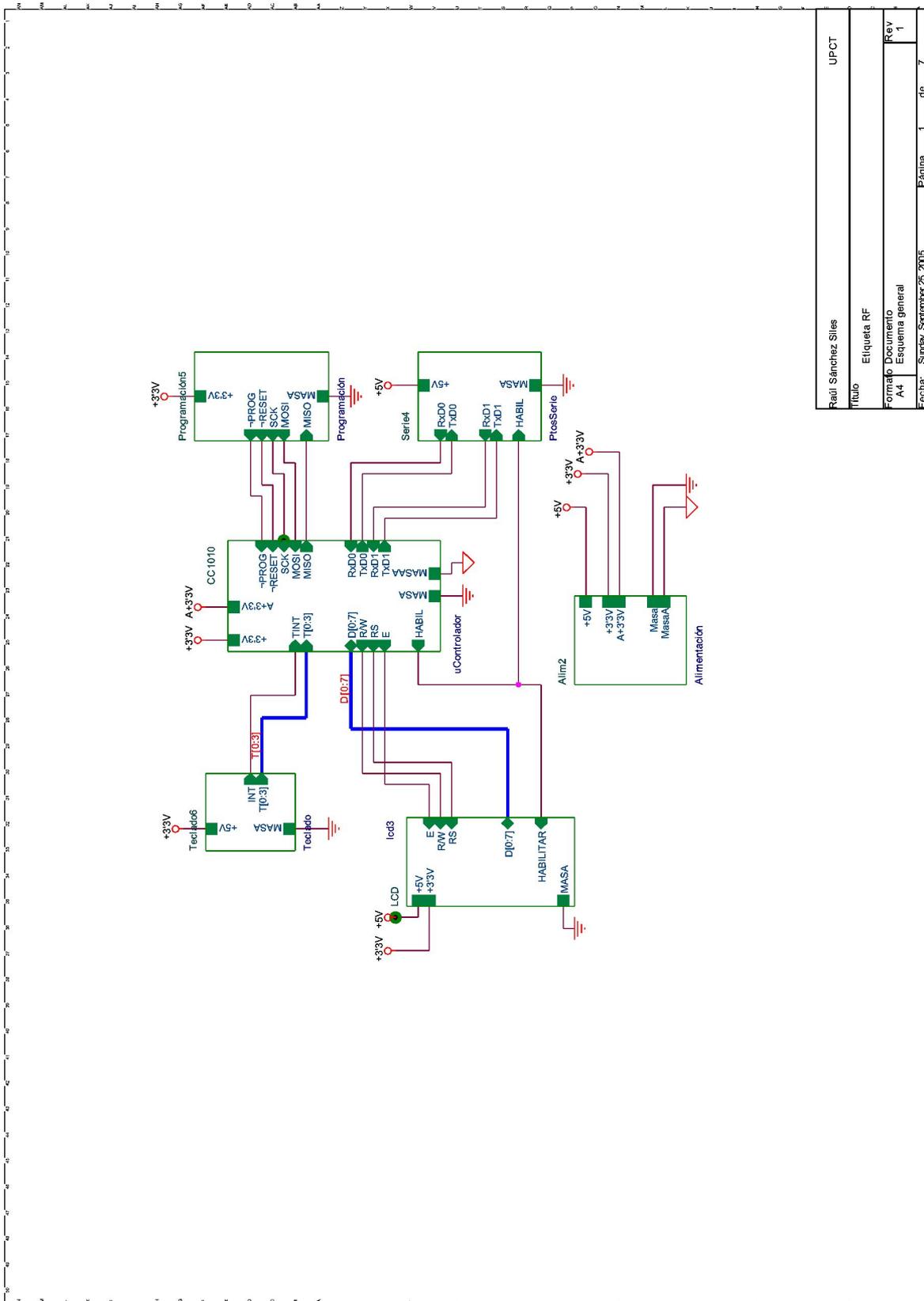
[UOC] : Mark Harrison, EPC Information Service, 2004,  
<http://www.ifm.eng.cam.ac.uk/automation/research/epcis.html>

