

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica



Diseño e Implementación de una estación de trabajo prototipo para desarrollo de sistemas empotrados en RidgeRun Limitada.

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura

RidgeRun Limitada

Erick Sanabria Calvo

Cartago, 21 Octubre de 2008

INSTITUTO TECNOLÓGICO DE COSTA RICA

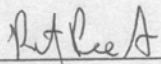
ESCUELA DE INGENIERÍA ELECTRÓNICA

PROYECTO DE GRADUACIÓN

TRIBUNAL EVALUADOR

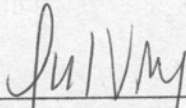
Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



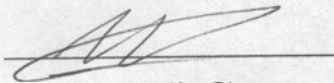
Ing. Roberto Pereira Arroyo

Profesor lector



Ing. Paola Vega Castillo

Profesora lectora



Ing. Adolfo Chaves

Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago 21 de Octubre de 2008

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 21 de Octubre

*Erick Sanabria Calvo*

Erick Sanabria Calvo

Céd: 3-383-611

## RESUMEN

RidgeRun Engineering Limitada ubicada en Curridabat, cantón de San José, tiene como actividad comercial principal, el desarrollo de sistemas empotrados.

Para esta actividad, la empresa requiere del empleo de múltiples herramientas de desarrollo, que por lo general tienen un alto precio en el mercado. Esta situación impide a la empresa a adquirir una suficiente cantidad de herramientas, para así dotar a cada uno de sus empleados con todas las herramientas que requieren para ejecutar sus actividades diarias.

Se han buscado formas de atenuar el faltante de herramientas mediante planes de coordinación entre los empleados para organizar el tiempo que cada uno usa una herramienta, sin embargo no se ha logrado eliminar el problema completamente. Esto ha dado pie a que RidgeRun plantee la posibilidad de realizar un diseño propietario que resuelva el problema mediante la integración de las herramientas con mayor demanda en un único dispositivo.

Es por esto que RidgeRun toma la decisión de financiar el presente proyecto, no sólo con la intención de resolver un problema existente, sino también con la idea de incursionar en el mercado comercial de los sistemas empotrados brindando a desarrolladores, la posibilidad de contar con una herramienta de desarrollo para sistemas empotrados muy bajo costo.

Es importante resolver el problema presente en RidgeRun, dado que el faltante de herramientas para desarrollo, genera demoras en la entrega de proyectos terminados a los clientes de la empresa.

**Palabras Clave:** Sistemas Empotrados, herramienta de desarrollo, RidgeRun

## **ABSTRACT**

The lack of a tool in RidgeRun Ltda, specially intended for embedded systems development has motivated the design and creation of an electronic device in where the most used tools are grouped together into a workstation for Linux,Windows and Mac OsX environments, in order to simplify the development and debug processes

The design offers to the user a great flexibility to interconnect modules and combine its properties if required. Also additional modules can be added for user's specific functions.

A detailed documentation is provided including the most important reasons considered for every module's design.

**Keywords:**Embedded systems, development tool , RidgeRun

## **DEDICATORIA**

**A Dios Todo Poderoso:** quien me dio la fortaleza en todo momento para cumplir esta meta.

**A mi familia:** quienes siempre me dieron su apoyo y me permitieron avanzar.

**À Mariana:** quem esteve do meu lado durante toda a minha vida universitária, me deu palavras de ânimo, companhia em todo momento e nunca me negou seu apoio incondicional em cada etapa da minha carreira profissional. Graças a ela, este esforço hoje é uma realidade e posso assim cumprir com mais uma etapa da minha vida.... Obrigado!

## **AGRADECIMIENTO**

A mi amigo y hermano José Manuel Peralta Carranza por su amistad, sus consejos y su invaluable apoyo a lo largo de toda la carrera.

A mi amigo José Umaña Aguilar por su gran apoyo que esta meta fuera una realidad.

Al personal de RigeRun, pero especialmente a los ingenieros Jorge Arrieta, Marco Villalta y Diego Dompe por su apoyo en este proyecto.

## INDICE GENERAL

CAPÍTULO 1: Introducción .....	15
1.1 Problema Existente e importancia de su solución .....	15
1.2 Solución Seleccionada .....	17
CAPITULO 2: Meta y Objetivos .....	20
2.1 Meta .....	20
2.2 Objetivo General .....	20
2.3 Objetivos Específicos .....	20
CAPÍTULO 3: Marco Teórico.....	21
3.1 Descripción del proceso por mejorar.....	21
3.2 Antecedentes Bibliográficos .....	22
3.2.1 Microcontrolador.....	22
3.2.2. Comunicación.....	23
3.2.2.1 USB.....	24
3.2.2.2 Protocolo SPI .....	26
3.2.2.3. JTAG .....	27
3.2.3 Medición de corriente .....	29
3.3 Descripción de los principales principios de software, electrónicos y físicos relacionados con la solución del problema.....	29
3.3.1 Principios de software relacionados .....	30
3.3.2 Principios electrónicos relacionados .....	30
3.3.2.1 Divisor de tensión .....	30
3.3.3 Principios físicos relacionados .....	31
3.3.3.1Efecto Hall .....	31
CAPITULO 4: Procedimiento Metodológico .....	33
4.1 Reconocimiento y definición del problema .....	33
4.2 Obtención y análisis de la información .....	34
4.3 Evaluación de las alternativas y síntesis de una solución.....	36
4.4 Implementación de la solución .....	37
4.5 Reevaluación y rediseño .....	38
CAPITULO 5. Descripción detallada de la solución .....	39



5.1 Análisis de soluciones y selección final.....	39
5.1.1 Soluciones propuestas .....	39
5.1.1.1 Propuestas para los módulos presentes en el sistema por desarrollar .....	39
5.1.1.2 Escogencia de las propuestas para los módulos que integrarán el sistema por desarrollar. ....	40
5.1.2 Propuestas de funcionamiento para algunos de los módulos incluidos. ....	41
5.1.2.1 Circuito de Fuente de alimentación CD .....	41
5.1.2.2 Propuestas para el circuito de fuente de alimentación CD.....	42
5.1.2.3 Circuito de interfaz de transmisión de datos/acople con el computador....	44
5.1.2.4 Interfaz sobre la cual se basa la posibilidad de expansión del sistema....	44
5.2 Descripción del Hardware .....	46
5.2.1 Módulo USB .....	47
5.2.1.1 Circuito Concentrador (Hub) USB .....	47
5.2.1.2 Circuito de interfaz USB con el computador.....	50
5.2.1.3 Circuito de Módulo Serial. ....	55
5.2.1.4 Circuito de Módulo de Control. ....	58
5.2.1.4.1 Submódulo Voltímetro .....	62
5.2.1.4.2 Submódulo Amperímetro.....	64
5.2.1.4.3 Submódulo LCD .....	66
5.2.1.5 Circuito de Módulo de Jtag.....	67
5.2.1.6 Circuito de Módulo de Expansión.....	71
5.2.1.7 Circuito de Módulo de Fuente CD. ....	74
5.3 Descripción del software .....	78
5.3.1 Programa Usuario. ....	78
5.3.2 Programación del circuito FT2232D .....	79
5.3.3 Programación del Microcontrolador.....	80
CAPÍTULO 6. Análisis de Resultados .....	87
6.1 Resultados Obtenidos .....	87
6.1.1 Módulo USB y Módulo Serial.....	88
6.1.2 Módulo de Control .....	89
6.1.2.1 Módulo de Control y submódulo de Voltímetro.....	89
6.1.2.2 Módulo de Control y submódulo de LCD.....	95
6.1.2.3 Módulo de Control y submódulo de Amperímetro .....	96

6.1.3 Módulo de JTAG .....	101
6.1.4 Módulo de Fuente CD .....	105
6.1.5 Prototipo Implementado .....	107
6.1.6 Circuito Impreso .....	110
6.2 Análisis de Resultados .....	112
6.2.1 Módulo USB y Módulo Serial.....	112
6.2.2 Módulo de Control .....	112
6.2.2.1 Módulo de Control y submódulo de Voltímetro.....	113
6.2.2.2 Módulo de Control y submódulo de Amperímetro .....	114
6.2.2.3 Módulo de Control y submódulo de LCD.....	115
6.2.3 Módulo de JTAG .....	115
6.2.4. Módulo de Fuente de CD .....	116
6.2.5. Margen de Error del Sistema.....	116
CAPÍTULO 7. Conclusiones y recomendaciones .....	118
7.1 Conclusiones.....	118
7.2 Recomendaciones.....	118
8 Bibliografía .....	119
9 APENDICES.....	121
Apéndice A1. Glosario.....	121
Apéndice A 2. Software desarrollado para el microcontrolador.....	124
Apéndice A 3 . Script desarrollado para realizar multiples lecturas con el voltímetro.....	136
Apéndice A 4 . Script desarrollado para realizar múltiples lecturas con el voltímetro.....	136
Apéndice A 5. Script desarrollado para realizar múltiples lecturas con el amperímetro.....	136
Apéndice A 6. Código para generar el archivo de configuración “dlp2232.cfg” de OpenOCD para que reconozca el dispositivo JTAG desarrollado.....	137
Apéndice A 7. Lista Materiales y cotización de la producción del dispositivo desarrollado.....	138
Apéndice A. 8 Información del Proyecto .....	140
Apéndice A 9 Breve resumen del circuito, esquemático final y circuito capas del circuito impreso .....	141

## INDICE DE FIGURAS

Figura 1.1 Disposición de las herramientas previo al desarrollo del proyecto .....	18
Figura 1.2 Disposición de las herramientas de acuerdo a la solución planteada ..	19
Figura 3.1: Esquema de organización de dispositivos bajo el protocolo SPI .....	27
Figura 3.2: Detalle de un punto de prueba en un circuito integrado .....	28
Figura 3.3 Esquema de un divisor de tensión .....	31
Figura 3.4 Principio del efecto Hall .....	32
Figura 4.1 Tarjeta de desarrollo empleada en RidgeRun .....	37
Figura 5.1 Diagrama de bloques del sistema desarrollado .....	46
Figura 5.2 Esquema interno del módulo USB .....	47
Figura 5.3 Circuito correspondiente al concentrador USB .....	49
Figura 5.4. Circuito de Interfaz USB .....	54
Figura 5.5 Circuito de módulo Serial .....	57
Figura 5.6 Circuito del módulo de Control .....	62
Figura 5.7 Circuito divisor de tensión como acondicionador de señal .....	63
Figura 5.8. Circuito sensor de corriente .....	65
Figura 5.9 Circuito de conexión al módulo de LCD .....	67
Figura 5.10 Conexión ARM9 JTAG .....	69
Figura 5.11 Distribución de pines para la conexión Voco ARM9 .....	69
Figura 5.12 Circuito del módulo JTAG .....	71
Figura 5.13 Módulo de Expansión .....	73
Figura 5.14 Circuito del módulo de Fuente CD .....	77
Figura 5.15 Captura de pantalla de la forma en que el usuario vería los datos mediante el empleo de Minicom .....	79
Figura 5.16 Pantalla correspondiente al software MPROG para programar el circuito integrado FT2232D .....	80
Figura 5.17 Diagrama de flujo del software desarrollado .....	1
Figura 6.1 Diagrama de bloques del sistema implementado .....	87
Figura 6.2 Dispositivo para prototipado DLP2232M .....	88
Figura 6.3 Dato enviado por el puerto USB después de ser serializado .....	89
Figura 6.4 Dato ASCII enviado por el puerto USB correspondiente a una letra “a”	

.....	90
Figura 6.5 Dato correspondiente a la primera posición del valor de tensión medida (3 en ASCII).....	92
Figura 6.7 Dato correspondiente a la tercera posición del valor de tensión medida (5 en ASCII).....	93
Figura 6.9 Dato correspondiente al carácter “m” en ASCII para indicar la escala en que está dado el dato.....	94
Figura 6.11 Datos de las distintas mediciones descritos en la tabla 6.1 enviados al PC.....	95
Figura 6.12 Despliegue en pantalla de una tensión medida con el voltímetro .....	96
Figura 6.13 Valores de corriente sensados con el módulo de amperímetro .....	101
Figura 6.14 Captura en pantalla de la primera terminal empleada.....	103
Figura 6.15 Captura de pantalla de la segunda terminal empleada .....	104
Figura 6.16 Captura de pantalla de la tercera terminal empleada.....	105
Figura 6.17 Sistema prototipo implementado .....	107
Figura 6.18 Prototipos de fuente de CD, Amperímetro y Voltímetro .....	108
Figura 6.19 Módulo de JTAG conectado a un dispositivo en desarrollo.....	109
La figura 6.20 muestra el diseño del circuito impreso correspondiente a la capa donde se ubican los componentes.....	110
Figura 6.20 Diseño del circuito impreso correspondiente a la capa de componentes .....	110
Figura 6.21 Diseño del circuito impreso correspondiente a la capa de cobre .....	111

## INDICE DE TABLAS

Tabla 5.1: Propuestas para los módulos que integrarán el sistema por desarrollar .....	40
Tabla 5.2: Cuadro comparativo de propuestas para los módulos que integrarán el sistema por desarrollar.....	43
Tabla 5.3: Propuestas para interfaz de comunicación sobre la cual se basa la posibilidad de expansión en el futuro. ....	45
Tabla 5.4: Distribución de pines del microcontrolador presente en el módulo de Control.....	58
Tabla 5.5: Conexión de los pines para el cable de interfaz .....	70
Tabla 5.6: Valores de tensión de salida para diferentes valores de resistencia R2 empleados.....	75
Tabla 5.7: Ubicación de jumpers para la selección de la tensión de salida de la fuente CD en los pines del conector P13 .....	76
Tabla 5.8: Resumen de los procedimientos desarrollados en el software del microcontrolador.....	81
Tabla 6.1 Tabla comparativa de valores de tensión obtenidos al emplear el voltímetro diseñado y un multímetro de origen comercial Radio Shack modelo 22-811 .....	91
Tabla 6.2: Tensión de salida del sensor ACS712ELCTR-20A-T para diferentes valores de corriente.....	97
Tabla 6.3: Tensión de salida del sensor ACS712ELCTR-20A-T para diferentes valores de corriente después de verificar las conexiones del sensor de corriente	98
Tabla 6.4: Corriente registrada para distintos valores de tensión aplicada a la terminal AMPMETER .....	100
Tabla 6.5: Tensiones predefinidas de salida del módulo Fuente CD sin carga...	106

Tabla 6.6: Tensiones predefinidas de salida del módulo Fuente CD con carga $R_L=2.2K\Omega$ .....	106
Tabla 6.7: Tensiones predefinidas de salida del módulo Fuente CD con carga $R_L=220\Omega$ .....	107
Tabla 6.8: Dimensiones finales del circuito impreso diseñado .....	111

## **CAPÍTULO 1: Introducción**

### **1.1 Problema Existente e importancia de su solución**

En 2005 RidgeRun limitada, empresa de origen norteamericano inicia sus actividades comerciales en Costa Rica, desarrollando software para sistemas empujados. En ese entonces únicamente se requerían los módulos de trabajo provistos por los clientes. Con el pasar del tiempo, y la expansión de la empresa, los productos que se desarrollan y las técnicas de trabajo han variado considerablemente hasta el día de hoy, y por ende los dispositivos que permiten: la medición de parámetros eléctricos, la comunicación entre los sistemas en desarrollo y los computadores, Ho bien los que permiten realizar depuraciones de software mediante terminales de uso general (GPIO),son los que se han convertido en elementos indispensables de trabajo y con una alta demanda de uso. Estas herramientas generalmente por las acciones técnicas que permiten realizar, tienen un alto valor económico que hace imposible que la empresa pueda adquirir una cantidad suficiente de unidades de cada dispositivo para así de ésta forma dotar a cada empleado involucrado en el proceso de desarrollo con las herramientas que requiere.

Previo al desarrollo del presente proyecto, cuando una herramienta era requerida por un desarrollador, éste tenía que dirigirse al departamento de hardware y hacer la solicitud. Si la herramienta se encontraba desocupada, inmediatamente se le asignaba, pero si la herramienta estaba siendo ocupada, o bien había más de una persona interesada en la misma, se tenía como prioridad otorgarle la herramienta a la persona que más la requiriese. Sin embargo, cuando la prioridad era la misma para ambas personas, ambos debía llegar a un acuerdo en el tiempo de uso de la herramienta con el fin de evitar retrasos en el desarrollo de las actividades laborales.

En algunos casos los desarrolladores, una vez que obtenían la herramienta que requerían, debían lidiar con problemas sencillos pero molestos, como: configurar

correctamente los dispositivos electrónicos, colocarlos de forma adecuada en sus puestos de trabajo, ubicar los cables y accesorios específicos para las actividades que seguidamente ejecutarían, etc.

La organización en el uso de herramientas como alternativa de solución al problema de carencia de las mismas, no siempre fue la mejor, dado que no sólo dejaba abierta la posibilidad de insertar retrasos en las actividades de desarrollo, sino también causar daños a los equipos, y generación de conflictos laborales (al excederse un desarrollador el tiempo disponible de uso de una herramienta asignada de alta demanda).

Dado lo anterior, las razones por las cuales la empresa necesita solucionar el problema son las siguientes:

- Eliminar las causas de retrasos en el desarrollo de sistemas empotrados, relacionadas con un faltante de herramientas
- Disminuir la probabilidad de daño en los costosos equipos debido a la constante manipulación y movimiento de un lugar a otro
- Evitar cualquier tipo de conflicto laboral-personal por consecuencia de éste faltante de herramientas.

Mediante el diseño de un dispositivo electrónico de bajo costo, se busca solucionar el problema antes planteado, de forma que con él, la mayoría de los problemas de requerimiento de equipo por parte de los empleados sean solucionados, y consecuentemente también todos los subproblemas derivados. También es necesario mencionar que al contar la empresa con un diseño propio de un dispositivo, los costos en cuanto a reparación y mantenimiento de los mismos, serán ampliamente reducidos por ser un sistema conocido.



## **1.2 Solución Seleccionada**

La presente sección tiene el objetivo de darle al lector una idea breve de los factores que se tomaron en cuenta a la hora de seleccionar la propuesta que se le hizo a la empresa

Para plantear una posible solución al problema documentado en la sección anterior, se debe tomar en cuenta los requerimientos y/o inquietudes de la empresa, y con base en esto canalizar la solución, para que ésta sea lo más compatible con las necesidades de la misma. La posible solución que se plantea debe estar orientada bajo la misma filosofía de código libre por la cual se rige RidgeRun Limitada.