LANDT01: Jeremy Landt, Shrouds of Time:The history of RFID, 2001

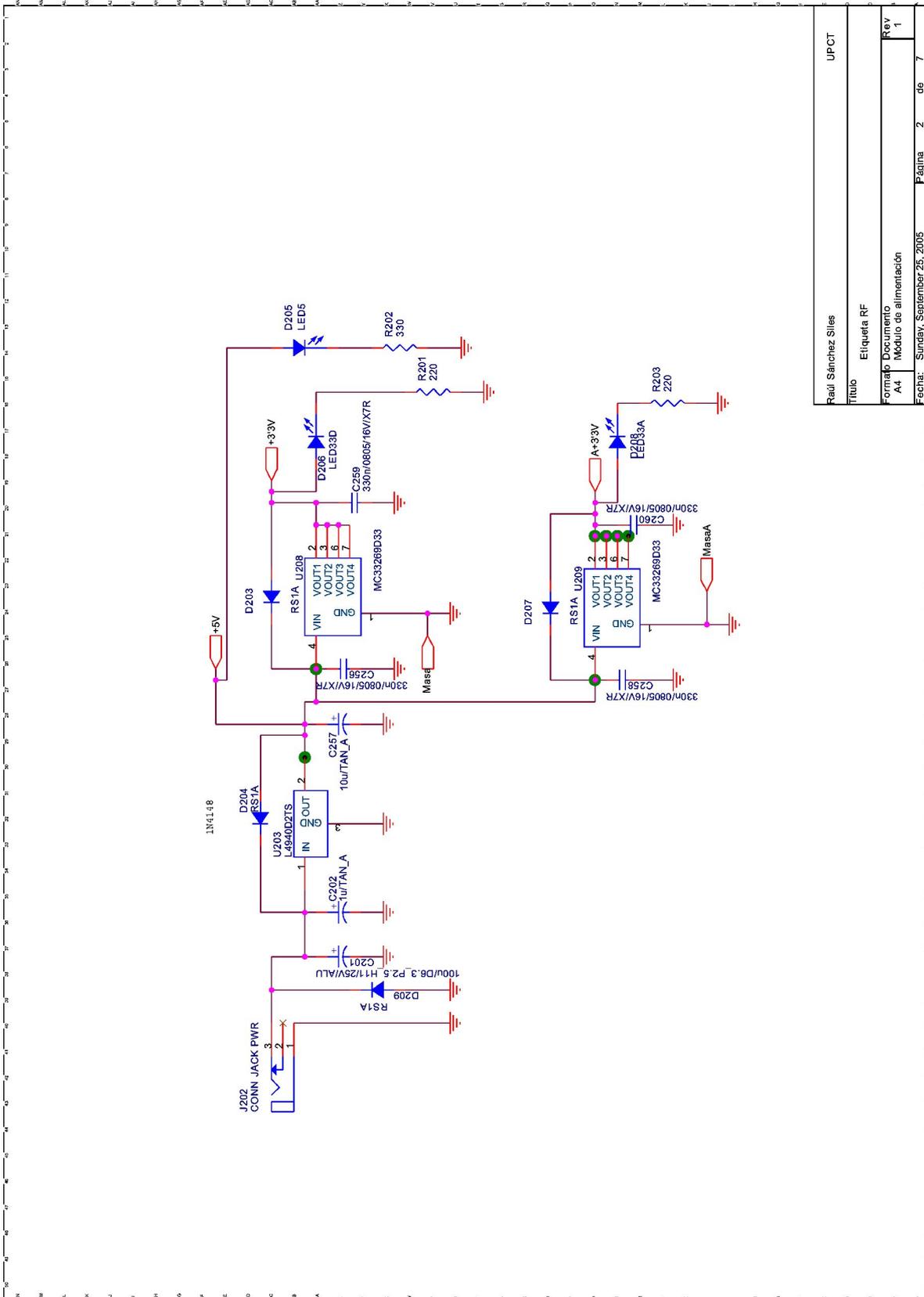
CH05: Chee Hau, The Impact of RFID Technologies on Supply Chain, 2005

[RFJO] : RFID Journal, RFID Journal, 2005,  
<http://www.rfidjournal.com/article/articleview/208>

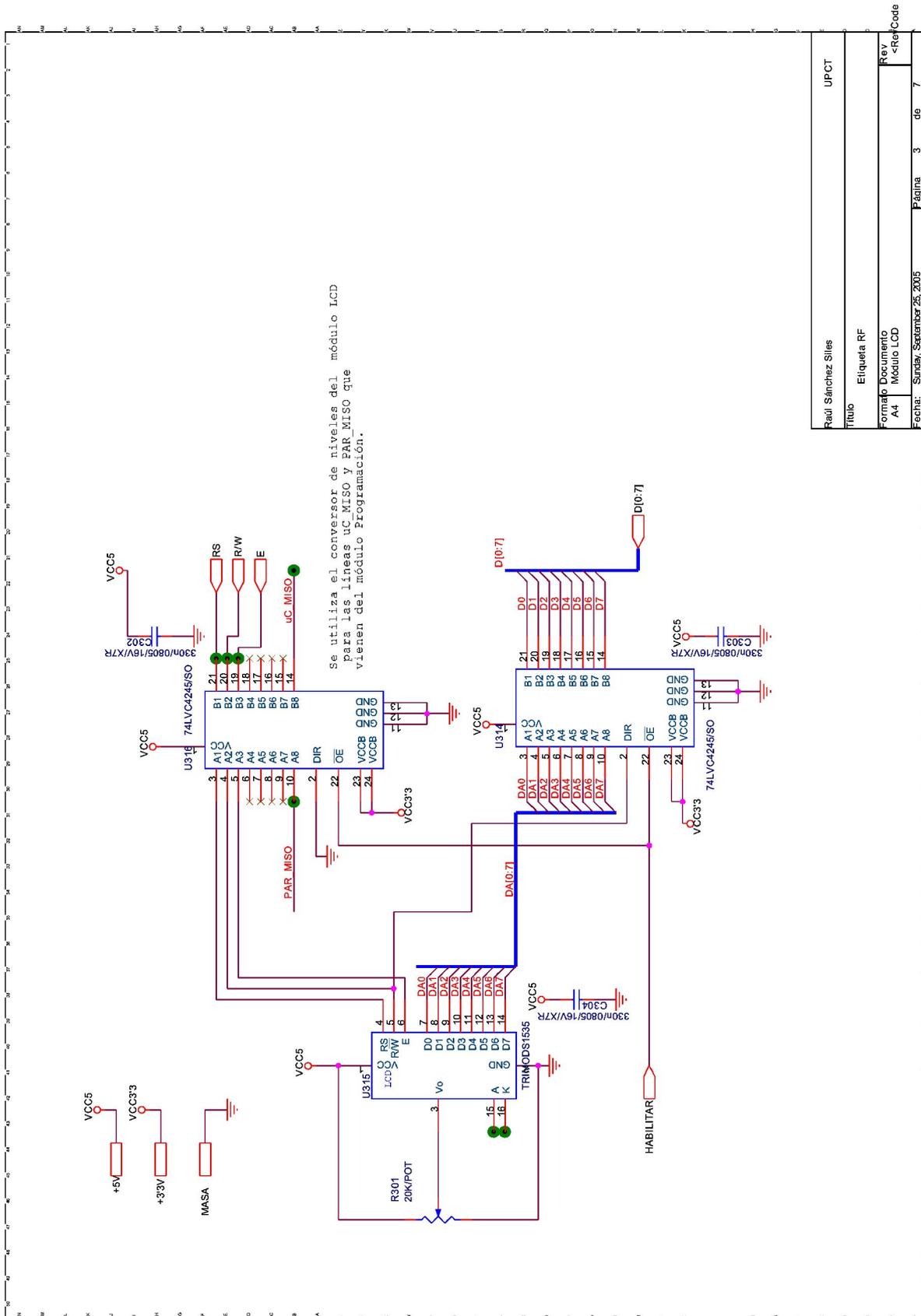
Apéndice A. Diseño electrónico.



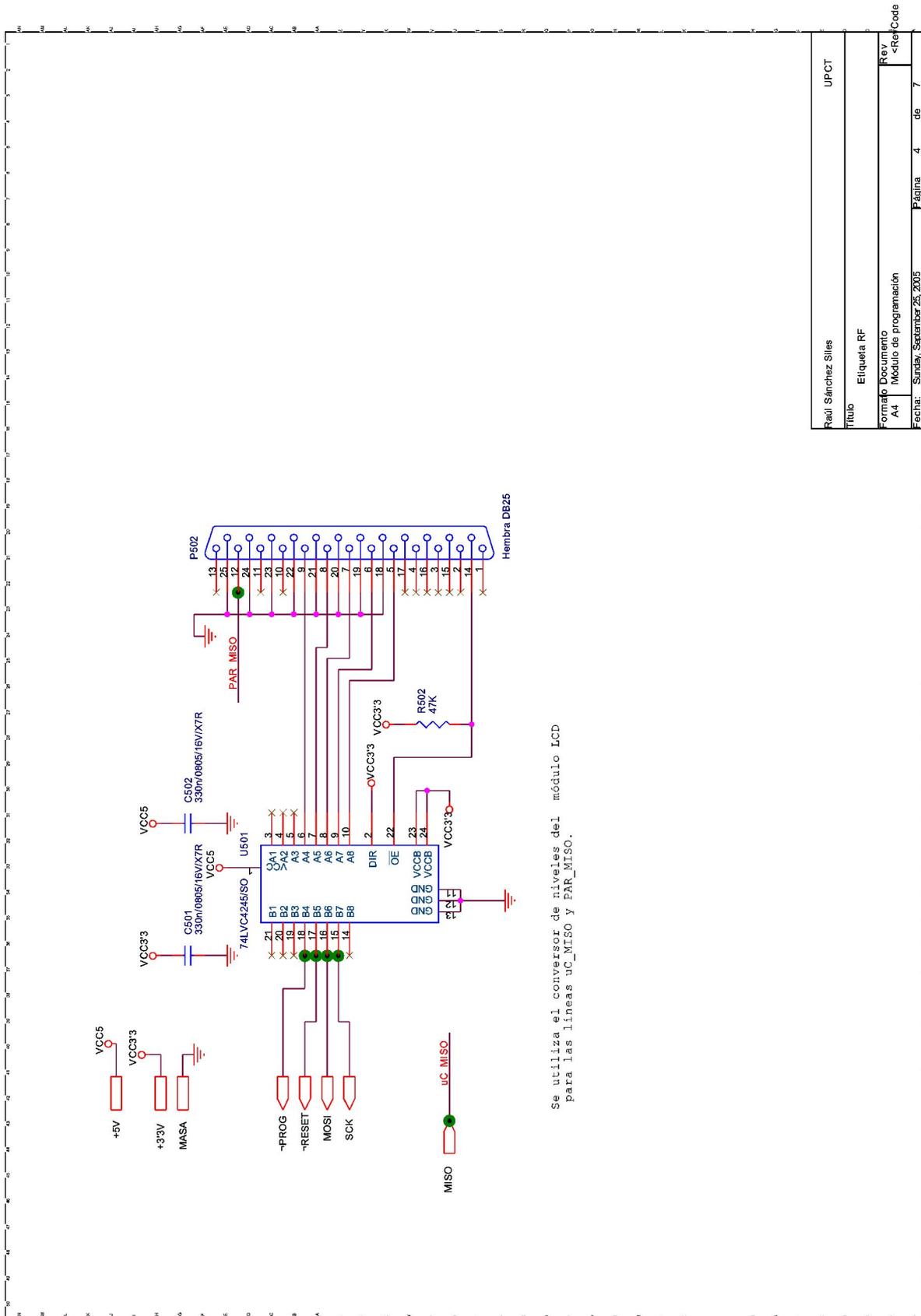
Raul Sánchez Siles		UPCT	
Titulo		Etiqueta RF	
Formato Documento		A4	
Esquema general		Rev 1	
Fecha:	Sunday, September 25, 2005	Página	1 de 7