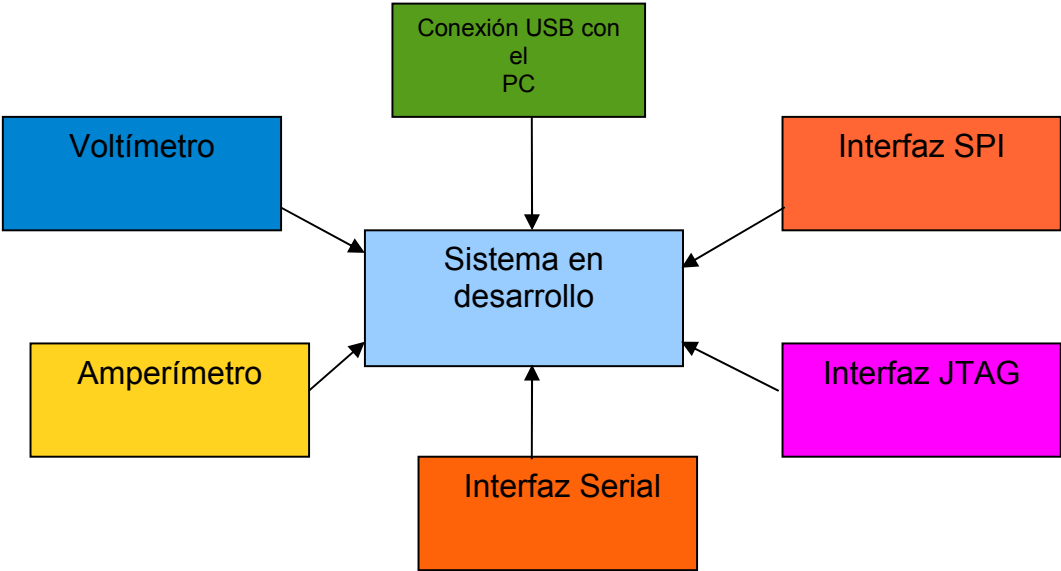
Para que la solución sea aprobada por la empresa, debe abarcar las siguientes necesidades:

- La propuesta debe estar enfocada a reducir demoras en la ejecución de tareas por falta de equipo.
- La inversión económica para producirla debe ser de bajo nivel.
- La propuesta debe dar capacidad a la empresa para dotar a la mayor cantidad de empleados con las herramientas que requieren
- La propuesta debe permitir modificaciones para el mejoramiento del equipo a futuro, sin que esto demande una alta inversión.
- Debe tomar en cuenta la comodidad del empleado durante su jornada laboral, de forma que no comprometa el desempeño del trabajador
- La propuesta de solución debe garantizar o permitir un fácil mantenimiento por parte del personal de la empresa.

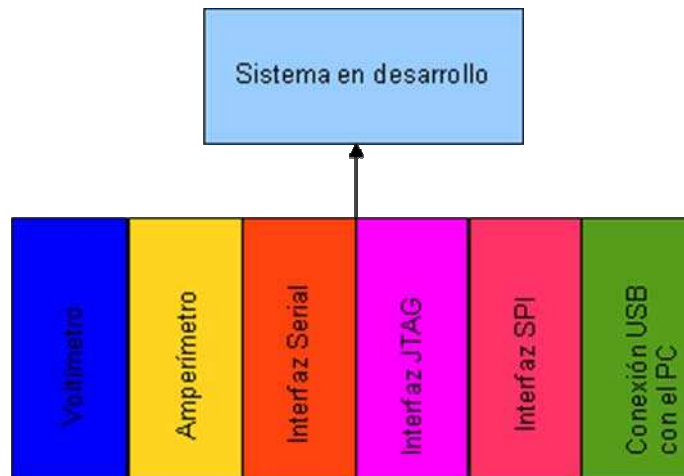
Dados los anteriores requerimientos de la empresa, se sugiere como posible solución, el diseño de un dispositivo electrónico, capaz de agrupar las herramientas de mayor demanda en la menor área posible (un único dispositivo como tal), de forma que como resultado se obtenga un dispositivo electrónico multiherramienta, pequeño y de bajo costo con capacidad para ser modificado y

mejorado en el futuro sin la necesidad de invertir una fuerte suma de dinero. Para que la propuesta tenga mayor calado, y no sólo cumpla, sino que supere las expectativas de la empresa, el dispositivo sugerido debe ser fácil de construir, con el afán de reproducirlo cuantas veces sea necesario para así fácilmente dotar a cada empleado con uno de estos dispositivos.

Ahora bien, para que la solución propuesta sea compatible con futuras modificaciones, el diseño debe ser compatible con la incorporación de nuevos módulos, para realizar alguna función específica no determinada aún(los módulos por agregar estarán sujetos a la capacidad de expansión y mejoramiento del dispositivo desarrollado)



**Figura 1.1 Disposición de las herramientas previo al desarrollo del proyecto**



**Figura 1.2 Disposición de las herramientas de acuerdo a la solución planteada**

Para tener una idea clara sobre la solución que se plantea, se pueden observar las figuras 1.1 y 1.2.

La figura 1.1, ejemplifica el sistema actual. Se muestra la disposición de las herramientas de desarrollo. En esta figura se pretende demostrar al lector que cada una de las herramientas que se deben utilizar para un procedimiento específico, es un módulo independiente que no guarda relación alguna con los demás. Esto fácilmente complica el empleo de las herramientas y la obtención de resultados cuando se requiere emplear varias de estas en forma conjunta.

En la figura 1.2 se puede observar que las herramientas que se requieren para el desarrollo están agrupadas en un mismo módulo, haciendo más fácil, rápido y seguro su empleo ya que el usuario en un mismo lugar posee las herramientas más importantes de uso diario y además no debe lidiar con conexiones de cables, que en la mayoría de situaciones son las que dificultan emplear las herramientas en forma conjunta.

## **CAPITULO 2: Meta y Objetivos.**

En el presente capítulo se expondrá cual será la meta del proyecto, y el alcance del mismo.

El alcance del proyecto está definido con base en los objetivos generales y específicos

### **2.1 Meta**

Proveer a RidgeRun Engineering Limitada de una herramienta prototipo para el desarrollo de sistemas empotrados.

### **2.2 Objetivo General**

Diseñar e implementar un prototipo de un sistema multiherramienta integrado para el desarrollo de sistemas empotrados.

### **2.3 Objetivos Específicos**

- Lograr una adecuada funcionalidad de cada uno de los módulos diseñados y construidos, de forma que el funcionamiento de los mismos sea similar a las herramientas producidas en forma comercial y adquirida por la empresa.
- Depurar el funcionamiento del sistema mediante la ejecución en repetidas ocasiones y bajo diferentes condiciones de todos los procesos de desarrollo para los cuales se planea emplear esta nueva herramienta.
- Desarrollar un software básico que permita configurar y utilizar los dispositivos desarrollados.
- Presentar una documentación exhaustiva del sistema diseñado, en donde se incluya un diagrama esquemático completo, con la finalidad de que la empresa a partir de éste pueda realizar un circuito impreso.

## **CAPÍTULO 3: Marco Teórico.**

### **3.1 Descripción del proceso por mejorar.**

Independientemente de las labores que un desarrollador tenga asignadas, o a cualquiera de los equipos de trabajo de la empresa a los cuales pertenezca un desarrollador, cuando este requiere utilizar una herramienta para alguna de sus asignaciones, debe dirigirse al departamento de Hardware y hacer la solicitud del equipo que requiere utilizar. Si el equipo está disponible, el interesado puede tomarlo, y llevarlo a su lugar de trabajo, en este caso su respectivo escritorio. Antes de poder emplear la herramienta, el desarrollador debe verificar las condiciones de conexión del dispositivo (tensión de entrada y polaridad), y además si los cables que le fueron provistos, son los que deben ser empleados.

En caso de que la herramienta no esté disponible, el interesado dependiendo de sus necesidades y sus prioridades, puede esperar un tiempo prudencial, o bien coordinar con la persona que esté ocupando en ese momento la herramienta en cuestión, con tal de agilizar el proceso para desocuparla lo antes posible.

Este procedimiento a simple vista, tiende a ser considerado como sencillo, sin embargo tiende a complicarse cuando hay más de una persona interesada en una herramienta para utilizarla por espacios de tiempo moderados, o bien el desarrollador se encuentra en una posición en la cual debe trabajar contra reloj.

Con el actual procedimiento existente para la solicitud de una herramienta, el interesado se puede ver afectado no sólo por retrasos dada la no disponibilidad de la herramienta, sino también por algunos factores de riesgo claramente asociados con las condiciones de trabajo, como lo son por ejemplo: conexión errónea de la polaridad del dispositivo, selección incorrecta de la tensión de entrada del dispositivo, o bien simplemente una incorrecta ubicación del dispositivo que generalmente con lleva a un daño físico en alguno de las sistemas que se tengan en la cercanía.

Por otra parte, cuando alguna de las herramientas de gran demanda en la empresa falla, sin importar la causa de la misma, el departamento de Hardware es el responsable de realizar las reparaciones, sin embargo si el repuesto requerido

no se encuentra disponible, la herramienta se vuelve inutilizable hasta que se obtengan las partes requeridas, y por ende se origina un retraso en los proyectos en los que se trabaja empleando esta herramienta.

En el párrafo anterior, el lector podrá haber notado que existe una dependencia muy estrecha entre la funcionalidad de las herramientas con las que cuenta la empresa, y los objetivos de producción, razón por la cual se busca modificar el proceso de producción, de manera que se pueda eliminar totalmente esta dependencia.

### **3.2 Antecedentes Bibliográficos**

Dado el carácter y la finalidad del proyecto, es necesario realizar una búsqueda extensiva, con el fin de conocer si se han realizado diseños previos similares, y además conocer tecnologías y componentes que eventualmente se podrían tomar en cuenta para un óptimo resultado. Cabe resaltar, que dada la tendencia del proyecto, y el área donde se desarrolla, la fuente de información principal será internet, dado la probabilidad de encontrar información actual es mucho más alta que si la investigación se basa en bibliografía impresa.

Seguidamente se muestran los resultados más importantes obtenidos

#### **3.2.1 Microcontrolador**

Después de analizar y comparar diferentes dispositivos posibles se concluyó tomando en cuenta la relación: precio/funciones, que la mejor manera de estructurar el sistema, era a partir de un microcontrolador.

El microcontrolador seleccionado sobre el cual el sistema estaría basado es el MSP430FG439 de Texas Instruments. Su elección se basó principalmente a su característica de bajo consumo de corriente, y su funcionamiento con lógica LVTTTL. En estado inactivo, el consumo de corriente de este microcontrolador puede ser menor a 1 micro amperio, lo que favorece el desarrollo de una herramienta, tomando en cuenta que esta piensa ser alimentada por puerto USB [1]

Otras de las características que también justifican la elección de este modelo de microcontrolador son:

- 2KB de memoria RAM
- ADC con capacidad para resolución de hasta 12 bits
- Encapsulado LQFP80 (su tamaño facilita la integración en espacios pequeños).

La familia MSP430 a la cual pertenece este microcontrolador, está enfocada al desarrollo de sistemas portátiles, lo cual la hace muy apropiada para los fines de este proyecto. Por otra parte, al contar con un microcontrolador en el diseño, se puede centralizar el control de las actividades propias del sistema, lo cual disminuye la complejidad en caso de querer realizarse una modificación.

En cuanto al desarrollo del software para el microcontrolador, existen varias opciones que se pueden utilizar. Las más conocidas son:

- Code Composer
- IAR
- MSPGCC

Se debe mencionar que tanto Code Composer como IAR, son herramientas de desarrollo de software propietarias, mientras que MSPGCC es una herramienta basada en código libre [2].

Las diferencia para herramientas de programación propietarias y herramientas basadas en código libre radica básicamente en el soporte que el creador le da a éstas. Por tanto ante el empleo de una herramienta basada en código libre, el desarrollador debe estar consciente de los retos que esto conlleva dada poca existencia documentación y soporte.

### **3.2.2. Comunicación**

Al no existir un protocolo de comunicación predefinido para ser usado con dispositivos empotrados, queda a criterio del diseñador, la selección del protocolo de comunicación que un sistema empotrado usará. Por esta razón se torna difícil el implementar un único protocolo de comunicación, ya que la alternativa de

solución que se pretende diseñar, se limitaría a un sólo tipo de dispositivos.

Por esta razón, y después de indagar con algunos fabricantes de componentes electrónicos, se halló el circuito integrado FT2232 de Future Technology Devices International (FTDI). Este chip, en una primera instancia es un convertidor del protocolo USB a los protocolos serie y paralelo, que además cuenta con las siguientes características [3]:

- Doble canal configurable para modo serie o paralelo
- Implementación del protocolo JTAG
- Implementación del protocolo SPI
- Implementación del protocolo I2C

La característica principal que posee este circuito integrado, y que lo hace sobresaliente en comparación con las ofertas existentes en el mercado es que la implementación del protocolo USB no requiere de diseño de controlador alguno, puesto que el firmware con que es dotado el dispositivo, es capaz de realizarlo.

### **3.2.2.1 USB**

El estándar USB (Bus en Serie Universal) es un estándar concebido por un conjunto de empresas, que surgió como solución a la necesidad de estandarizar la forma de conectar dispositivos al computador sin tener que emplear tarjetas adicionales para ello o de reiniciar el computador para que estos fuesen detectados.

Las principales características que hacen a este estándar muy utilizado son:

- Transferencias de información en 3 distintas velocidades: Low (1.5Mbit/s), Full (12Mbit/s) y High (480 Mbit/s)
- Capacidad de conectar hasta 127 periféricos en un mismo bus
- Control de errores en la transferencia de información
- En cuanto a alimentación se refiere, el estándar proporciona la tensión de alimentación para los dispositivos que se deseen conectar siempre y cuando no consuman una corriente superior a 500mA [4]



- Capacidad de control de consumo, que permite poner en reposo o en marcha un determinado dispositivo, de forma que el control de consumo de corriente puede ser manipulable

A manera de funcionamiento del protocolo USB, el *host* o anfitrión USB que siempre es el computador, es el encargado de manejar toda la complejidad del protocolo, por lo que permite simplificar el funcionamiento de los dispositivos esclavos o periféricos.

Las transacciones USB son hechas a través de paquetes de información. Cada transacción está compuesta de las fases:

- **Fase Token o señalización:** el *host* indica el tipo de la futura transacción.
- **Fase de datos:** El dato es transmitido a través de un paquete
- **Fase de Handshake o Chequeo:** un paquete de información es enviado, indicando el éxito o fracaso de la transacción de información realizada]

El estándar USB como tal es un protocolo tipo polling o encuesta, en donde el *host* o anfitrión consulta a cada uno de los dispositivos en forma periódica si requiere hacer una transacción de datos. Es únicamente en este momento, cuando el dispositivo esclavo es encuestado, en que puede enviar los datos al computador, de otra forma debe esperar a que sea nuevamente encuestado. En caso de que el dispositivo esclavo no requiera intercambiar ninguna información, el *host* lo pasa por alto y continua encuestando los demás dispositivos conectados al bus. [5]

Con respecto a las transacciones o transferencias de información que el *host* ejecuta con los dispositivos esclavos, estas se catalogan de la siguiente manera:

- **Transferencia de Control:** En esta transferencia se envían datos de control que sirven para configurar el periférico en el momento en que se conecta al computador .
- **Transferencia Bulk o al por mayor:** empleada para la transmisión de importantes cantidades de información de carácter no urgente, cuya transmisión puede demorarse, como información hacia impresoras.

- **Transferencia tipo interrupción:** transferencia empleada para enviar y recibir pequeños paquetes de información en forma rápida, como los son los movimientos del ratón.
- **Transferencia Isocrónica:** transferencia de datos empleada para transmitir datos de audio y vídeo. Este tipo de transferencia funciona en tiempo real y tiene la mayor prioridad.

### 3.2.2.2 Protocolo SPI

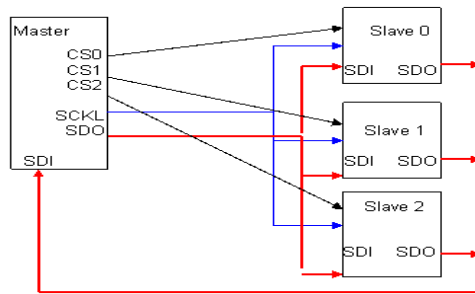
El protocolo SPI (Serial Peripheral Interface) desarrollado por Motorola está basado en la comunicación serial sincrónica entre un dispositivo *host*, generalmente un procesador y un grupo de periféricos.

Dentro del protocolo SPI existen dos tipos de dispositivos: maestros y esclavos. El dispositivo maestro es aquel que provee la señal de reloj a todos los dispositivos tipo esclavo, y además selecciona el dispositivo con el cual requiere comunicarse.

El protocolo SPI requiere 2 líneas de control llamadas CS y SCLK y dos líneas para transmisión de datos llamadas SDI y SDO. La línea de control CS corresponde al *Chip Select* o habilitación del chip, mientras que la línea SCLK corresponde a la señal de reloj o señal de sincronía que genera el dispositivo maestro.

Por otra parte, a través de la línea SDI, los datos son ingresados al dispositivo en cuestión, mientras que la línea SDO permite la salida de los mismos[6]

El protocolo SPI posee dos configuraciones básicas: MOSI y MISO. La primera configuración, MISO corresponde a la configuración por su nombre en inglés *Master In, Slave Out* o maestro receptor, esclavo emisor, mientras que MOSI corresponde a *Master Out, Slave In* o maestro emisor, esclavo receptor..



**Figura 3.1: Esquema de organización de dispositivos bajo el protocolo SPI**

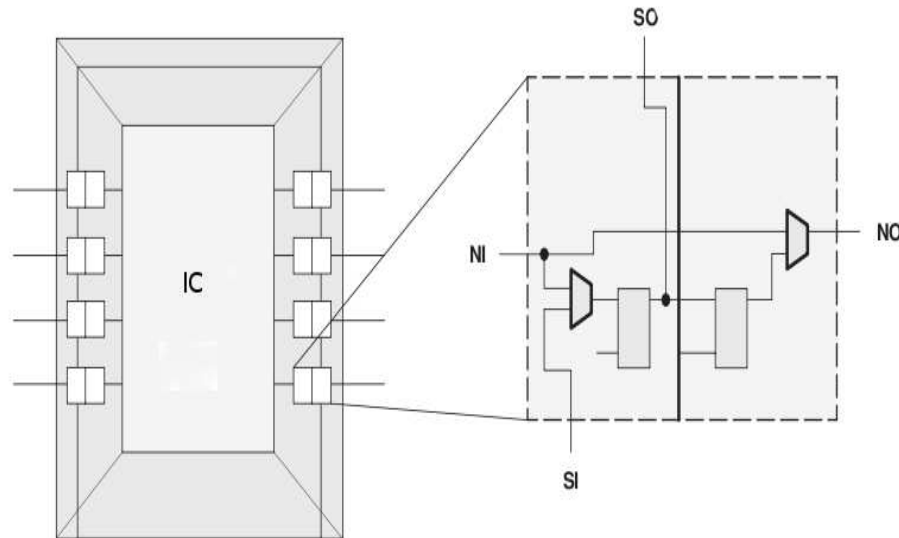
Para ilustrar el funcionamiento del protocolo SPI, se puede observar la configuración en la que se encuentran colocados los dispositivos de la figura 3.1[7]

El elemento maestro es capaz de controlar y entablar la comunicación con los tres dispositivos esclavos de la derecha mediante las terminales SDI (Serial Data Input) y SDO (Serial Data Output). Nótese que la señal de sincronía o reloj es recibida al mismo tiempo por los dispositivos esclavos, sin embargo la habilitación de los mismos es dada de forma individual mediante el control de la línea CSX de cada esclavo por parte del dispositivo maestro.

### 3.2.2.3. JTAG

El protocolo JTAG (Joint Test Action Group) es un protocolo desarrollado como solución al problema que tenían los fabricantes de hardware, de no poder realizar pruebas a sus diseños, debido a la dificultad de montaje y acceso a los pines de los circuitos integrados empleados.

El protocolo JTAG, también llamado Boundary Scan implica la colocación de puntos de prueba, llamados *scan cells* en cada una de las entradas y salidas de un circuito integrado, bajo un mismo encapsulado. Mediante el empleo de los puntos de prueba se pueda conocer el flujo de datos a través del pin, o bien controlar el estado de un pin.



**Figura 3.2: Detalle de un punto de prueba en un circuito integrado**

La figura 3.2 ilustra propiamente como está constituido un punto de prueba en un circuito integrado compatible con el protocolo JTAG.

El punto de prueba cuenta con una entrada y una salida de datos (NI y NO respectivamente) que se encuentra conectada a la parte del encapsulado donde se ubica propiamente la lógica que constituye el circuito integrado.

Cuando el protocolo JTAG no es empleado, la conexión entre los pines NI y NO es directa, por lo que se puede interpretar como que existe un pequeño cable que une ambos extremos. Cuando el protocolo es empleado, la conexión entre ambos puntos es interferida para que se pueda observar el desplazamiento de datos.

Por otra parte el punto SI permite ingresar datos para que sean monitoreados, o simplemente para fijar pines en un estado requeridos, mientras que el punto SO permite monitorear un dato conforme se desplaza a través de un pin [8].

Dada la orientación que tiene el protocolo, también es empleado para la prueba de submódulos de circuitos integrados y depuración de aplicaciones empotradas, dado que le permite al desarrollador tener una entrada alternativa hacia dentro del sistema, razón por lo cual el empleo de este protocolo es muy popular.

El protocolo Jtag está basado en 5 líneas o señales, pero se considera un protocolo tipo serie dado que únicamente cuenta con una sola línea para entrada de datos:

- T.D.I o datos de entrada de prueba
- T.D.O o datos de salida de prueba
- TCK: reloj de prueba
- TMS: Selector de modo
- TRST: Reset de modo de prueba

### **3.2.3 Medición de corriente**

Existen muchos métodos para medir corriente, sin embargo la tendencia hoy día se inclina a el empleo de sensores basados en el efecto Hall por la facilidad con la que permiten medir corriente.

El sensor ACS712 de Allegro Microsystems tiene características que lo hacen especial y sobresalir entre muchas opciones existentes en el mercado. Al ser un sensor de efecto Hall, la magnitud de la corriente sensada puede ser mayor a los valores máximos de medición que ofrecen otros fabricantes. Su baja resistencia al paso de la corriente, así como su tensión proporcional a la magnitud de la corriente que pasa a través de él y claro está, su precio, lo hacen ser el sensor más apto para este tipo de proyecto[9].

### **3.3 Descripción de los principales principios de software, electrónicos y físicos relacionados con la solución del problema.**

Dado que el presente proyecto tiene como fin eliminar varias necesidades de diversa índole, las cuales constituyen un problema general, se emplean varios conceptos tanto físicos como electrónicos y computacionales, los cuales no presentan una relación directa entre si, sin embargo el encontrar una forma de relacionarlos conlleva a una ayuda significativa en la obtención de la solución global.

### **3.3.1 Principios de software relacionados**

El empleo de un microcontrolador en todo proyecto, tiene asociado un desarrollo de software, de forma que el microcontrolador funcione tal y como el programador lo desea.

El desarrollo de software se realiza en el computador, pero no es sino en el microcontrolador donde se ejecuta con el fin de que este funcione como se quiere y responda al entorno de variables en el cual ha sido inmerso.

La programación del microcontrolador puede llevarse a cabo mediante un lenguaje de alto o bajo nivel, o bien la combinación de ambos.

El programa generado, está estructurado de forma que contenga las instrucciones necesarias para ejecutar cada una de las acciones que se han pensado realizar.

El desarrollo pleno del software del microcontrolador, está basado en el lenguaje C, mientras que algunas instrucciones, como las de inicialización de la pila de datos, registros de uso general y específico son programadas en ensamblador para mayor facilidad a la hora de inicializar todo el software.

El empleo del lenguaje C, permite al desarrollador utilizar técnicas de programación, tales como arreglos y procedimientos, que permiten al programa ser más eficiente.

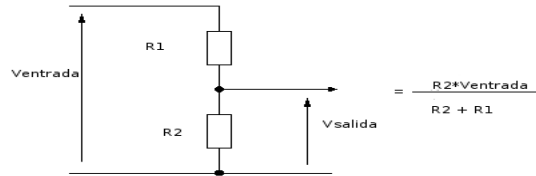
Mediante el empleo de procedimientos, el software se estructura en pequeños módulos con funciones específicas, lo que hace que las muchas de las diversas instrucciones que componen el programa sean ejecutadas únicamente cuando son requeridas.

### **3.3.2 Principios electrónicos relacionados**

#### **3.3.2.1 Divisor de tensión**

El divisor de tensión, es una técnica empleada basada en la ley de Ohm. Consiste en configurar un circuito eléctrico, de forma que la tensión se reparta entre una o mas impedancias conectadas en serie y de esta forma lograr un control de la

tensión repartida. Los divisores de tensión son muy empleados en etapas de circuitos donde se quiere tener como salida, una tensión equivalente a un porcentaje dado de la tensión de entrada



**Figura 3.3 Esquema de un divisor de tensión**

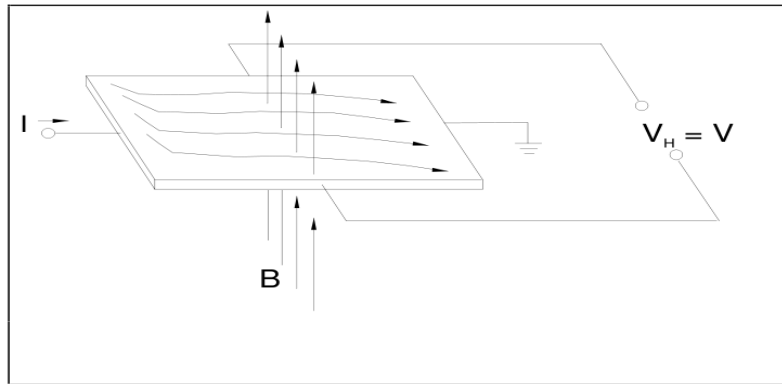
Como se puede observar en la figura la tensión de salida está dada por la relación de resistencias R1 y R2, y para que ésta relación se cumpla, la tensión de salida debe ser tomada en el resistor R2.

### **3.3.3 Principios físicos relacionados**

#### **3.3.3.1 Efecto Hall**

El efecto Hall consiste en que cuando una lámina delgada de algún material semiconductor conduce una corriente eléctrica y se halla situada en un campo magnético perpendicular a la dirección de la corriente, se desarrolla sobre la placa un campo eléctrico transversal. Este campo eléctrico es perpendicular al sentido de la corriente y es denominado campo de Hall. Este campo resulta ser el resultado de las fuerzas ejercidas por el campo magnético sobre las partículas de la corriente eléctrica, sean estas positivas o negativas, o positivas en un sentido y negativas en el otro. Una de las consecuencias principales del efecto, descubierto por Edwin Duntey Hall, es la acumulación

de carga de forma que en el otro lado de la placa se tenga una carga opuesta y por ende se produzca una diferencia de potencial.[12]



**Figura 3.4 Principio del efecto Hall**

La figura 3.4 muestra el principio del efecto Hall, mediante la interacción de un campo magnético **B** perpendicular a una lámina delgada de un semiconductor, que a su vez conduce una corriente **I**, dando como resultado una diferencia de potencial **V** en los extremos de la placa. La diferencia de potencial es directamente proporcional al producto cruz de la corriente **I** por el campo magnético.

La ecuación 3.1 ejemplifica lo anterior:

$$V \propto I \times B$$

**EC (3.1)**



## **CAPITULO 4: Procedimiento Metodológico**

En el presente capítulo, el lector encontrará una serie de procedimientos, así como las decisiones que motivaron a ejecutar estos procedimientos, con el objetivo de alcanzar la culminación del proyecto.

### **4.1 Reconocimiento y definición del problema**

El reconocimiento del problema en el presente proyecto, fue un proceso que se dio de forma lenta y completamente paralela al crecimiento y desarrollo de la empresa, en donde estuvo involucrado la mayor parte del personal.

De manera conjunta, el personal llegó a la conclusión que en la mayoría de los casos en los que algún desarrollador sufría un retraso en la culminación de su trabajo, la falta de equipo, ya sea por encontrarse dañado, o bien por ser insuficiente se perfilaba como la principal causa de esta situación.

A pesar de que el personal de forma conjunta realizó un correcto reconocimiento del problema, no se preocupó por definirlo.

El éxito de un proyecto está completamente relacionado con la identificación de las causas que motivan a realizar ese proyecto, por lo que el saber qué es lo que se requiere, es vital.

Para definir un problema, sin importar si ya se conoce o no, se debe hacer la siguiente pregunta: ¿Qué es lo que se desea?. Para el caso específico de RidgeRun, no basta con sólo conocer que la carencia de herramientas está relacionada con atrasos en la entrega de trabajos. Al realizarse la pregunta ¿Qué es lo que se desea?, se puede llegar a la rápida conclusión de que lo que se desea no es la adquisición de muchas unidades de un pequeño grupo de herramientas de desarrollo, sino que lo que se desea es buscar la forma de que la empresa pueda solucionar el problema del faltante de herramientas tanto a corto, como a mediano plazo de una forma rentable y fácil.

## 4.2 Obtención y análisis de la información

La calidad de la información que se obtenga, y el análisis que se le de a la misma, son los pilares fundamentales, sobre los cuales se desarrollará la solución del problema. En este caso, la mejor forma para obtener información relevante, es mediante la realización de una pequeña entrevista o conversación con cada uno de los desarrolladores de la empresa, quienes son las personas que requieren directamente la solución del problema.

El tema central de la conversación o entrevista se basó en las preguntas:

- ¿Describa una herramienta que le sea multifuncional en su trabajo?,
- ¿Qué propiedades debería tener para que a usted le sea funcional?
- ¿Cuáles son sus principales necesidades?

Una vez concluida la etapa de entrevistas, se pudo constatar mediante las respuestas de cada uno de los desarrolladores, que la mayoría coincidió en que para su tipo de trabajo, una herramienta multifuncional sería aquella que fuese compatible con los distintos conectores y protocolos existentes en el mercado. Además, todos los desarrolladores concluyeron, que uno de los puntos a los cuales le daban mucha importancia, era a la comodidad para trabajar que pudiese ofrecer una herramienta, en otras palabras, que fuese lo más compacta posible.

Ahora bien, a pesar de que la herramienta de la entrevista es sumamente útil, esta no evita que muchos factores o criterios queden por fuera, por lo que se debe contar con una segunda fuente de información. En este caso, ésta segunda fuente de información es Internet. Una correcta forma de iniciar la investigación en Internet, tomando en cuenta los intereses y objetivos de este proyecto, es realizando una serie de preguntas preliminares que harán la función de filtro, con el fin de evitar que información no funcional sea tomada

en cuenta. En este punto, las preguntas que componen dicho filtro deben ser sencillas, pero directas y guardar una estrecha relación con el objetivo al cual se desea llegar. Algunas de las preguntas que permitieron filtrar la información de una mejor forma fueron:

- ¿Qué herramientas se usan en el campo de los sistemas empotrados?
- ¿A qué procedimientos están ligadas las herramientas existentes para dispositivos empotrados
- ¿Cuál es el precio de esas herramientas, y quienes son los fabricantes?
- ¿Existen módulos que contemplan varias herramientas en un sólo bloque?
- ¿Se puede encontrar información de forma fácil para modificar y reparar las herramientas para desarrollo de sistemas empotrados?
- ¿Tienen restricciones estas herramientas a la hora de ser usadas?
- ¿Existe alguna licencia para éste tipo de herramientas?
- ¿Compatibilidad con software? ¿Para qué sistema operativo la herramienta es funcional?
- ¿Es posible obtener el diseño completo del dispositivo empleado?

Seguidamente, una vez que se tiene establecido un sistema sencillo de filtrado de información, se puede emplear una técnica de investigación, que tiene una alta eficacia asociada: la búsqueda modular o en escalera. Partiendo de un dispositivo comercial seleccionado, se trata de saltar a un nivel inferior siempre y cuando se pueda, con el fin de conocer cómo está integrado internamente. Una vez que se conoce esto, se puede bajar un nivel en la etapa de

modularización para consultar las hojas de datos de cada dispositivo, e investigar con profundidad no sólo su funcionamiento, sino también algunos diseños que por lo general los fabricantes sugieren en sus hojas de datos. En este punto de la investigación, se debe hacer un sencillo análisis de costos, con la intención de verificar si la información encontrada justifica la continuación de la investigación y garantiza que de un modo u otro y en forma objetiva, se podrá resolver el problema sin realizar una inversión comparable con la de adquirir herramientas totalmente nuevas.

#### **4.3 Evaluación de las alternativas y síntesis de una solución.**

La evaluación de cada propuesta o alternativa de solución que se plantea, es una actividad vital relacionada directamente con el éxito del proyecto. Es por ello que cada alternativa que surja, debe ser analizada desde múltiples puntos de vista, entre ellos los más importantes: económico, ingenieril y productivo, con el fin de mejorar la propuesta a un punto máximo, logrando de esta forma convertirla en una síntesis o idea principal sobre la cual girará la solución.

RidgeRun, mediante el asesor designado para el proyecto, presenta como punto de partida para una eventual alternativa de solución, una tarjeta de depuración, llamada “*debug card*” utilizada específicamente en uno de los dispositivos que se encuentra en desarrollo. Esta tarjeta de depuración, no es más que un grupo de conectores, tanto estándares (DB9, mini-USB y RCA), como no estándares (conector únicamente utilizado en el dispositivo en desarrollo) interconectados e integrados en una tarjeta de circuito impreso. Esta sugerencia, unida con los resultados de las entrevistas realizadas, dan a conocer, que la integración de diferentes funciones en una misma herramienta se perfila como la mejor alternativa para solucionar el problema planteado



**Figura 4.1 Tarjeta de desarrollo empleada en RidgeRun**

La figura 4.1 muestra la tarjeta de desarrollo empleada en RidgeRun que se mencionó en el párrafo anterior, y que a partir de esta se estableció la síntesis de solución.

#### **4.4 Implementación de la solución**

Una implementación exitosa no es sinónimo de una solución, exitosa, pero para que esto suceda es necesario cumplir una serie de pasos durante este proceso que a continuación se mencionan:

1. Conocer a profundidad las bases del proyecto, en otras palabras: conocer por qué se realiza el proyecto, qué lo inició y tener claro a qué se quiere llegar.
2. Tener claras las limitantes tanto del proyecto como del entorno, para así evitar a toda costa forzar los límites de este.
3. En el caso específico de RidgeRun, toda implementación previa a su realización, debe ser confrontada contra un análisis de costos, con el fin de determinar la utilidad de la misma, y por ende debe contar con el respaldo de la empresa.

Ahora bien, previo a la implementación final de cualquier solución planteada, se debe realizar una implementación a manera de prototipo, con el cual se puedan evaluar los resultados obtenidos, y con base en estos considerar si un rediseño es requerido para lograr los resultados deseados.

Por otra parte, una implementación final nunca está completa, sin una referencia para quien vaya a utilizarla. Es por esto que un manual técnico de usuario, o bien una guía de uso debe ser provista. Quien vaya a utilizar dicha implementación, debe tener una guía o fuente de referencia inmediata sobre los procedimientos a seguir para lograr los resultados deseados.

#### **4.5 Reevaluación y rediseño**

Todo proyecto electrónico no sólo debe ser funcional y cumplir los estándares para los cuales fue concebido. Este debe contar con la posibilidad de rediseñarse a partir de un proceso de evaluación. Este proceso de evaluación debe ser crítico y meramente objetivo. La empresa lo puede ejecutar, ya sea cuando lo considere necesario, o bien en un lapso de tiempo posterior a la culminación del proyecto (generalmente el tiempo prudencial es de 6 meses a 2 años), con el fin de prolongar la vida útil del diseño realizado y mantenerlo vigente.

Una forma de simplificar el proceso de reevaluación y rediseño es mediante la previsión de la necesidad de actualización del sistema, dándole al sistema facilidad para acoplar bajo un protocolo conocido y variables previamente establecidas nuevos módulos aún no determinados.

## **CAPITULO 5. Descripción detallada de la solución**

### **5.1 Análisis de soluciones y selección final**

Como se ha podido conocer durante el transcurso del presente proyecto, la solución global está orientada a la agrupación de varios módulos, cada uno con una función específica dentro de un sistema general

A continuación se muestran las soluciones planteadas a la hora de realizar el diseño final, y los criterios que se tomaron en cuenta para aprobar o rechazar cada solución.

#### **5.1.1 Soluciones propuestas**

##### **5.1.1.1 Propuestas para los módulos presentes en el sistema por desarrollar**

Como primera propuesta de solución del sistema por desarrollar, se planteó diseñar e implementar un dispositivo electrónico, capaz de conectarse por puerto serie al computador, y que comprendiera un analizador lógico, un multímetro, una fuente CD de alimentación para uso general (amplio rango de tensión de salida), interfaz de: Jtag, SPI, Ethernet, para controlar el dispositivo construido remotamente. Para el diseño e implementación de esta propuesta, se requiere la adquisición de componentes electrónicos con un alto nivel de integración, y de difícil obtención. El diseño del analizador lógico, está basado en un diseño libre, y que utiliza como base un FPGA Spartan 3 de Xilinx.

Como segunda propuesta se planteó diseñar e implementar un sistema electrónico, que comprendiera un módulo de interfaz Jtag, un módulo de comunicación serial, un voltímetro, un amperímetro y un módulo de depuración de propósito general (compuesto por un banco de pines que funcionan como entradas y salidas ). El módulo propuesto tendría conexión al computador a través del puerto USB. Además se le agregaría un módulo que funcionaría como fuente de alimentación CD (con tensiones de salida predefinidos por la empresa), para alimentar los dispositivos que se desarrollen y además para

darle más valor agregado al dispositivo por diseñar. Se propone adicionalmente, dejar implementados para cuando el caso lo amerite, la conexión de módulos a través de las interfaces SPI, serie y paralelo. Para éstas interfaces disponibles, se deja listado un grupo de dispositivos, que en el futuro podrían ser conectados al sistema diseñado. Entre estos módulos se pueden mencionar : módulos SPI para conexión Ethernet, memorias EEPROM , etc.

### 5.1.1.2 Escogencia de las propuestas para los módulos que integrarán el sistema por desarrollar.

Para lograr tener un mejor criterio de las propuestas de diseño en cuanto a los módulos que integrarán el sistema, es necesario realizar una comparación de las ventajas y desventajas que puedan presentar al proyecto durante su implementación y por ende funcionamiento en general.

**Tabla 5.1: Propuestas para los módulos que integrarán el sistema por desarrollar**

	<i>Ventajas</i>	<i>Desventaja</i>
Propuesta #1	Disponibilidad de un mayor número de herramientas	Elevado costo del sistema
	Sistema general de desarrollo multidispositivo, también funcional para artefactos no empotrados	Nivel de desarrollo requerido es muy elevado y complejo para lograr un funcionamiento similar al comercial
		Tiempo prolongado de desarrollo
Propuesta #2	Costo de desarrollo inferior	Menor cantidad de herramientas disponibles
	Tiempo de desarrollo no tan prolongado	Sistema sólo para dispositivos empotrados



Para el diseño de la estructura se seleccionó la propuesta #2, tomando en cuenta como principal factor el costo del sistema. En cuanto a la preferencia de la empresa, se debe mencionar que también se inclinó por esta opción, y un factor que hizo se inclinase por la misma, es que la propuesta del proyecto queda abierta para futuras mejoras, sin necesidad de hacer una fuerte inversión en dinero y recursos. La empresa considera, que la opción #1 podría ser implementada y sería útil, sin embargo no es apropiada para efecto de un proyecto de graduación, ya que el simple hecho de desarrollar un analizador lógico con una resolución muy elevada como la que se requiere para el desarrollo de sistemas empotrados, implicaría una inversión elevada de tiempo y esfuerzo por parte del desarrollador, y el resultado esperado, sería muy difícil de alcanzar en un tiempo corto.

### **5.1.2 Propuestas de funcionamiento para algunos de los módulos incluidos.**

#### **5.1.2.1 Circuito de Fuente de alimentación CD**

La primera propuesta que se tiene para este circuito, es diseñar e implementar una fuente de alimentación con tensión de salida variable en un rango de 0V a 33V, y corriente de salida máxima de 2A controlada por el microcontrolador, a través de el computador. Con un diseño de éstas características, no sólo los sistemas empotrados se podría alimentar, sino que la opción de uso sería más generalizada, y casi que cualquier dispositivo CD podría ser utilizado mientras no superase las especificaciones máximas de la fuente. Ésta propuesta requiere de componentes electrónicos tanto activos como pasivos, de fácil adquisición, sin embargo el tamaño y el peso del dispositivo final aumentaría considerablemente debido al transformador que se requiere utilizar.