UPCT	
Raul Sánchez Siles	
Etiqueta RF	
Formato Documento	
A4 Módulo de alimentación	
Fecha:	Sunday, September 25, 2005
Página	2 de 7
Rev	1

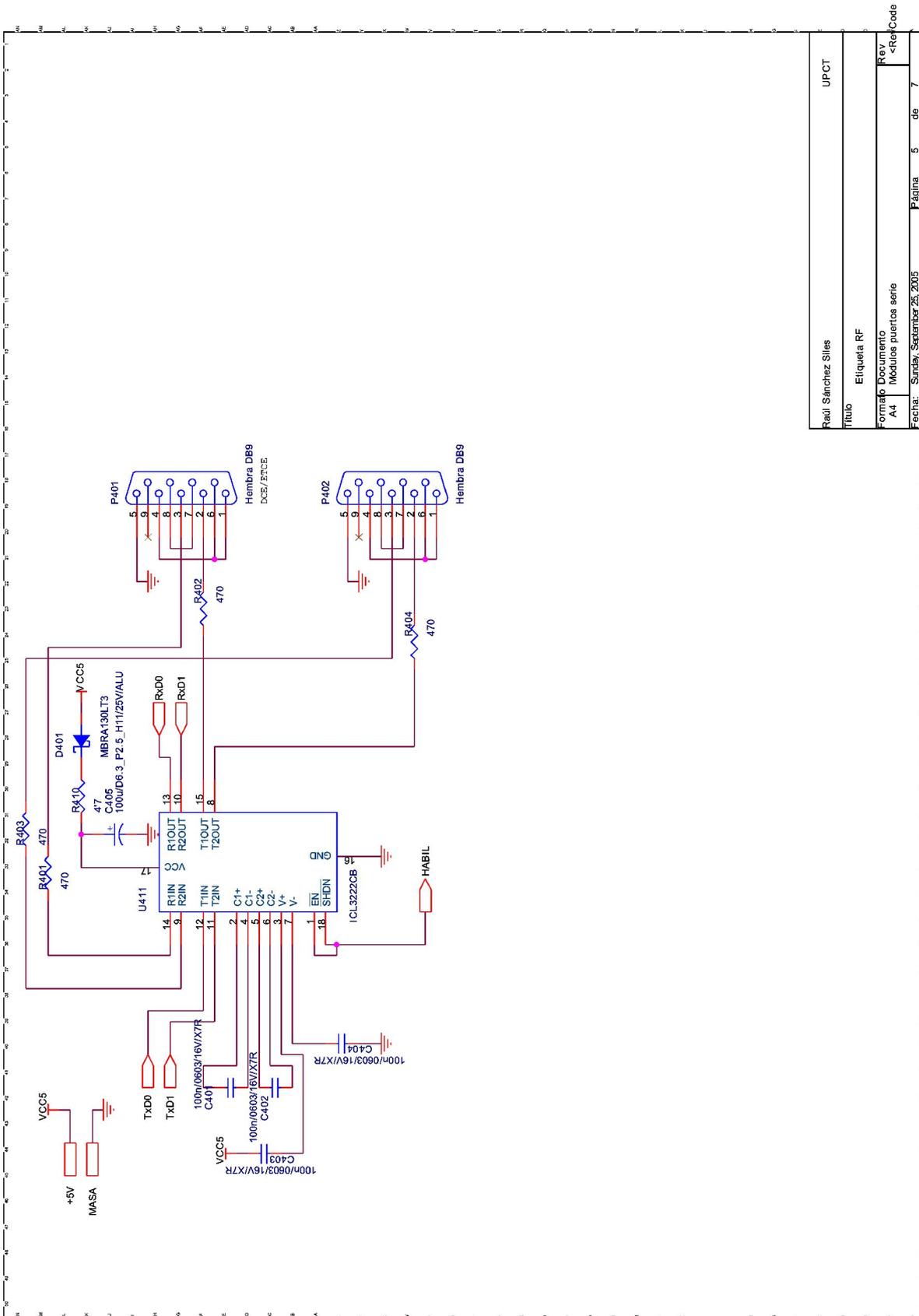


Raul Sánchez Siles		UPCT
Titulo Etiqueta RF		
Formato Documento A4 Módulo LCD		Rev <RevCode
Fecha: Sunday, September 26, 2005	Página 3	de 7