La segunda propuesta, es diseñar una fuente de alimentación CD, enfocada principalmente a alimentar sistemas empotrados, por lo que los valores de tensión serían definidos por la empresa (analizando las tensiones de alimentación de los dispositivos con los que normalmente se trabajan) y el valor máximo de salida no superaría los 10V. Se propone emplear en sustitución de un transformador y un circuito de rectificación, un adaptador AC de uso comercial, logrando además la reducción del tamaño del circuito final. En cuanto a la selección de la tensión de salida, se plantea que sea el usuario el que realice dicha selección por medio de algún switch, de forma que no dependa de la obligación de tener el computador encendido para realizar la selección de la tensión de salida.

#### **5.1.2.2 Propuestas para el circuito de fuente de alimentación CD.**

De acuerdo con el siguiente cuadro comparativo, se pueden observar las diferencias más importantes entre ambas propuestas.

**Tabla 5.2: Cuadro comparativo de propuestas para los módulos que integrarán el sistema por desarrollar.**

	<i>Ventajas</i>	<i>Desventaja</i>
Propuesta #1	Amplio rango de funcionamiento	Su capacidad de amplio rango de funcionamiento no será aprovechada al máximo
	Forma de selección de tensión de salida rápida y programable	Tamaño: el uso de un transformador aumentará el tamaño del dispositivo final Requerimiento/dependencia de tener el computador encendido para variar las tensiones de salida
Propuesta #2	Tamaño reducido al no tener circuito rectificador	Rango de funcionamiento limitado sólo para sistemas empotrados
	Menor costo	Tensiones de salida modificables sólo en forma manual
	No requiere tener el computador encendido para seleccionar o modificar la tensión de salida	

Después de analizar las propuestas, se selecciona la propuesta #2. Entre los factores que motivaron esta decisión, están la no dependencia del computador para modificar las tensiones de salida, y la reducción de costos, al existir la posibilidad de utilizar un adaptador AC de uso general como etapa rectificadora

### **5.1.2.3 Circuito de interfaz de transmisión de datos/acople con el computador**

La primera propuesta para éste módulo, se orientó a acoplar el sistema diseñado, con el computador, a través del puerto serie, de forma que la comunicación entre ambos fuese lo más sencilla posible, sin embargo en el futuro se podrían tener limitantes a la hora de mejorar el sistema, relacionados con la velocidad de transmisión de datos entre el computador y el sistema en general.

La segunda propuesta se orientó a modificar el circuito de acople, por uno basado en el protocolo USB, ya que se evitarían problemas relacionados con velocidades de transferencia de datos, y además la forma de conectar el dispositivo sería mucho más sencilla.

La segunda propuesta fue la seleccionada, dada la garantía que ofrece al permitir conectar el sistema con el computador, aún cuando el computador no cuente con un puerto serie

### **5.1.2.4 Interfaz sobre la cual se basa la posibilidad de expansión del sistema**

En vista de la necesidad de darle al proyecto una opción de mejora en el futuro, se propuso al terminar el proyecto, dotar al dispositivo con las conexiones necesarias requeridas por algún protocolo que permitiesen agregar algún módulo en forma sencilla

Como primera opción se propuso la interfaz SPI, es decir, basar la posibilidad de expansión o mejora del sistema en ésta interfaz. El protocolo SPI permite acoplar módulos al sistema principal con gran facilidad y rapidez. Esta propuesta se basa en el resultado de una búsqueda en las páginas de diversos fabricantes, en donde ofrecen herramientas para sistemas empotrados, basadas en este protocolo.

Como segunda opción se propuso basar la posibilidad de mejora del sistema en una interfaz paralela. Esto se hizo dado la forma fácil de conexión y depuración de errores que ofrece ésta interfaz, y además se tomó en cuenta lo sencillo que resulta diseñar nuevos elementos bajo esta forma de transmisión de datos.

Para tomar la decisión se analizó el siguiente cuadro comparativo

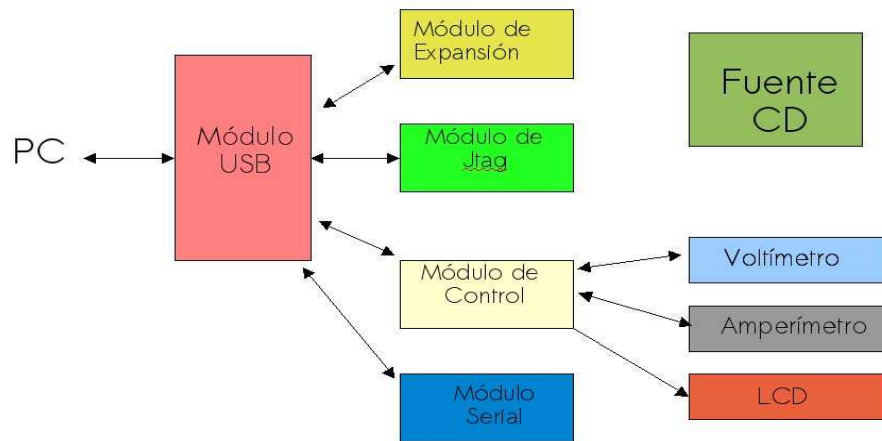
**Tabla 5.3: Propuestas para interfaz de comunicación sobre la cual se basa la posibilidad de expansión en el futuro.**

	<i>Ventajas</i>	<i>Desventaja</i>
Propuesta #1	Fácil implementación	Requiere tomar en cuenta temporizaciones
	Muchos dispositivos basados en esta interfaz	
Propuesta #2	Fácil implementación	Mayor cantidad de líneas requeridas
	Relativa facilidad a la hora de diseñar	Algunas computadores carecen de puerto serial

La propuesta seleccionada fue la #1, y la justificación de su selección se basó en la gran cantidad de dispositivos que se encuentran en el mercado compatibles con este protocolo. Al contarse con una gran cantidad de opciones disponibles para agregar al proyecto, el valor agregado del mismo crece significativamente.

## 5.2 Descripción del Hardware

El hardware descrito en la siguiente sección está basado en el siguiente esquema:



**Figura 5.1 Diagrama de bloques del sistema desarrollado**

El sistema está constituido por un módulo USB, el cual es el encargado de realizar la interfaz entre el computador y el resto de los demás módulos presentes en el sistema.

El módulo de expansión permite al sistema acoplarse con nuevos módulos en caso de ser necesario.

El módulo de Jtag permite realizar operaciones donde el protocolo de Jtag sea necesario. Algunas de las tareas asociadas con este módulo son las que se encuentran ligadas a la programación, edición y borrado de memorias de algunos de los dispositivos que se desarrollan en la empresa.

El módulo de control tiene como función, dirigir y controlar las acciones de los módulos: de LCD, amperímetro y voltímetro.

El módulo LCD permite al usuario leer datos tanto del amperímetro como del voltímetro, mientras que estos dos últimos módulos, como su nombre lo indica, tienen la función de medir corriente y tensión respectivamente.

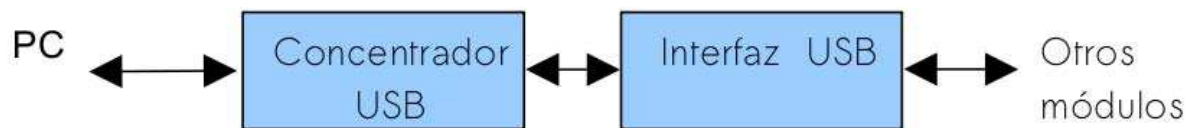
Por último el módulo serial permite establecer comunicaciones con dispositivos del mismo tipo .

Antes de iniciar la descripción del hardware, se debe mencionar que los módulos de Expansión, Jtag, Control y Serial comparten un circuito integrado propio del módulo USB, que para efectos de una mejor descripción de la constitución y diseño del sistema completo, cada módulo en las siguientes secciones se tratará de forma independiente, claro está, mencionando los vínculos existentes con respecto a los otros módulos.

Seguidamente se detallará cada uno de los módulos que integran el diagrama de la figura 5.1.

### 5.2.1 Módulo USB

Este módulo está constituido por un circuito llamado concentrador USB y un circuito de interfaz USB con el computador, tal y como se muestra en la figura 5.2



**Figura 5.2 Esquema interno del módulo USB**

#### 5.2.1.1 Circuito Concentrador (*Hub*) USB

El concentrador USB es un circuito cuya función es agregar varios puertos USB a partir de un único puerto del mismo tipo. Su empleo dentro del presente proyecto se justifica mediante la necesidad de que el usuario tenga otro grupo de puertos USB diferentes a los que ya existen en su computador, donde poder conectar elementos USB.

El circuito del concentrador USB que se observa en la figura 5.3 esta basado

en el circuito integrado CY7C65630A de *Cypress Semiconductor Corporation*.

Para dicho circuito, no fue necesario realizar diseño alguno, ya que se empleó el diseño sugerido por el fabricante en la hoja de datos.

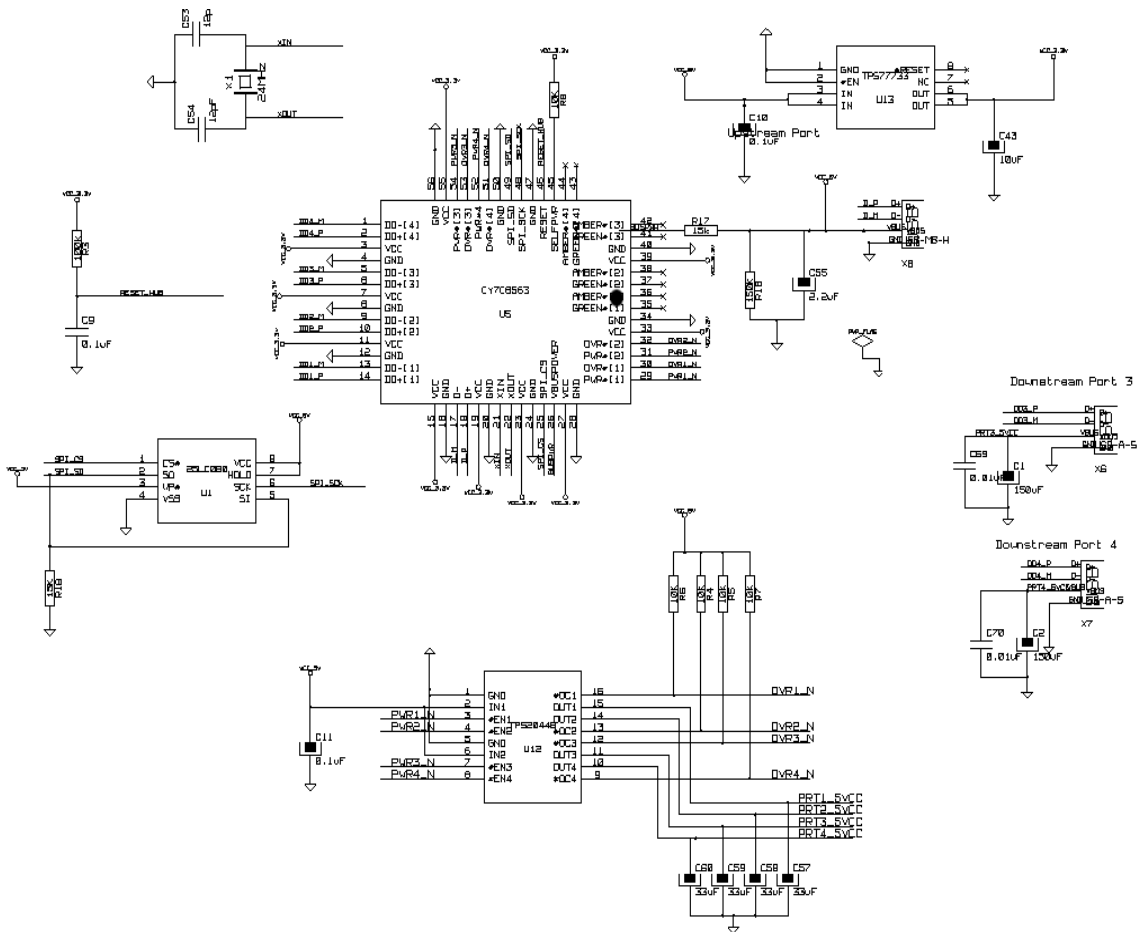
Sin embargo se consideró realizar una variante en el circuito empleado en el presente proyecto. Esta variante fue la eliminación de algunos componentes como resistencias y leds, empleados para mostrar el estado del circuito. La eliminación de estos componentes se justifican bajo la necesidad de reducir el consumo de corriente, dado que el circuito en general será alimentado por la corriente del puerto USB de la computadora. Con la eliminación de estos componentes, se logra, para el peor de los casos, cuando todos los leds se encuentran encendidos al mismo tiempo ,un ahorro de 19.4mA.

Con respecto al chip de manejo de corriente, se utilizó el circuito integrado TPS2044 de Texas Instruments, y las razones de peso para su selección fueron: su capacidad para entregar corriente continua (500 mA) y su baja resistencia interna (80mΩ).

Como el circuito requiere una tensión de 3.3V para funcionar, se agregó un regulador TPS77733 de Texas Instruments cuyo voltaje de salida es de 3.3V, logrando de esta manera polarizar el circuito.

La selección del circuito TPS77733 para este módulo y el resto de módulos donde se emplea el mismo circuito integrado, se justifica en que es un regulador LDO (*Low Drop Out*) que es la característica eléctrica de poder operar con un voltaje diferencial entrada-salida sumamente bajo.





**Figura 5.3 Circuito correspondiente al concentrador USB**

**Funcionamiento:**

El circuito integrado obtiene los 3.3V que requiere para funcionar a partir de un regulador TPS77733 de Texas Instruments, el cual a su vez se alimenta de la tensión VCC (5V) del puerto USB.

Cada vez que el circuito es energizado, éste obtiene de una memoria EEPROM de interfaz SPI, los datos correspondientes a la forma en que identificará el dispositivo. VID(identificación del vendedor),PID(identificación del producto), DID (identificación del producto. Dado que el fabricante provee el chip junto con el firmware, este no requiere ningún comando o software de configuración o control.

El suministro de la tensión de VCC a cada puerto es realizado mediante el chip de distribución de energía TPS2044B. En caso de darse un cortocircuito o una sobrecarga en el sistema, el circuito CY7C65630A es capaz de detectar la situación y enviar una señal al chip que distribuye la energía, para que el puerto con exceso de demanda de corriente sea apagado inmediatamente.

En la figura 5.3 se muestran las conexiones del módulo USB. El puerto *Upstream* es el puerto que por donde se realiza la conexión entre el computador y el concentrador USB, mientras que los puertos *Downstream* son aquellos donde se conectan los periféricos que se desean conectar con el computador por medio del concentrador.

La mayoría de los concentradores o Hubs USB poseen 4 puertos de tipo *Downstream*, sin embargo en la figura 5.3 únicamente se observan 2. La razón de esto se debe a que los puertos 1 y 2 fueron conectados directamente con los demás dispositivos presentes en el sistema desarrollado.

Los puertos 3 y 4 que se observan, se encuentran disponibles para que en caso de que el usuario requiera conectar dispositivos USB, lo pueda hacer a través de estos puertos.

#### **5.2.1.2 Circuito de interfaz USB con el computador**

Este circuito es uno de los más importantes del sistema, dado que es a través de éste que se realiza la comunicación de datos entre el computador y los demás módulos del sistema mediante la conversión de datos del protocolo

USB a diferentes protocolos o modos de funcionamiento según la necesidad del usuario.

Este módulo está basado principalmente en dos circuitos integrados FT2232D de *Future Technology Devices*, los cuales tienen funciones independientes asignadas que posteriormente serán discutidas. El firmware provisto por el fabricante para el circuito FT2232D, es capaz de realizar el manejo completo de el protocolo USB e interfazar los datos con alguno de los modos que pueden ser seleccionados por este mismo circuito integrado. Esta ventaja hace posible que no sea necesario diseñar y emplear un software que tenga la función de realizar el manejo del protocolo USB. La selección del módulo que se desea implementar en cada uno de los puertos del FT2232D, se realiza a través de un software creado por el fabricante para tal fin. Los posibles modos a implementar son:

- Modo Serie:
- Modo Paralelo-FIFO
- Modo Multiprotocolo.

El modo Paralelo-FIFO es una versión de funcionamiento del protocolo paralelo de comunicación, el cual cuenta con 8 líneas para datos y 4 líneas para control

El modo Multiprotocolo permite la implementación de los protocolos sincrónicos seriales tales como JTAG y SPI, sin embargo el modo Multiprotocolo es únicamente funcional según disposición del fabricante en el puerto A, mientras que tanto el modo serie como el modo paralelo pueden ser implementados en ambos puertos (A y B) sin diferencia alguna.

La descripción sobre la programación de estos modos será tratada en la sección de software del presente capítulo.

De igual forma que para el circuito del apartado anterior, el circuito de la figura 5.4 que muestra la conexión básica para el funcionamiento del circuito FT2232D, no requirió de diseño alguno puesto que se empleó el diseño sugerido por el fabricante indicado en la hoja de datos.

En el circuito propuesto por el fabricante, y de igual forma en el circuito de la figura 5.4, se puede observar que la conexión básica del FT2232D incluye un transistor FET de canal P. La función de este transistor es mediante la acción de el pin #PWREN del FT2232D que controla el *gate* del transistor, mantener el consumo de corriente por debajo de los 100mA durante la de inicialización del dispositivo USB, para aquellos circuitos en donde hay conmutación de etapas de potencia.

El FT2232D puede funcionar, tanto con lógica TTL como con lógica LVTTTL. La selección de la lógica de trabajo se lleva a cabo mediante la conexión de los pines 14 y 31 (VCCIOA y VCCIOB respectivamente) a 3.3V si se desea emplear lógica LVTTTL, o bien a 5V si lo que se requiere es emplear lógica TTL.

### **Funcionamiento:**

Cuando el circuito es energizado mediante su acople con un puerto USB del computador, el circuito FT2232D lee el contenido de la memoria EEPROM mediante la interfaz de comunicación SPI, para obtener los parámetros que identifican al elemento USB: VID, PID y DID. Además de estos datos, el FT2232D lee la información referente al protocolo que se deberá implementar en cada uno de los puertos. Una vez leída la información, el circuito FT2232D queda listo para funcionar.

La comunicación de este módulo con el concentrador USB o bien con el

computador mediante el protocolo USB se da a través de los pines DD1\_M y DD1\_P quienes son los encargados de recibir y enviar datos.

La alimentación de éste módulo se da a través del pin PRT1\_5VCC, el cual y como se mencionó anteriormente, en caso de presentarse una sobrecarga o un cortocircuito, es apagado por acción del circuito integrado TP2044B integrado en el módulo de concentrador USB.

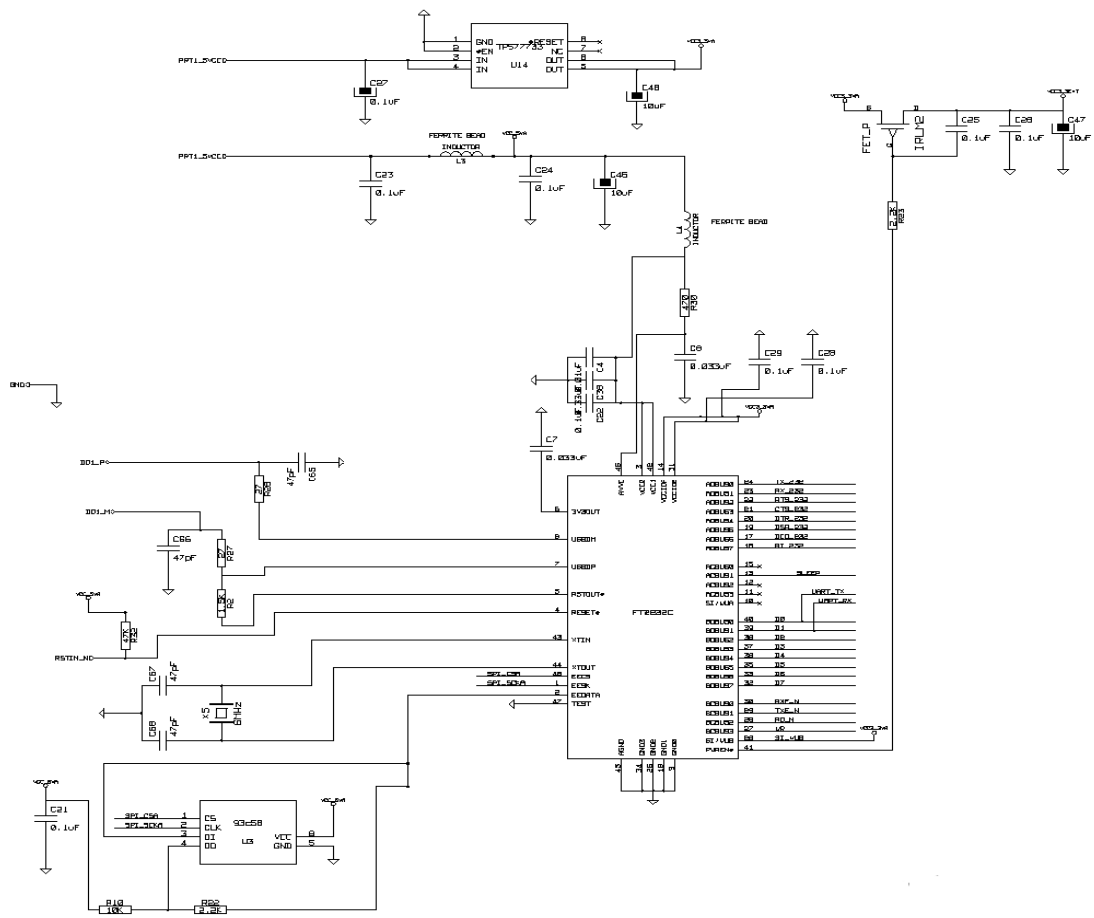


Figura 5.4. Circuito de Interfaz USB

### **5.2.1.3 Circuito de Módulo Serial.**

El circuito que constituye el módulo serial se puede observar en la figura 5.5 . Está constituido por el circuito básico de interfaz de USB basado en uno de los dos integrados FT2232D presentes en el diseño general ( figura 5.4). Este integrado está interconectado al módulo de control mediante las líneas de RX y TX (pines 39 y 40 respectivamente) del FT2232D y además conectado a un circuito constituido por un circuito integrado, cuya función es convertir señales de nivel TTL, en señales de nivel RS232

Para el diseño de éste módulo se tomó en cuenta no sólo la necesidad de contar con una forma de serializar los datos entrantes y salientes del puerto USB, sino que además se tomó en cuenta la sugerencia de la empresa canalizada a través del asesor, para establecer una conexión con el microcontrolador presente en el módulo de Control, de forma que en un futuro el microcontrolador pudiese ser empleado para actividades de desarrollo y depuración a través el puerto serial

La metodología de diseño consistió en conectar los pines de transmisión serial tanto del FT2232D con los del microcontrolador MSP430FG439 del módulo de Control. Seguidamente se ubicó y seleccionó de entre un conjunto de propuestas a nivel comercial, un circuito integrado multicanal que permitiese realizar el cambio entre niveles de tensión LVTTTL o TTL a RS232 y viceversa de forma independiente para el conjunto de líneas empleadas en el protocolo RS232. El circuito seleccionado fue el MAX211 de Texas Instruments.

Los capacitores empleados en el diseño del circuito constituido por el integrado MAX211, fueron seleccionados, de acuerdo a un rango de posibles valores que el fabricante sugirió emplear.

Con respecto a la selección del modo de funcionamiento del circuito integrado, el Modo Serial fue seleccionado para ser implementado en el puerto B del primer circuito FT2232D. La justificación del empleo del puerto B para este

modo, no obedece a ningún criterio específico dado que este modo puede ser implementado en ambos puertos de igual forma.

Una particularidad que se debe tomar en cuenta, es que se sugiere emplear en el diseño un conector DB9 con sus pines en ángulo recto. Este requerimiento obedece al hecho de que en caso de realizarse un circuito impreso, el conector con pines en ángulo recto, resulta mucho más sencillo de colocar en los bordes de la tarjeta, por lo que el espacio destinado a un conector se reduce considerablemente.

### **Funcionamiento:**

Como se describió anteriormente, cada vez que es energizado el circuito FT2232D, éste lee los parámetros de configuración almacenados en la memoria. Entre estos parámetros se encuentra la información relacionada con el modo de uso de cada puerto que también es leída. Una vez leídos estos datos, la interfaz USB-Serie queda establecida, permitiendo realizar la conversión de datos del protocolo USB a serie y viceversa.

La comunicación de este módulo con el concentrador USB o bien con el computador mediante el protocolo USB se da a través de los pines DD2\_M y DD2\_P quienes son los encargados de recibir y enviar datos.

La alimentación de éste módulo se da a través del pin PRT2\_5VCC, que también es controlado por circuito integrado TP2044B integrado en el módulo de concentrador USB.

Con respecto a la alternativa de conexión empleando el módulo Serial y el microcontrolador presente en el módulo de Control, su funcionamiento queda sujeto al uso que la empresa le quiera dar.



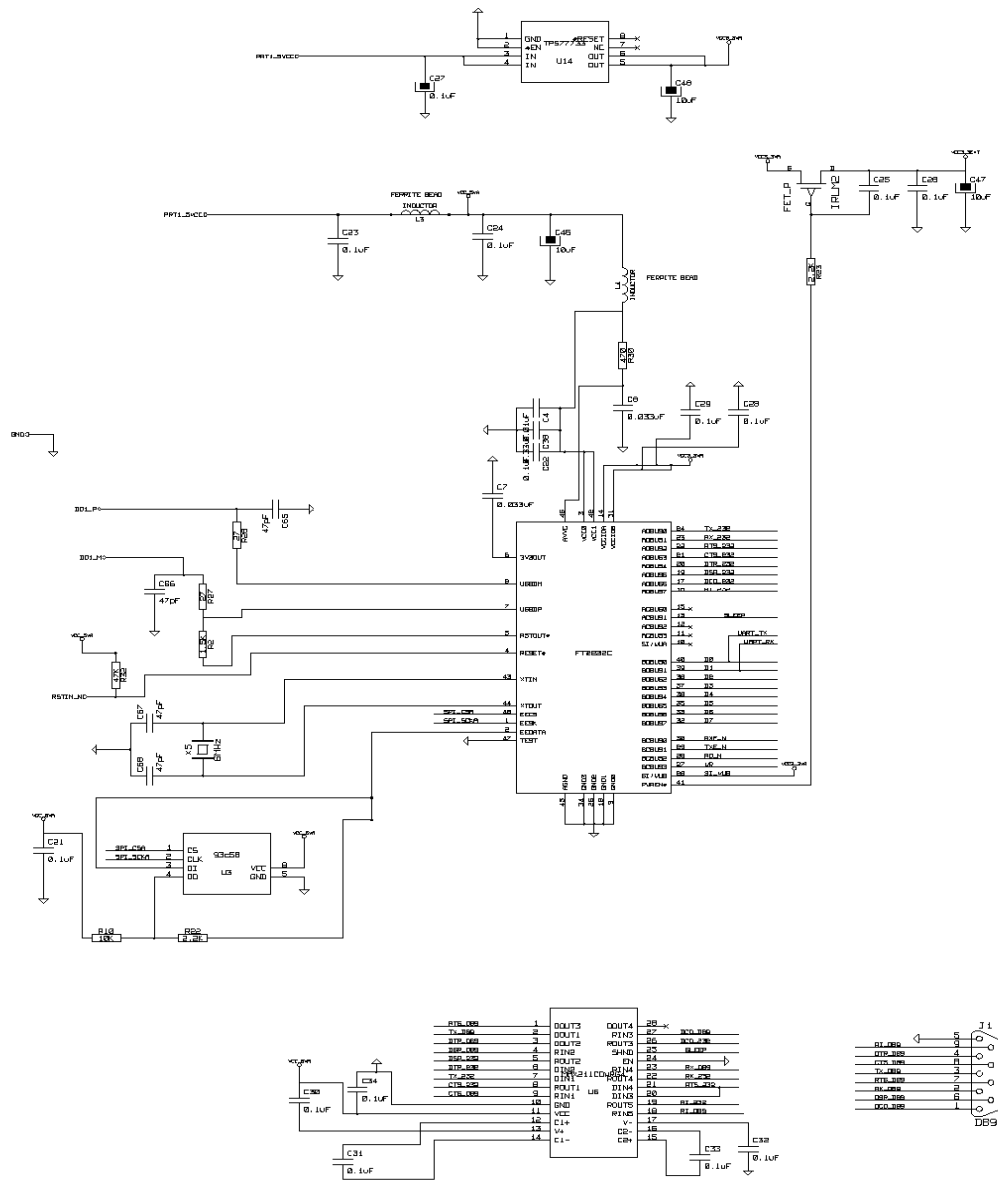


Figura 5.5 Circuito de módulo Serial

#### 5.2.1.4 Circuito de Módulo de Control.

El módulo de Control, como se mencionó anteriormente es el encargado de dirigir y administrar las funciones de los módulos LCD, Voltímetro, Amperímetro y eventualmente cualquier módulo o aplicación que se requiera desarrollar a través de la conexión con el módulo Serial.

Este módulo está constituido por un microcontrolador MSP430FG439 de Texas Instruments. La selección del microcontrolador empleado, estuvo basada principalmente en aspectos eléctricos, físicos y económicos: disipación de potencia, resolución de los convertidores analógico digital, funciones, tamaño del encapsulado y precio.

De la misma manera que los módulos: Serial, Jtag y Expansión, este módulo se encuentra basado en el circuito integrado FT2232D. La interconexión del microcontrolador con el FT2232D fue realizada a través del protocolo Paralelo-FIFO, utilizando 8 líneas para datos y 4 líneas para el control, empleando para ello el canal A de este circuito integrado.

En la siguiente tabla se describe el la función asignada a cada uno de los pines del microcontrolador empleados en el presente proyecto

**Tabla 5.4: Distribución de pines del microcontrolador presente en el módulo de Control**

Pin #	Referencia	Tipo	Función	Descripción
1	DVcc1	Entrada	Alimentación	Alimentación 3.3V
2	P6.3	Bidireccional	GPIO1	Pin Propósito General
3	P6.4	Bidireccional	GPIO2	Pin Propósito General
4	P6.5	Bidireccional	GPIO3	Pin Propósito General
5	P6.6	Bidireccional	GPIO4	Pin Propósito General
6	P6.7	Bidireccional	GPIO5	Pin Propósito General
7	VREF	Salida	Referencia	Salida de tensión de referencia de ADC
8	XIN	Entrada	Oscilador	
9	XOUT	Salida	Oscilador	

Pin #	Referencia	Tipo	Función	Descripción
10	VeREF+	Entrada	Referencia	Referencia para ADC
11	VREF	Salida	Referencia	Referencia para ADC
12	P5.1	Salida	RSTIN_N	Reset FT2232A
13	P5.0	Bidireccional	GPIO6	Pin Propósito General
14	P4.7	Salida	DB0	Bit 0 LCD
15	P4.6	Salida	DB1	Bit 1 LCD
16	P4.5	Salida	DB2	Bit 2 LCD
17	P4.4	Salida	DB3	Bit 3 LCD
18	P4.3	Salida	DB4	Bit 4 LCD
19	P4.2	Salida	DB5	Bit 5 LCD
20	P4.1	Salida	DB6	Bit 6 LCD
21	P4.0	Salida	DB7	Bit 7 LCD
30	P2.7	Entrada	INT_SPI_U	Solicitud Int SPI
31	P2.6	Entrada	INT_ETH	Solicitud Int mod Ethernet
36	P3.7	Salida	CS_SPI_U	Chip Select SPI
37	P3.6	Salida	RST_SPI_U	Reset SPI
38	P3.5	Salida	RST_ETHN	Reset Ethernet
39	P3.4	Salida	CS_ETHN	Chip Select Ethernet
40	P3.3	Salida	SPI_SCKG	Clock SPI
41	P3.2	Entrada	MISO	SPI Master In-Slave Out
42	P3.1	Salida	MOSI	Master Out-Slave In
45	P5.2	Salida	LCD_E	Enable LCD
46	P5.3	Salida	LCD_RW	Read/Write LCD
47	P5.1	Salida	LCD_RS	LCD Data/Instruction
49	P5.5	Bidireccional	GPIO7	Pin Propósito General
50	P5.6	Bidireccional	GPIO 8	Pin Propósito General
51	P5.7	Bidireccional	GPIO9	Pin Propósito General
52	DVcc2	Entrada	Alimentación	Alimentación lógica digital
53	DVss2	Salida	GND	Nodo Común
54	P2.5	Entrada	UART_RX	Recepción UART
55	P2.4	Salida	UART_TX	Transmisión UART
56	P2.3	Entrada	WR	Habilita el dato actual en D0..D7. Procesa el nuevo byte de datos del buffer de recepción
57	P2.2	Entrada	RD_N	Habilita el dato actual en D0..D7. Procesa el nuevo byte de datos del buffer de recepción
58	P2.1	Salida	TXE_N	Permite la escritura en el buffer cuando está en bajo

Pin #	Referencia	Tipo	Función	Descripción
59	P2.0	Salida	RXF_N	Indica que un dato deber ser leído en el buffer cuando su estado es bajo.
60	P1.7	Bidireccional	D7	Modo Paralelo-Fifo D7
61	P1.6	Bidireccional	D6	Modo Paralelo-Fifo D6
62	P1.5	Bidireccional	D5	Modo Paralelo-Fifo D5
63	P1.4	Bidireccional	D4	Modo Paralelo-Fifo D4
64	P1.3	Bidireccional	D3	Modo Paralelo-Fifo D3
65	P1.2	Bidireccional	D2	Modo Paralelo-Fifo D2
66	P1.1	Bidireccional	D1	Modo Paralelo-Fifo D1
67	P1.0	Bidireccional	D0	Modo Paralelo-Fifo D0
68	XT2OUT	Salida	Oscilador	
69	XT2IN	Entrada	Oscilador	
75	P6.0	Entrada	VMETER	Entrada Voltímetro
76	P6.1	Entrada	Ampmeter	Entrada Amperímetro
77	P6.2	Bidireccional	GPIO10	Pin Propósito General
78	AVss	Salida	GND	GND lógica Analógica
79	DVss	Salida	GND	GND lógica digital
80	AVcc	Entrada	Alimentación	Alimentación lógica digital

Por otra parte, los componentes que conforman el circuito de soporte del microcontrolador (conjunto de componentes que le permiten funcionar al microcontrolador como cristales y capacitores) fueron empleados basándose en los valores que sugiere el fabricante en la hoja de datos [13].

### **Funcionamiento:**

Tanto el funcionamiento como el diseño de este módulo están más vinculados a la sección de software. Por lo tanto es allí donde se puede encontrar la información referente a este módulo. Sin embargo se debe mencionar que al igual que para los casos anteriores y dado que el módulo está basado en el circuito integrado FT2232, el sistema adquiere los datos de configuración del dispositivo USB almacenados en la memoria EEPROM, y el modo que será empleado en cada uno de los puertos del FT2232D justo después que el

circuito es energizado. En el modo Paralelo-FIFO cuando un dato (tamaño máximo 128 bytes) es enviado desde el sistema anfitrión (PC), este queda almacenado en el buffer de recepción que posee el FT2232D, el cual puede ser leído al manipular el pin RD\_N. Cuando se desea enviar un dato desde un periférico hacia el sistema anfitrión, éste debe ser escrito en el buffer de transmisión. Esto se logra al manipular el pin WR. Una vez que ha sucedido esto, el sistema anfitrión es el encargado de remover el dato para su respectivo proceso.

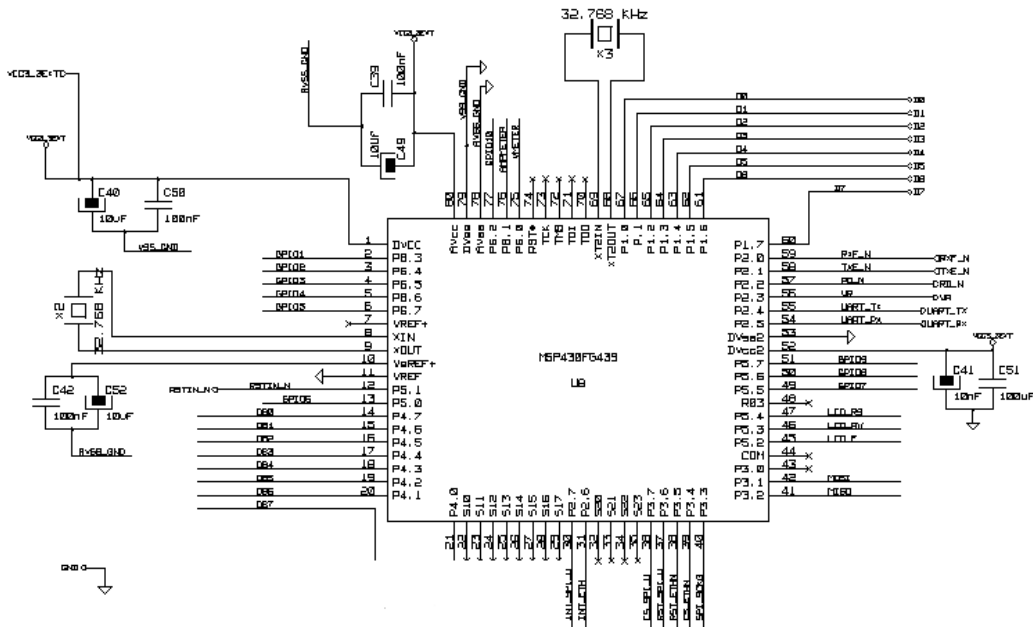
Por otra parte, la señal TXE\_N permite la escritura en el buffer cuando está en bajo. Cuando está en alto no permite la escritura del dato en el buffer. La señal RXF\_N indica cuando un dato deber ser leído en el buffer, cuando su estado es bajo. Cuando está en alto indica que el dato aún no debe ser leído.

Con respecto a la transmisión y recepción de datos, esto se hace por medio de las líneas D0 a D7 que son líneas de datos bidireccionales.

Por otra parte, los pines UART\_TX y UART\_RX son empleados en este módulo en caso de que una transmisión serial sea requerida. Mediante el pin UART\_TX los datos son transmitidos, mientras que los datos son recibidos mediante el pin UART\_RX.

Por otro lado, con respecto a la medición de valores mediante los convertidores analógico-digital presentes en microcontrolador que integra este módulo, basta con aplicar a los pines llamados AMPMETER y VOLTMETER una tensión dentro del rango de tolerancia descrita por el fabricante del microcontrolador. La conversión es detallada en la sección de Software.

La polarización del módulo se hace mediante la línea VCC3\_3EXT.