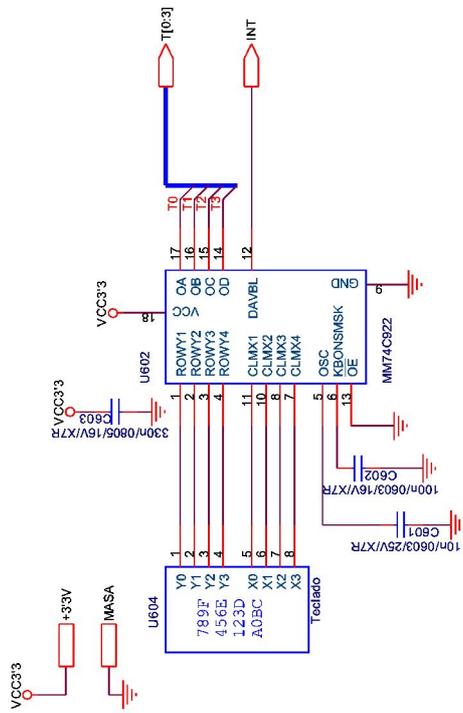


Se utiliza el convertor de niveles del módulo LCD para las líneas uc\_MISO y PAR\_MISO.

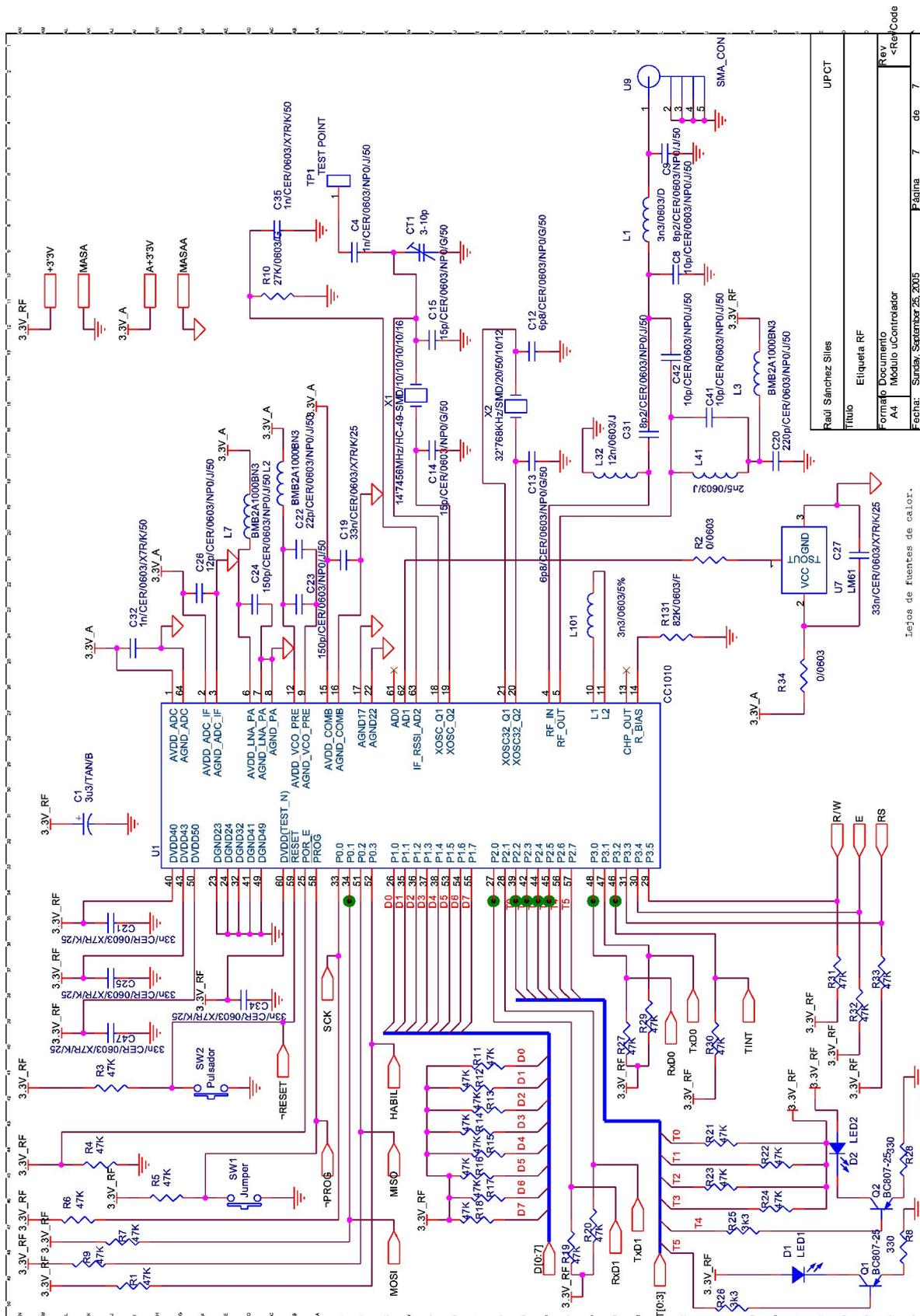
Raul Sánchez Siles		UPCT
Titulo		Etiqueta RF
Formato Documento	A4	Módulo de programación
Fecha:	Sunday, September 25, 2005	Página 4 de 7
Rev	<	RevCode