**Figura 5.6 Circuito del módulo de Control**

### 5.2.1.4.1 Submódulo Voltímetro

El Voltímetro es un módulo asociado al módulo de Control, pero dada su dependencia con éste módulo, se puede ser llamado un submódulo, dada su falta de autonomía para funcionar sin recurrir al módulo Control.

Su diseño está basado en la necesidad o especificación de la empresa de medir tensiones en un rango entre 0V y 33V. El rango de medición de tensión por parte del voltímetro, es considerado más que apropiado, dado que en el desarrollo de sistemas empotrados, difícilmente se supera el valor máximo indicado.

El circuito del voltímetro está constituido por un convertidor analógico-digital (ADC) integrado en el microcontrolador y un circuito acondicionador de señal que cumple la función de atenuador, de forma que permita la medición de hasta 33V, tomando

en cuenta que la tensión máxima posible que puede tolerar y medir el ADC es 3.3V

La forma más simple que se halló para realizar el diseño fue mediante el empleo de un circuito divisor de tensión de alta impedancia, de forma que funcionase como circuito acondicionador de señal. Para evitar cargar el circuito, se seleccionaron resistencias en el orden de los k $\Omega$ .

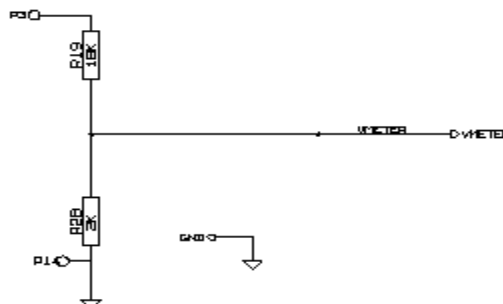
Con respecto a la relación del voltaje de entrada vrs voltaje de salida, se seleccionó un valor de 10, dado que éste valor permitía simplificar enormemente el desarrollo del software propio del microcontrolador.

### **Funcionamiento:**

Para lograr medir una tensión desde el punto de vista del funcionamiento de hardware, el usuario debe aplicar una tensión que desee medir en el nodo P3 (ver esquemático) . Esta tensión aplicada será leída por el convertidor analógico-digital mediante el pin 75 del microcontrolador, pero disminuida 10 veces con respecto a su valor original, dada la función del circuito acondicionador de señal, con el fin de evitar dañar el microcontrolador al aplicarle tensiones superiores a 3.3V.

El tratamiento que se le da al dato una vez leído, se describe en la sección de software.

La siguiente figura muestra el circuito acondicionador de señal empleado para la medición de tensiones



**Figura 5.7 Circuito divisor de tensión como acondicionador de señal**

#### **5.2.1.4.2 Submódulo Amperímetro**

El Amperímetro es un otro módulo asociado al módulo de Control, el cual dada su dependencia con éste módulo, también puede ser llamado un submódulo.

El requerimiento de un amperímetro está basado en la necesidad de la empresa de medir el consumo de corriente de los dispositivos que desarrolla, principalmente para hacer estudios relacionados con baterías.

El diseño del amperímetro está basado en un sensor de corriente de efecto Hall ACS712ELCTR-20A-T de Allegro MicroSystems capaz de medir hasta 20A. Este fue diseñado por el fabricante para tener una respuesta lineal en la tensión de salida ante los cambios de corriente (principal parámetro por el cual el sensor fue seleccionado). De acuerdo a la hoja de datos, la tensión de salida del sensor para cuando la corriente sensada es cero, es de 2.5V, y este aumenta 100mV por cada amperio sensado hasta un máximo de 20A a una tensión de 4.5V.

Ahora bien, dado que el valor máximo que se desea medir es 2A el voltaje de salida del sensor nunca será mayor a 2.75 V aproximadamente, razón por la cual se puede acoplar directamente la salida del sensor con uno de los canales del convertidor analógico-digital del ADC.

Para limitar el rango de acción del sensor y evitar que se mida una corriente mayor, provocando un aumento de la tensión de salida del sensor y consecuentemente dañando el microcontrolador, se integró en el diseño del medidor de corriente, un fusible de 2A, de forma que al superarse este valor de corriente, el fusible se destruya y por ende, el microcontrolador no se llegue a dañar.



## Funcionamiento:

El usuario para medir una corriente, deberá conectar la entrada de corriente al conector P1 (ver esquemático en la sección de anexos), y la salida de la corriente al conector P2, de forma que la corriente establezca una trayectoria a través de este. El sensor tendrá en su pin de salida ( pin 7) un voltaje proporcional a la corriente que circula a través del sensor. Este pin es conectado directamente al pin 76 del microcontrolador llamado AMPMETER.

La interpretación dada al dato por el microcontrolador será discutida en la sección de software.

En la siguiente figura se puede apreciar el circuito medidor de corriente en el cual se basa el módulo del amperímetro.

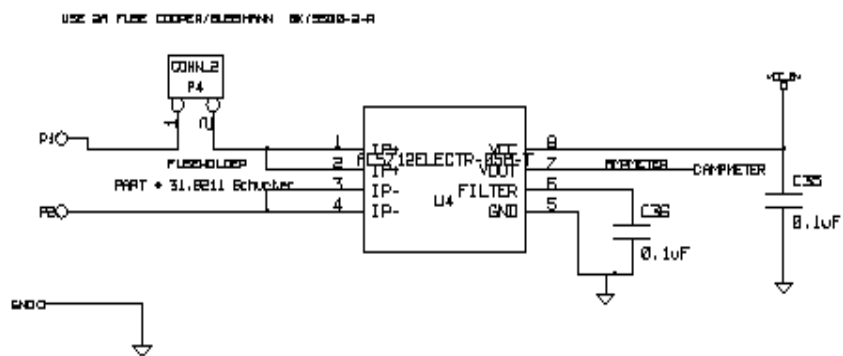


Figura 5.8. Circuito sensor de corriente

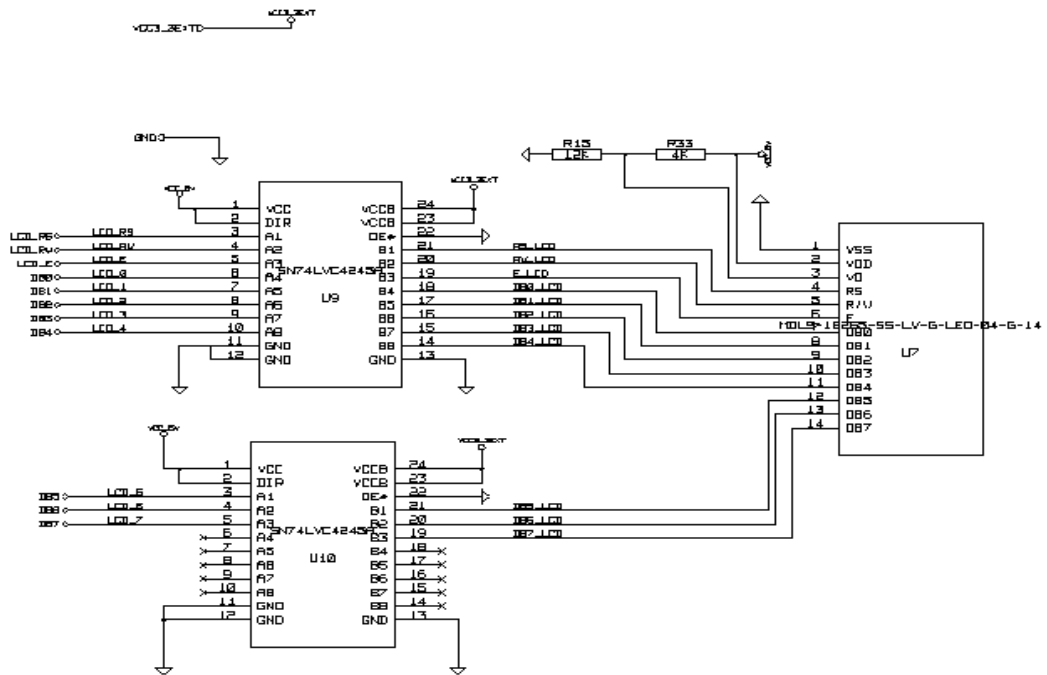
#### **5.2.1.4.3 Submódulo LCD**

El submódulo LCD cumple la función de desplegar los datos adquiridos tanto por el voltímetro, como por el amperímetro, y también puede ser empleado para cualquier función donde se requiera desplegar datos para el usuario en un futuro.

Este módulo depende completamente de las acciones de control del microcontrolador, el cual es parte del módulo de Control.

La selección de el LCD empleado en este proyecto, se justifica mediante la necesidad de desplegarle al usuario lo datos adquiridos mediante un dispositivo de fácil manejo. Este LCD basado en el controlador H44870 de Hitachi hace que sea sencillo de programar y utilizar. Las rutinas de funcionamiento del LCD se pueden observar en la sección de software.

En la siguiente figura se puede observar el circuito del LCD con sus respectivas conexiones.



**Figura 5.9 Circuito de conexión al módulo de LCD**

En el circuito de la figura 5.9 se emplearon los circuitos integrados SN74LVC4245A. Su empleo se justifica en el requerimiento de interfazar la lógica LVTTTL con la cual funciona el microcontrolador, con la lógica TTL con la cual funciona el LCD. Estos circuitos integrados cumplen la función de convertir los niveles de las señales de una lógica a otra para que el LCD pueda funcionar correctamente.

El divisor de tensión conformado por las resistencias R15 y R33 tiene la función de fijar una tensión de referencia con la cual se pueda ajustar el contraste de la pantalla y por ende sea visible por el ojo humano. En caso de requerir un contraste ajustable para distintas intensidades de luz, ambas resistencias pueden ser sustituidas por un potenciómetro.

### 5.2.1.5 Circuito de Módulo de Jtag.

El módulo de Jtag está basado al igual que el módulo Serial, el módulo de

Control y el módulo de Expansión, en el circuito integrado FT2232.

Para este módulo se implementó el módulo Multiprotocolo a través del puerto A del segundo circuito integrado FT2232D presente en el diseño general.

Se debe mencionar que la selección del modo Multiprotocolo no se puede realizar de la misma manera que el modo Serie y Modo Paralelo. La selección de este modo se hace a través de software, sin embargo una particularidad del modo Multiprotocolo radica en que el firmware del circuito integrado FT2232D no implementa todo el protocolo Jtag, sino que únicamente permite configurar al dispositivo para que el protocolo pueda funcionar.

Como el objetivo de la empresa es emplear este módulo con procesadores ARM de Texas Instruments, y tomando en cuenta la filosofía de la empresa, se empleó el software OpenOCD de distribución libre para el manejo de todo este protocolo.

El diseño de este módulo estuvo orientado a buscar la forma de interfazar el canal A del circuito FT2232D configurado en modo Multiprotocolo, con el principal dispositivo empotrado en desarrollo de RidgeRun. Este dispositivo utiliza un procesador ARM9, por lo que la forma más simple que se halló para realizar esta interfaz, fue mediante un cable diseñado para tal fin. Dado que no sólo para un único dispositivo será empleada esta herramienta, es requerido estandarizar cualquier conexión sobre la cual basar el diseño del cable antes mencionado. Tomando en cuenta estos factores, y como sugerencia de la empresa, se optó por asumir como base una distribución de 20 pines llamada "ARM 9 JTAG" . Así pues los pines del puerto A del circuito integrado FT2232D designados para ser usados en este módulo, son asociados a un conector #3421-6000 de la marca 3M. La distribución de pines de la conexión ARM 9 JTAG es la siguiente:

TMS	1	2	TRST
TDI	3	4	GND
TVD	5	6	GND
TD0	7	8	GND
TCK-RET	9	10	GND
TCK	11	12	GND
EMU0	13	14	GND
SRST	15	16	GND
EMU2	17	18	GND
EMU4	19	20	GND

**Figura 5.10 Conexión ARM9 JTAG**

La distribución de pines requerida para ser empleada en el principal dispositivo en desarrollo es llamada “Voco ARM 9” de 14 pines, la cual emplea un conector FFSD-10-D-06.00-01-N de la marca Samtec.

La distribución de pines de la conexión “Voco ARM 9” es la siguiente

TMS	1	2	TRST -
TDI	3	4	GND
PD (+5v)	5	6	no pin
TD0	7	8	GND
TCK-RET	9	10	GND
TCK	11	12	GND
EMU0	13	14	EMU1

**Figura 5.11 Distribución de pines para la conexión Voco ARM9**

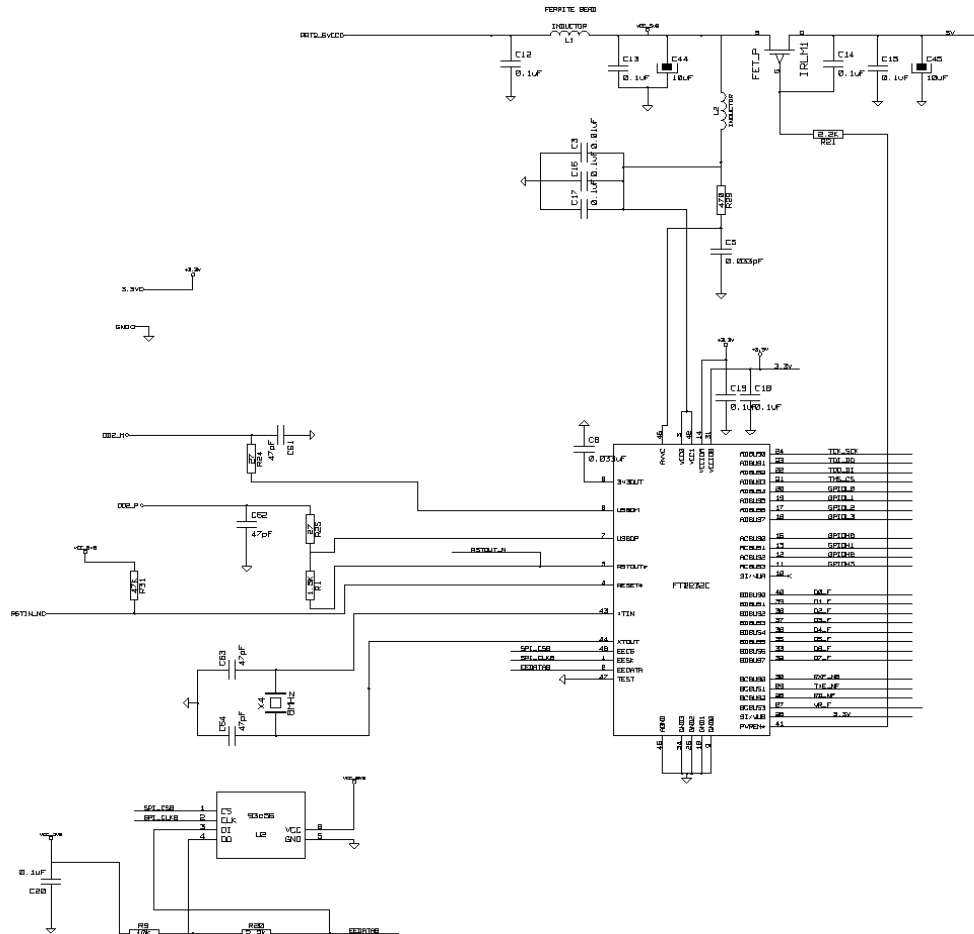
Analizando las anteriores configuraciones, se construyó un cable con la siguiente distribución de pines.

**Tabla 5.5: Conexión de los pines para el cable de interfaz**

ARM 9 Jtag	Voco ARM 9	Señal
1	5	VTREF
2	NC	VSUPPLY (no usado)
3	2	TRST-
4	4	GND
5	3	TDI
6	4	GND
7	1	TMS
8	4	GND
9	11	TCK
10	4	GND
11	9	RTCK
12	4	GND
13	7	TDO
14	4	GND
15	NC	SRST-
16	4	GND
17	NC	BDGRQ
18	4	GND
19	NC	BDGACK
20	4	GND

NC: No conectar

El circuito que integra este módulo se puede observar en la siguiente figura:



**Figura 5.12 Circuito del módulo JTAG**

**Funcionamiento:**

El funcionamiento de este módulo, dado está documentado en la sección de software.

**5.2.1.6 Circuito de Módulo de Expansión**

El circuito módulo de Expansión está incluido en el diseño general, con la intención de dejar previstas para cuando sea necesario, una gama de posibilidades para mejorar y actualizar el sistema desarrollado cuando se requiera.

A diferencia de los demás módulos, el módulo de Expansión está conformado por la unión de el puerto B que queda libre del segundo FT2232D incluido en el diseño general, más las líneas de propósito general (GPIO) concentradas en el conector P9, junto con las líneas para conectar un dispositivo SPI cuando sea requerido concentradas en el conector P10.

En la siguiente figura se muestran los elementos que componen este módulo. Cabe aclarar que en el circuito esquemático global, los elementos que integran esta figura, se encuentran distribuidos de acuerdo al circuito integrado que pertenecen.

Con respecto al funcionamiento de este módulo, este está sujeto a los requerimientos y funciones que se le quieran otorgar.



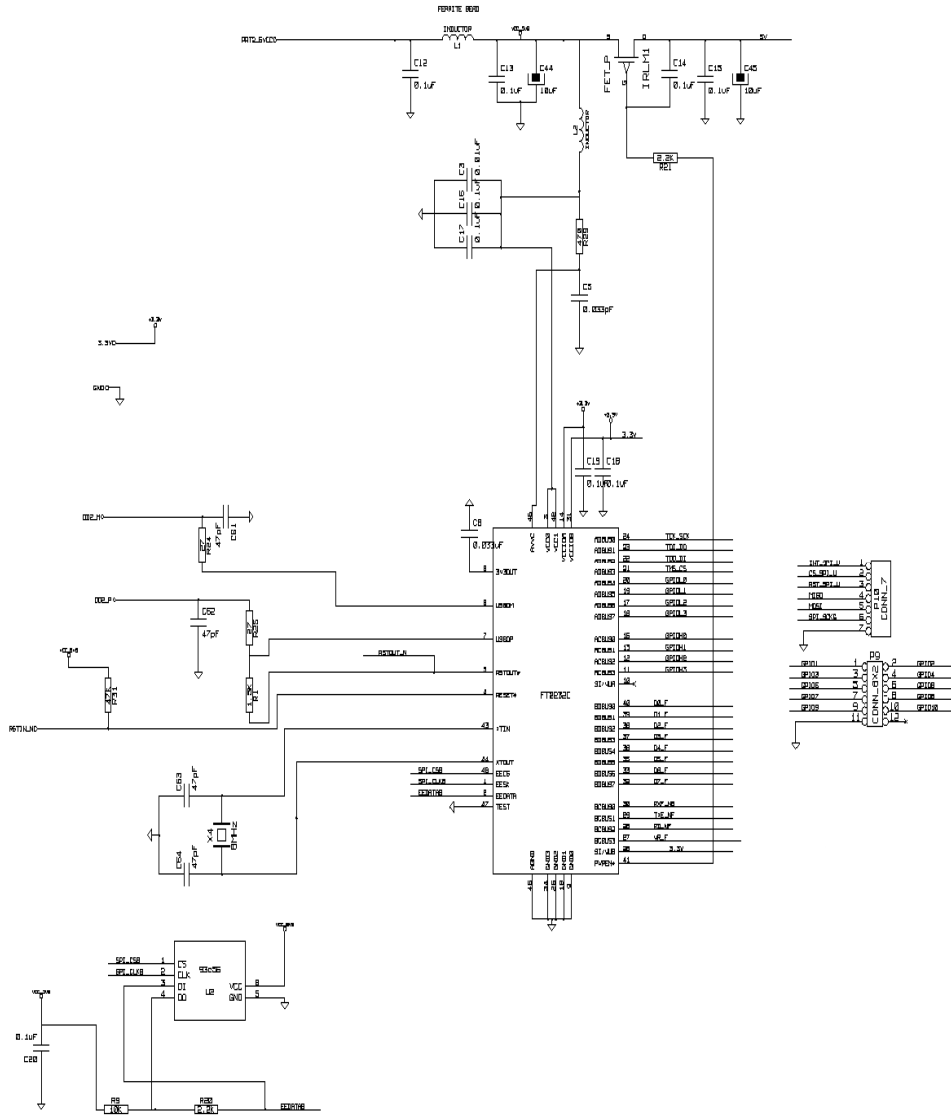


Figura 5.13 Módulo de Expansión

### 5.2.1.7 Circuito de Módulo de Fuente CD.

Sin duda alguna, este módulo tiene la probabilidad de ser la herramienta más empleada de todo el sistema diseñado, ya que por su naturaleza, es indispensable para energizar los dispositivos empotrados con los que se pretende trabajar.