Raul Sánchez Siles		UPCT
Titulo		Etiqueta RF
Formato Documento		Rev
A4		<Rev
Fecha: Sunday, September 25, 2005		de
Página		7



Raul Sánchez Siles		UPCT
Etiqueta RF		
Formato Documento		Rev
A4 Módulo de teclado		<RevCode
Fecha: Sunday, September 25, 2005	Página 6	de 7



## Apéndice B. Código fuente.

```

/*****
*
*          *****
*          *****
*          ***      ***
*          ***    +++    ***
*          ***   ++   ***
*          ***   +
*          ***  ++   ***
*          ***   +++   ***
*          ***      ***
*          *****
*          *****
*
*****
* This program demonstrates the use of the SPP library, timers, and the
* serial port. The program is mainly interrupt driven.
*
* A pair of CC1010EBs running this program will function as a wireless
* null-modem cable, between serial port 0 on each board.
*
*
* +-----+           +-----+           +-----+           +-----+
* |      | UART0 RX |           |           | UART0 RX |           |
* |      |----->|           | RF/SPP with ACK |           |-----<|
* | PC |           | CC1010 |           | CC1010 |           | PC |
* |      | UART0 TX |           |           | UART0 TX |           |
* |      |-----<|           |           |           |           |----->|
* +-----+           +-----+           +-----+           +-----+
*
*          38400 baud           RF @ 19200 baud           38400 baud
*
*
* FEATURES:
*   - Max RS232 baud rate: 38400 (no parity bit)
*   - Max RF baud rate: 19200/NRZ
*   - Half-duplex
*   - Hardware flow control (CTS only)
*   - Automatic retransmission and transmission crash recovery through
*     randomized timeouts
*   - Serial input/output while the transceiver is active (through use of
*     interrupts)
*   - Automatic recalibration of the transceiver (every 2 minutes)
*
* EXPECTED PERFORMANCE (serial port data transfer):
*   - 1k Xmodem: >7000 bps with RF@9600 and UART@38400
*   - 1k Xmodem: >11500 bps with RF@19200 and UART@38400
*
* LED INDICATORS:
*   - Red:    Waiting for retransmission
*   - Yellow: Transmitting
*   - Green:  Toggled for each received packet
*   - Blue:   Hardware flow control activated (CTS)
*
* HOW IT WORKS:
* The main loop takes care of RF transmission and reception.
* The UART0 ISR takes care of UART transmission and reception.

```

```

*
* RF->UART:
*   When an RF packet has been received, the RF RX and UART TX buffer
*   are switched, and the packet data is transmitted over the UART
*   (after making sure that it's ready).
*
* UART->RF:
*   When the first byte is received, a start RF countdown begins. To
*   avoid buffer overruns the serial port flow control is enabled
*   after TX_BUFFER_LIMIT bytes. The RF TX buffer and the UART RX
*   buffers are switched, and the RF transmission begins. In the mean
*   while the UART is still ready for receptions (into the other
*   buffer).
*****
* Author:          JOL
*****
* Revision history:
* 1.0 2002/07/18      First public release
*
* $Log: rf232.c,v $
* Revision 1.5 2003/04/03 12:27:36  tos
* Bugfix:
* Disabled timer 3 interrupt in UART0_ISR (Caused the program to hang on
TX_BUFFER_LIMIT)
* Removed SPP timer function calls after recalibration (sppSetupRF).
*
* Revision 1.4 2003/02/03 08:25:33  tos
* Correction of UART flow control: activate flow control to avoid buffer
overrun.
*
* Revision 1.3 2003/01/30 07:29:19  tos
* Corrected recalibration procedure: re-initialization of function pointers
(Call Back).
*
* Revision 1.2 2002/11/20 14:10:00  kht
* Added startup macros
* Moved call to sppSetupRF to be called before setting callbacks, this was
* required due to change in SPP library
*
* Revision 1.1 2002/10/14 10:55:46  tos
* Initial version in CVS.
*
*
*****/

#include <chipcon/reg1010.h>
#include <chipcon/cc1010eb.h>
#include <chipcon/hal.h>
#include <chipcon/cul.h>
#include <stdlib.h>      // srand(), rand()

// MACRO: Switch two values without a third temporary variable
#define SWITCH(a,b) do { b^=a; a^=b; b^=a; } while (0)

// Buffer sizes
#define RX_BUFFER_LENGTH 255
#define TX_BUFFER_LENGTH 255

// Address definitions
#define DESTINATION_ADDRESS 255
#define MY_ADDRESS 255