El diseño de la fuente está fundamentado en la petición de la empresa de contar con una fuente de alimentación para sistemas empotrados, con valores de tensión predefinidos, seleccionados por el usuario de forma digital. Cabe destacar que como el tamaño de éste módulo tendría una incidencia directa en el tamaño del sistema general, se optó por diseñar el circuito de selección de las tensiones de salida, y para la alimentación del mismo, se optó por emplear un dispositivo de uso cotidiano, llamado comúnmente “adaptador AC”, el cual es una unidad convertidora de tensión AC a tensión DC para uso casero, con tal de no incluir dentro del sistema desarrollado una etapa rectificadora de tensión, dado que sólo el tamaño del transformador, generalmente tiende a ser relativamente grande.

Como tensión de entrada para el módulo de la fuente CD se emplean 6V dado que la mayoría de los circuitos empotrados funciona con tensiones menores a ésta, y es relativamente fácil conseguir en cualquier comercio un adaptador AC de buena calidad.

Para el diseño de la etapa de selección de tensiones, se empleó el regulador de tensión variable LM317HVT de National Semiconductor el cual es capaz de entregar una corriente de 1.5A y cuenta con protección contra cortocircuitos

El comportamiento de la tensión de salida de éste regulador de tensión está regido por la siguiente ecuación:

$$V_O = \left(1 + \frac{R_2}{R_1}\right) + (I_{ADJ} * R_2) \quad \text{EC (5.1)}$$

Para el valor de  $I_{ADJ}$  se toma como constante el valor de  $100 \mu A$ , ya que se emplea el criterio de diseño de utilizar una corriente de ajuste cuyo valor sea para el peor de los casos.

Si se selecciona el valor de  $R1$  en  $240\Omega$  por recomendación del fabricante, se puede despejar el valor de  $R2$  para diferentes valores de tensión requeridos.

Los valores de las resistencias  $R35, R36, R37, R38, R39, R40$  y  $R41$  de la figura 5.14 fueron encontrados mediante el despeje de la ecuación 5.1 para distintos valores de tensión previamente sugeridos por la empresa..

En la tabla 5.6 se pueden observar los diferentes valores de resistencias requeridos para las tensiones de salida predefinidas sugeridas por RidgeRun

**Tabla 5.6: Valores de tensión de salida para diferentes valores de resistencia R2 empleados**

Valor de Salida (V)	Valor de Resistencia R2 en Ohmios
1.5	48.7
1.8	107
2.5	240
3.0	340
3.3	390
3.6	453
5.0	715
Tensión Variable	Potenciómetro de 1K

Por otra parte, uno de los requisitos para esta fuente, por parte de la empresa, es que esta pueda ser controlada digitalmente, pero que no se dependa del hecho de tener el computador encendido para poder realizar los cambios en la tensión de salida. Dadas estas circunstancias, se emplea un circuito integrado 74HC238 el cual es un decodificador de 3 a 8 líneas. Mediante la variación de

los estados lógicos de los pines de entrada de este circuito por medio de *jumpers*, como se muestra en la tabla 5.7, es posible realizar la selección de la tensión de salida.

Al ponerse en alto un pin de salida específico y el resto de pines en bajo en el decodificador, se polariza la base de un transistor BJT. Cada circuito integrado MMPQ3904 es en realidad un arreglo de 4 transistores BJT 2N3904.

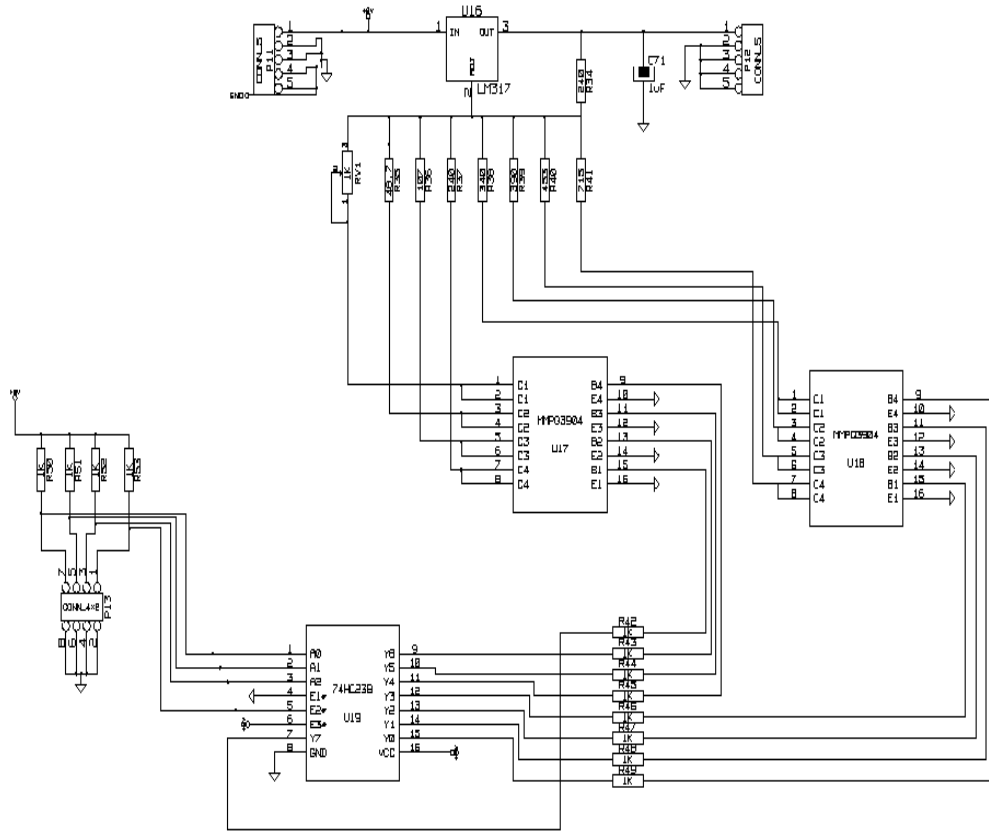
Al polarizarse la base del transistor BJT, este entrará en estado de conducción, y por tanto la ruta seleccionada permitirá que la corriente encuentre su camino hacia tierra, pasando a través de la resistencia que define el valor de tensión de salida.

**Tabla 5.7: Ubicación de jumpers para la selección de la tensión de salida de la fuente CD en los pines del conector P13**

Pines 7-8	Pines 5-6	Pines 3-4	Pines 1-2	Tensión de Salida (V)
OFF	ON	ON	ON	1.5
OFF	ON	ON	OFF	1.8
OFF	ON	OFF	ON	2.5
OFF	ON	OFF	OFF	3.0
OFF	OFF	ON	ON	3.3
OFF	OFF	ON	OFF	3.6
OFF	OFF	OFF	ON	5.0
OFF	OFF	OFF	OFF	[1.5-5.0]

Nota: ON debe entenderse como un jumper colocado entre dos pines.

OFF debe entenderse como jumper sin colocar entre un par de pines.



**Figura 5.14 Circuito del módulo de Fuente CD**

### **5.3 Descripción del software**

En esta sección se describirá el software empleado en cada uno de los módulos que ameritan.

#### **5.3.1 Programa Usuario.**

El usuario requiere de una interfaz en el computador tanto para seleccionar el módulo que desea emplear, ya sea voltímetro o bien amperímetro, y además para recibir los datos producto de las mediciones realizadas con estos módulos.

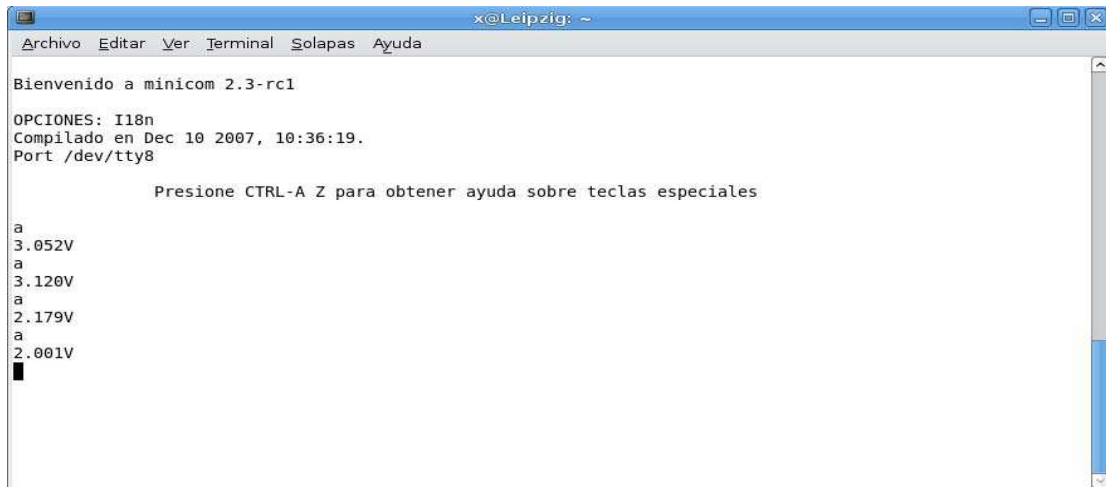
Para realizar la selección del módulo, el microcontrolador debe recibir un carácter Ascii enviado por el computador a través del puerto USB. El microcontrolador, dependiendo de si el carácter recibido es una “a” o una “d”, habilita el voltímetro o bien una “e”, que habilita el amperímetro.

Para la empresa, el programa usuario no es importante, dado que en un futuro pretende asignar a un ingeniero a desarrollar una interfaz de control. Para ello recomendó emplear como programa para el usuario, el software Minicom.

Minicom es un software para comunicaciones seriales empleado en Linux y utilizado desde interfaz de línea de comando. Minicom permite enviar y recibir datos tanto a los puertos serie del computador, como a los puertos USB que tengan alguna especie de dispositivo serializador conectado, como es el módulo USB basado en el circuito integrado FT2232D en el presente proyecto.

En la figura 5.15 se puede observar la captura de pantalla realizada a Minicom, y la forma como el usuario enviaría y recibiría datos. El carácter “a” que se muestra en la figura sobre el primer valor de tensión desplegado, implica que

el usuario desea usar el voltímetro. En la siguiente línea, se observa el valor medido por el voltímetro y la forma como es desplegado. Así pues, para cada medición que se quiere realizar, es necesario enviar un carácter al sistema.



**Figura 5.15** Captura de pantalla de la forma en que el usuario vería los datos mediante el empleo de Minicom

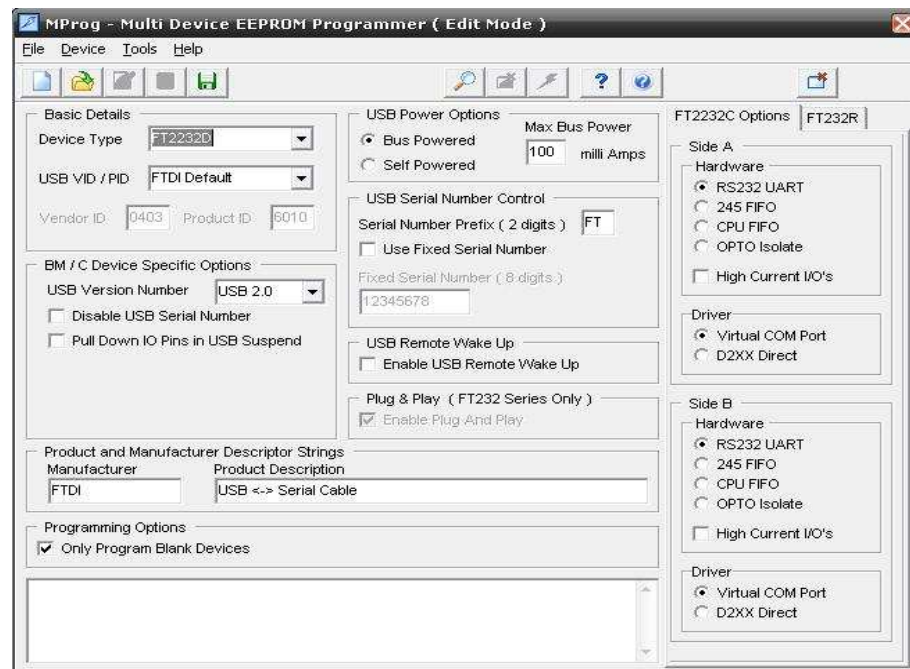
### 5.3.2 Programación del circuito FT2232D

El almacenamiento de los datos de configuración en la memoria EEPROM que lee el circuito integrado FT2232D se hace a través del programa MPROG, provisto por el fabricante.

En la figura 5.16 se puede observar la pantalla correspondiente al software de programación. Es en esta ventana donde se ingresan los datos de configuración tales como: identificación del fabricante, descripción del dispositivo, así como la forma en que será energizado, ya sea a través del bus USB, o bien de forma independiente.

La zona de la derecha de la figura, le muestra al usuario las opciones para

configurar cada uno de los puertos del circuito integrado FT2232D. Para este caso únicamente vale la pena recalcar que al seleccionar la opción RS232 UART, ya sea para el puerto A o el B, este funcionará en modo Serie. Si la opción seleccionada es 245 FIFO, el funcionamiento del puerto será en modo paralelo.



**Figura 5.16** Pantalla correspondiente al software MPROG para programar el circuito integrado FT2232D

### 5.3.3 Programación del Microcontrolador

El programa que controla el microcontrolador MSP430FG439 es el software más importante desarrollado en el presente proyecto, dado que por medio de este se logra el funcionamiento del módulo de control, y por ende el funcionamiento de los módulos que de él dependen, como lo es el caso del el módulo que comprende al amperímetro, voltímetro y LCD.



El software para el microcontrolador fue desarrollado mediante el empleo de la herramienta de desarrollo MSPGCC, una herramienta basada en la filosofía de código libre basada en el lenguaje de programación C [14] [15].

Para simplificar el desarrollo del código, éste se dividió en 11 rutinas o procedimientos, de forma que todos fuesen dependientes del módulo principal.

En la siguiente tabla se resume la función de cada procedimiento desarrollado. En caso de que se quiera profundizar, se puede consultar la sección de anexos donde se lista el código completo.

**Tabla 5.8: Resumen de los procedimientos desarrollados en el software del microcontrolador**

<b>Procedimiento</b>	<b>Función</b>
Delay	Genera ciclos de no operación en el microcontrolador
Sendbyte	Escribe un carácter en el LCD
Clean_LCD	Limpia la pantalla del LCD
Display_off_LCD	Deshabilita la visualización de caracteres en el LCD
Cursor_Home	Envía el cursor a la posición inicial del LCD
Init_LCD	Inicializa el LCD para que muestre información
Pos_LCD	Desplaza el cursor una posición hacia adelante
volt	Realiza el cálculo de la tensión basado en el valor leído por el ADC
volt_10	Realiza el cálculo de la tensión basado en el valor leído por el ADC tomando en cuenta la escala
current	Realiza el cálculo de la corriente basado en el valor leído por el ADC
Main	Procedimiento principal que controla el llamado a los demás procedimientos.

### **Funcionamiento del software:**

La figura 5.17 ilustra el funcionamiento del software mediante un diagrama de flujo

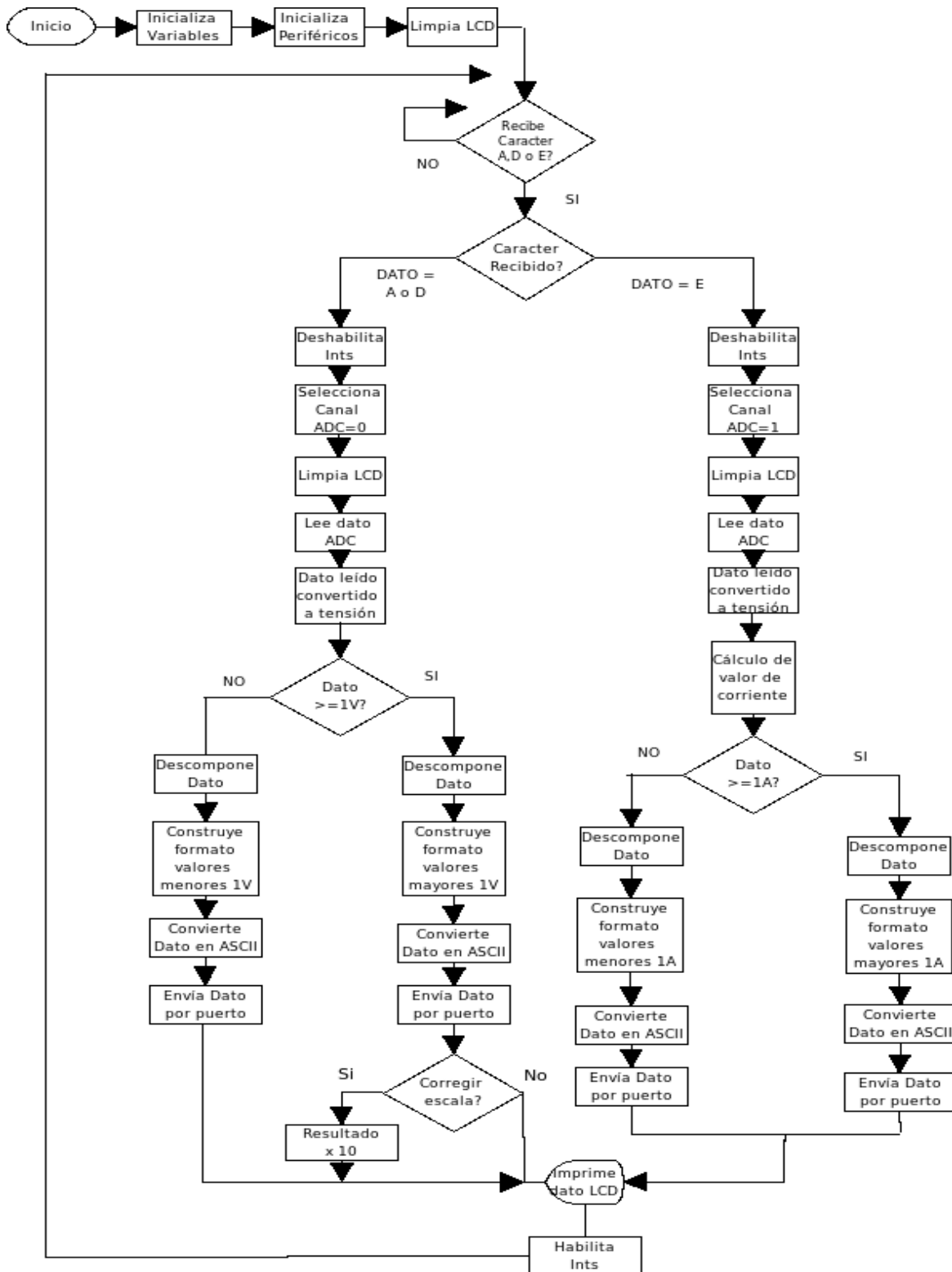


Figura 5.17 Diagrama de flujo del software desarrollado

Cuando el sistema es encendido, las variables del programa son inicializadas. Seguidamente se ejecuta el método principal o *main*. La primera acción que este método ejecuta es la inicialización de periféricos y habilitación de interrupciones. Además se realiza la definición de la función que ejecutará cada pin configurable en el microcontrolador, ya sea como de función especial, o bien de propósito general. También se configura cada pin de acuerdo al sentido de la información: entrada, salida, o bidireccional.