```

```

// UART0 flow control
#define UART0_FLOW_CONTROL (P1_5)
#define STOP 1
#define READY 0
#define TX_BUFFER_LIMIT 150

// Data buffers
byte xdata pRXBuffer[2][RX_BUFFER_LENGTH]; byte rxDataLen[2]; // RF RX data
buffer, valid length (set by RF RX)
byte xdata pTXBuffer[2][TX_BUFFER_LENGTH]; byte txDataLen[2]; // RF TX data
buffer, valid length (set by UART RX)
byte xdata uartRXBuffer, uartTXBuffer, uartRXPos, uartTXPos, rfrRXBuffer,
rfTXBuffer;

// Transmission request flag
byte xdata txRequest;
#define TX_REQUEST_ON 1
#define TX_REQUEST_OFF 0

// Transmission timeout
word xdata startTXTimeout;
#define START_TX_TIMEOUT 7
#define SHORT_TX_TIMEOUT 2

// Recalibration request flag
byte xdata recalibRequest;
#define RECALIB_REQUEST_ON 1
#define RECALIB_REQUEST_OFF 0

// Recalibration timeout
word xdata recalibTimeout;
#define RECALIB_TIMEOUT 12000 // Every 2 minutes

// Reception timeout
#define NORMAL_TIMEOUT 100
#define RETRY_TX_TIMEOUT (1 + (rand() & 0x001F))

// Used to detect and ignore retransmissions (SPP_SEQUENCE_BIT)
byte xdata lastRXflags;

// SPP variables
SPP_SETTINGS xdata sppSettings;
SPP_RX_INFO xdata RXI;
SPP_TX_INFO xdata TXI;

// Function prototypes
void startTX (void);
void startRecalib (void);

//-----
// MAIN PROGRAM
//-----
void main() {

#ifdef FREQ868
    // X-tal frequency: 14.745600 MHz
    // RF frequency A: 868.277200 MHz Rx
    // RF frequency B: 868.277200 MHz Tx
    // RX Mode: Low side LO
    // Frequency separation: 64 kHz

```

```

// Data rate: 19.2 kBaud
// Data Format: NRZ
// RF output power: 4 dBm
// IF/RSSI: RSSI Enabled

RF_RXTXPAIR_SETTINGS code RF_SETTINGS = {
    0xA3, 0x27, 0x11,    // Modem 0, 1 and 2
    0x75, 0xA0, 0x00,    // Freq A
    0x58, 0x32, 0x8D,    // Freq B
    0x01, 0xAB,          // FSEP 1 and 0
    0x40,                // PLL_RX
    0x30,                // PLL_TX
    0x6C,                // CURRENT_RX
    0xF3,                // CURRENT_TX
    0x32,                // FREND
    0xFF,                // PA_POW
    0x00,                // MATCH
    0x00,                // PRESCALER
};
#endif

// Calibration data
RF_RXTXPAIR_CALDATA xdata RF_CALDATA;

// Initialize peripherals
WDT_ENABLE(FALSE);
// RLED_OE(TRUE); RLED = LED_OFF;
// YLED_OE(TRUE); YLED = LED_OFF;
// GLED_OE(TRUE); GLED = LED_ON;
// BLED_OE(TRUE); BLED = LED_OFF;

// Set optimum settings for speed and low power consumption
MEM_NO_WAIT_STATES();
FLASH_SET_POWER_MODE(FLASH_STANDBY_BETWEEN_READS);

// Local settings
sppSettings.myAddress = MY_ADDRESS;
sppSettings.rxTimeout = NORMAL_TIMEOUT;
sppSettings.txAckTimeout = 2;
sppSettings.txAttempts = 2;
sppSettings.txPreambleByteCount = 7;

// Setup serial port flow control
PORTDIRBIT(1, 5, POUT);
UART0_FLOW_CONTROL = READY;

// Initialize the buffer state variables
uartRXBuffer = 1;
uartTXBuffer = 1;
uartRXPos = 0;
uartTXPos = 0;
rfrXBuffer = 0;
rfTXBuffer = 0;

// RF setup
sppSetupRF(&RF_SETTINGS, &RF_CALDATA, TRUE);

// Initialize the SPP timer
sppStartTimer(CC1010EB_CLKFREQ);
SPP_INIT_TIMEOUTS();

// Initialize the start TX timeout
txRequest = TX_REQUEST_OFF;

```

```

startTXTimeout = 0;
sppSetTimerCB (SPP_CUSTOM_0_TIMER, startTX, &startTXTimeout);

// Initialize the recalibration timeout
recalibRequest = RECALIB_REQUEST_OFF;
recalibTimeout = RECALIB_TIMEOUT;
sppSetTimerCB (SPP_CUSTOM_1_TIMER, startRecalib, &recalibTimeout);

// Initialize the RF transmit structure
TXI.destination = DESTINATION_ADDRESS;
TXI.pDataBuffer = pTXBuffer[rfTXBuffer];
TXI.flags = SPP_ACK_REQ;
TXI.status = SPP_TX_FINISHED; // Nothing has gone wrong yet...

// Initialize the RF receive structure
RXI.maxDataLen = RX_BUFFER_LENGTH;
RXI.pDataBuffer = pRXBuffer[rfRXBuffer];
lastRXflags = TXI.flags | SPP_SEQUENCE_BIT;

// Initialize the random number generator (used to generate random
timeouts)
srand(0x1234);

// Serial port 0 setup
UART0_SETUP(38400, CC1010EB_CLKFREQ, UART_NO_PARITY | UART_RX_TX |
UART_ISR);

// Run forever
while (TRUE) {

// Receive a packet
if (sppReceive(&RXI) == SPP_RX_STARTED) {

// Wait for RF RX to complete
do {
if (txRequest && (TXI.status == SPP_TX_FINISHED) && (RXI.status
== SPP_RX_WAITING)) {
sppReset();
}
} while (SPP_STATUS() != SPP_IDLE_MODE);

// Check the results (ignore retransmitted messages)
if (RXI.status == SPP_RX_FINISHED) {
if (SPP_SEQUENCE_BIT & (lastRXflags ^ RXI.flags)) {

// Wait for UART TX to complete
while (uartRXPos != rxDataLen[uartRXBuffer]);

// Only process packets with payload data
if ((rxDataLen[rfRXBuffer] = RXI.dataLen) != 0) {

// Switch buffers and initiate UART0 transmission
UART0_SEND(pRXBuffer[uartRXBuffer][uartRXPos]);

//GLED =
~GLED;

SWITCH(uartRXBuffer, rfRXBuffer);
RXI.pDataBuffer = pRXBuffer[rfRXBuffer];
uartRXPos = 0;
UART0_SEND(pRXBuffer[uartRXBuffer][uartRXPos]);
}
}
lastRXflags = RXI.flags;
}
}

```

```

    }
}

// Transmit packet, if requested
if (txRequest) {
    if (sppSend(&TXI) == SPP_TX_STARTED) {

        // Wait for the transmission to finish
        //YLED = LED_ON;
        while (SPP_STATUS() != SPP_IDLE_MODE);
        //YLED = LED_OFF;

        // Check out the results
        if (TXI.status == SPP_TX_FINISHED) {
            sppSettings.rxTimeout = NORMAL_TIMEOUT;
            //RLED = LED_OFF;
            txRequest = TX_REQUEST_OFF;
        } else {
            //RLED = LED_ON;
            sppSettings.rxTimeout = RETRY_TX_TIMEOUT;
        }
    }
}

// Recalibrate the transceiver if requested to
if (recalibRequest) {
    sppSetupRF(&RF_SETTINGS, &RF_CALDATA, TRUE);
}
} // main

```

```

//-----
// Start transmission (called by TIMER3_ISR(), in sppTimer.c)
//-----
void startTX() {

    if (!txRequest) {
        SWITCH(uartTXBuffer, rfTXBuffer);
        TXI.pDataBuffer = pTXBuffer[rfTXBuffer];
        TXI.dataLen = uartTXPos;
        uartTXPos = 0;
        UART0_FLOW_CONTROL = READY; BLED = LED_OFF;
        txRequest = TX_REQUEST_ON;
    } else {
        startTXTimeout = SHORT_TX_TIMEOUT;
    }
} // startTX

```

```

//-----
// Recalibrate the transceiver (called by TIMER3_ISR(), in sppTimer.c)
//-----
void startRecalib (void) {
    recalibRequest = RECALIB_REQUEST_ON;
    recalibTimeout = RECALIB_TIMEOUT;
} // recalibrateRF

```

```

//-----
//  UART 0 interrupt service routine:
//-----
void UART0_ISR () interrupt INUM_UART0 {

    // Byte sent (RF RX)
    if (TI_0) {
        INT_SETFLAG (INUM_UART0_TX, INT_CLR);

        // Push bytes until the buffer is empty
        if (++uartRXPos < rxDataLen[uartRXBuffer]) {
            UART0_SEND(pRXBuffer[uartRXBuffer][uartRXPos]);
        } else {
            uartRXPos = rxDataLen[uartRXBuffer];
        }
    }

    // Byte received (RF TX)
    if (RI_0) {
        INT_SETFLAG (INUM_UART0_RX, INT_CLR);
        if (uartTXPos < TX_BUFFER_LENGTH) {
            pTXBuffer[uartTXBuffer][uartTXPos++] = UART0_RECEIVE();

            // Schedule a start TX callback when the first data byte arrives
            through the UART
            INT_ENABLE (INUM_TIMER3, INT_OFF);
            if (!startTXTimeout) {
                startTXTimeout = START_TX_TIMEOUT;
            } else {
                if (uartTXPos == TX_BUFFER_LIMIT) {
                    // Prevent buffer overrun by turning CTS off (RTS at our
                    end of the cable)
                    UART0_FLOW_CONTROL = STOP; BLED = LED_ON;
                    startTXTimeout = SHORT_TX_TIMEOUT;
                }
            }
            INT_ENABLE (INUM_TIMER3, INT_ON);
        }
    }
} // UART0_ISR

```