El siguiente método que es ejecutado a través del método principal, es el método encargado de realizar una limpieza del LCD, dado que cuando este es energizado, caracteres indeseados son impresos en la pantalla.

Después de ejecutados estos primeros métodos, el software a través del método principal, hace que el microcontrolador entre en un ciclo infinito, donde espera constantemente hasta que se de el arribo de un carácter ASCII, ya sea una "a", una "d" o bien una "e" enviado por el usuario.

Al recibir una carácter "a" o "d" el módulo medidor de tensiones es automáticamente habilitado.

El carácter "a" le indica al sistema que las tensiones medidas estarán en el intervalo de 0V a 3.3V, mientras que si el carácter "d" le indica al sistema que la medición de tensiones debe realizarse con un ajuste de escala dado que el rango de medición es de 1V a 33V.

Seguidamente la atención de las interrupciones es inhabilitada, para evitar de esta forma que una nueva interrupción detenga el procedimiento que se está dando a cabo. Como el circuito acondicionador de señal del voltímetro mencionado en la sección 5.2.1.4.1 está conectado al canal 0 del ADC, éste es seleccionado. Como manera de precaución, inmediatamente es ejecutado el método de limpieza de pantalla para evitar que un dato existente pueda

confundir al usuario, y por ende se de una lectura errónea.

Una vez que el dato es leído por el ADC, el valor registrado es convertido a un valor de tensión proporcional a la lectura.

La conversión del valor proporcional dado por el ADC para obtener un valor de tensión se basa en la siguiente ecuación:

$$Voltaje = \left( \frac{N_{ADC} * 3.052}{4095} \right) \quad \text{EC(5.2)}$$

en donde  $N_{ADC}$  es el valor correspondiente a la conversión analógico-digital del ADC, el cual varía de 0000 en 0V a 4095 EN 3.3 V.

Al finalizar el procedimiento de conversión, el dato correspondiente a la tensión leída por el ADC, es desplegado con base en la existencia o no del requerimiento de corrección de escala.

El despliegue del dato leído se hace tanto en el LCD como en el PC del usuario. Como la recepción de la información que el usuario ve en pantalla se da a través de el software Minicom, los datos deben ser enviados en formato Ascii, sin embargo antes de realizar la conversión de estos, se ejecuta un procedimiento que descompone o separa cada uno de los dígitos que conforman el valor total leído con tal de simplificar la conversión.

La conversión del dato obtenido, al formato ASCII es realizada mediante la operación OR del dato obtenido con un 30 hexadecimal.

Una vez terminados estos procedimientos, las interrupciones son habilitadas nuevamente y el sistema queda a la espera de la recepción de un nuevo dato por parte del usuario.

En caso de que el usuario envíe una "e" y el sistema la reciba de forma

satisfactoria el módulo del amperímetro inmediatamente es habilitado.

Al seleccionarse este módulo, inmediatamente se deshabilitan las interrupciones, para evitar que una nueva interrupción impida terminar el procedimiento que se está ejecutando.

En la sección 5.2.1.4.2 se mencionó la conexión del canal 1 del ADC con la salida del sensor de corriente. Es por ello que el siguiente paso después de la deshabilitación de las interrupciones, es la selección del canal 1 del ADC integrado en el microcontrolador.

De igual forma, y como prevención para evitar la confusión del usuario, la pantalla del ADC se limpia.

Una vez finalizado el método de limpieza del LCD, el ADC realiza una lectura de la tensión aplicada en el pin del canal 1, y con base en esta genera un valor representativo, el cual posteriormente es operado para realizar el cálculo del valor de corriente.

El cálculo del valor de corriente se encuentra basado en la ecuación:

$$V = 0.2 * I + 2.5 \quad \text{EC(5.3)}$$

La ecuación 5.3 es la ecuación de la recta que describe el funcionamiento del sensor de corriente con respecto a su tensión de salida, al variarse la corriente que pasa por este.

Producto de esta ecuación se obtuvo la ecuación 5.4 sobre la cual se generó el código para esta sección:

$$\frac{(V * 1000 - 2500) * 10}{2} = I \quad \text{(5.4)}$$

Se debe mencionar que para simplificar el procedimiento de cálculo de los valores de corriente, en la ecuación 5.4 los datos son multiplicados por múltiplos de 10.

Seguidamente y de igual forma que para el caso del voltímetro, el sistema debe separar el dato en cada uno de los dígitos que lo conforman, con la intención de convertirlo en Ascii y desplegarlo.

Finalmente, una vez que el dato ha sido construido, éste es enviado al usuario y desplegado en pantalla.

Para cada medición, es necesario que el usuario envíe un carácter. En caso de que se requieran realizar muchas mediciones, y tomando en cuenta el funcionamiento del sistema, se puede realizar un pequeño programa desde línea de comando llamado script, en caso de que se use el sistema operativo Linux, en donde se automatice el envío del carácter y la frecuencia con la que se envíe, facilitando de esta forma la variación del período de muestreo tanto de la corriente como del voltaje que se quieren medir.

En la sección de anexos se incluye tanto el código fuente del software del microcontrolador, como el código fuente del programa hecho desde línea de comando para automatizar las mediciones de corriente y voltaje.

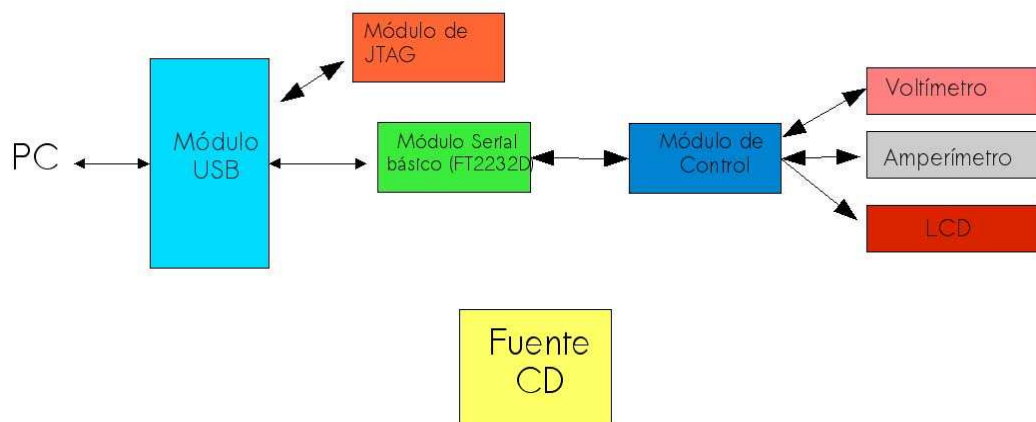
## CAPÍTULO 6. Análisis de Resultados

### 6.1 Resultados Obtenidos

La presente sección reúne los datos obtenidos mediante el empleo del sistema desarrollado con el fin de evaluar su funcionalidad.

Antes de avanzar en el presente capítulo, es necesario mencionar que el sistema que se implementó, se le hicieron algunas modificaciones con respecto al sistema originalmente diseñado, claro está que la modificación en la implementación, no se aparta de los objetivos propuestos en el capítulo 2.

El diagrama de bloques del sistema implementado se puede observar en la figura 6.1.



**Figura 6.1 Diagrama de bloques del sistema implementado**

La diferencia entre el sistema propuesto en la figura 5.1 y el mostrado en la figura 6.1 se debe a que el objetivo de RidgeRun es realizar un circuito integrado prototipo de el sistema diseñado, por lo que el sistema implementado únicamente le es útil a la empresa para demostrar la funcionalidad de algunos

módulos y validar el presente proyecto.

### **6.1.1 Módulo USB y Módulo Serial**

La implementación del módulo USB se dio únicamente para el circuito de interfaz con el computador. Tomando en cuenta el único tamaño disponible del circuito integrado FT2232D (LQFP -48 pines) y el tiempo que tomaría cablear el circuito dado su pequeño tamaño, se optó por emplear el dispositivo DLP2232M, el cual es un pequeño circuito impreso para desarrollar prototipos a nivel comercial, que contiene el circuito integrado FT2232D implementado de la forma que sugiere el fabricante para que sea funcional. De esta forma, el emplear la versión para desarrollo de prototipos comercial, no representa diferencia alguna en el funcionamiento con respecto al circuito de utilizado en el diseño de la figura 5.4 . En la figura 6.2 se puede observar el módulo DLP2232D

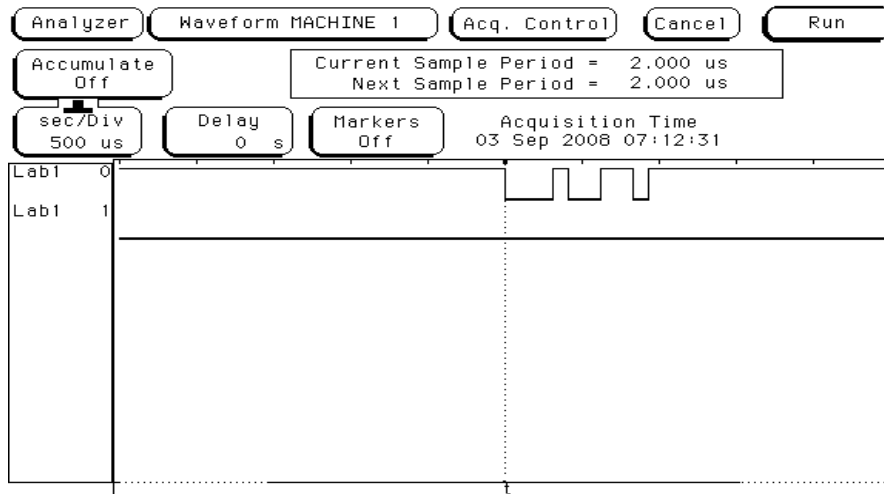


**Figura 6.2 Dispositivo para prototipado DLP2232M**

En la figura 6.2 se demuestra el funcionamiento de éste módulo en conjunto con el módulo Serial. Dado el caso que el módulo serial está constituido también por un circuito integrado FT2232D, el módulo USB y el módulo Serial pueden verse como un sólo bloque para el caso del circuito implementado, es decir, la entrada y salida de datos en un extremo del dispositivo DLP2232M sería por medio del protocolo USB, mientras que la entrada y salida de datos en el otro extremo serían los mismos datos, pero de forma serial .



La figura 6.3 muestra como dato un carácter ASCII (una letra “d” correspondiente a un 64 hexadecimal), es enviado por el puerto USB y serializado. De esta manera se comprueba el correcto funcionamiento tanto del módulo USB y el funcionamiento básico del módulo Serial.



**Figura 6.3 Dato enviado por el puerto USB después de ser serializado**

Con respecto al circuito de Concentrador USB (HUB USB) que también forma parte del módulo USB, este a petición de la empresa no fue implementado, dado que se consideró que su presencia no resultaba ser indispensable. En otras palabras, la ausencia de este concentrador no retrasaba el avance del proyecto dado que los datos obtenidos no serían afectados.

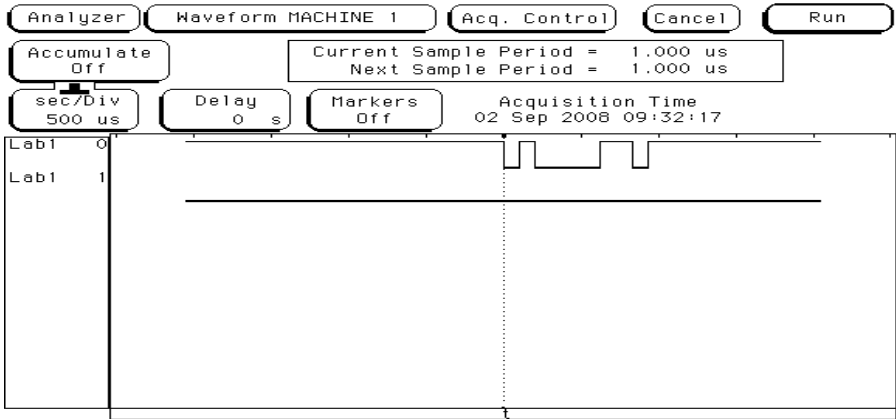
### **6.1.2 Módulo de Control**

Para evaluar la funcionalidad del módulo de Control, ésta se debe hacer en forma conjunta con cada uno de los submódulos que dependen de éste módulo.

#### **6.1.2.1 Módulo de Control y submódulo de Voltímetro**

La figura 6.4 muestra un comando enviado al módulo de control a través del módulo USB y módulo Serial, a los cuales previamente se les evaluó su

funcionalidad en la sección anterior. El comando enviado fue el código ASCII equivalente a la letra “a” que hace que el voltímetro dependiente del módulo de Control quede habilitado.



**Figura 6.4 Dato ASCII enviado por el puerto USB correspondiente a una letra “a”**

Para realizar una prueba de funcionamiento, y obtener resultados experimentales con el voltímetro diseñado, se envió en varias ocasiones el comando “a” por medio del puerto USB y se varió la tensión aplicada al circuito acondicionador de señal del voltímetro.

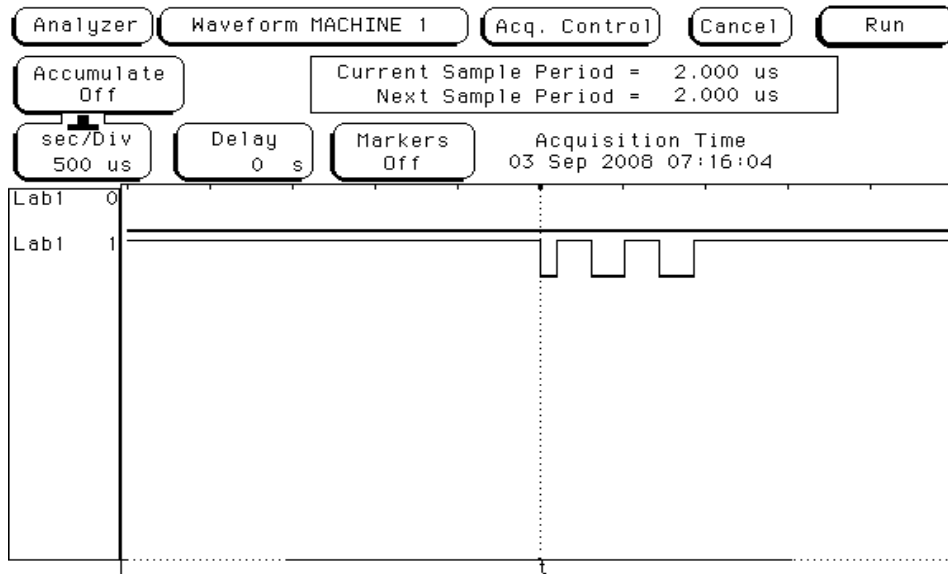
Las mediciones obtenidas por el sistema desarrollado, fueron comparadas con lecturas realizadas por un multímetro de origen comercial marca Radio Shack, modelo 22-811. Los resultados obtenidos se pueden observar en la tabla 6.1

**Tabla 6.1 Tabla comparativa de valores de tensión obtenidos al emplear el voltímetro diseñado y un multímetro de origen comercial Radio Shack modelo 22-811**

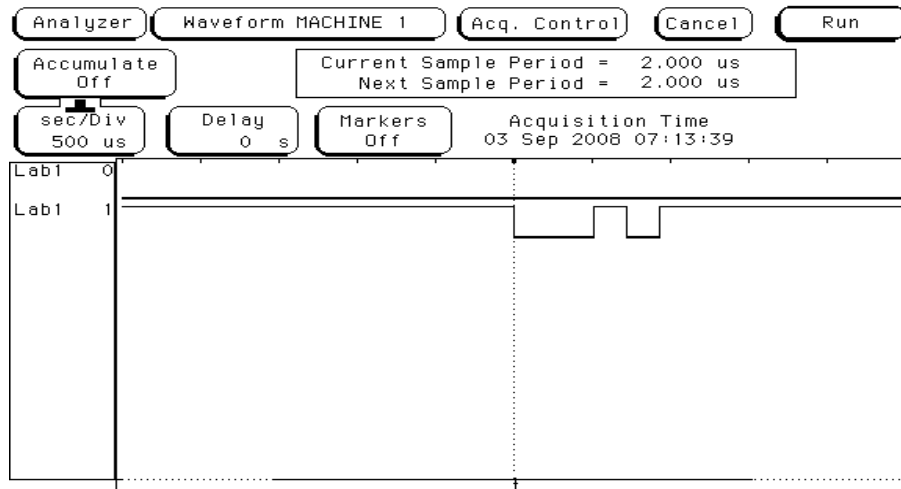
Medición	Tensión medida con el voltímetro diseñado	Tensión medida con el multímetro comercial	Diferencia %
1	0.304	0.303	0.33
2	0.568	0.559	1.61
3	0.645	0.638	1,09
4	1.002	0.990	1.21
5	1.901	1.891	0.52
6	2.314	2.303	0.48
7	3.052	3.040	0.39
8	5.150	5.139	0.21
9	8.680	8.674	0.06
10	10.310	10.301	0.08
11	12.640	12.637	0.26
13	13.800	13.740	0.44
14	14.730	14.722	0.05
15	15.010	14.922	0.58

Las figuras 6.5,6.6,6.7,6.8,6.9 y 6.10 muestran los diferentes dígitos enviados por el módulo de control hacia el computador mediante el módulo Serie y el módulo USB generados a partir de lecturas realizadas por el voltímetro.. Se debe recordar que el software del microcontrolador, para que el formato de los datos sea compatible con Minicom, descompone el resultado y lo separa en dígitos ASCII para transmitirlos uno a uno.

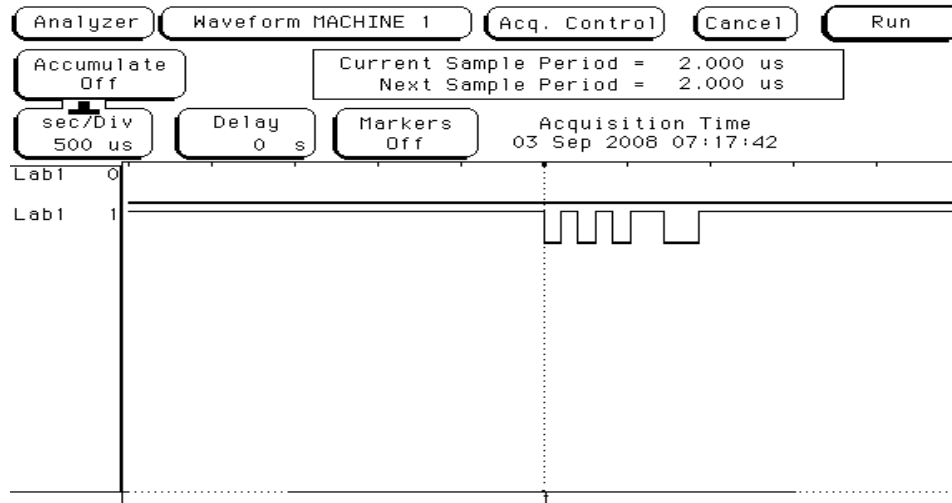
En este caso las figuras 6.5, 6.6, 6.7, 6.8 y 6.9 corresponden al valor medido y anotado en la séptima posición de la tabla 6.1: 3052mV.



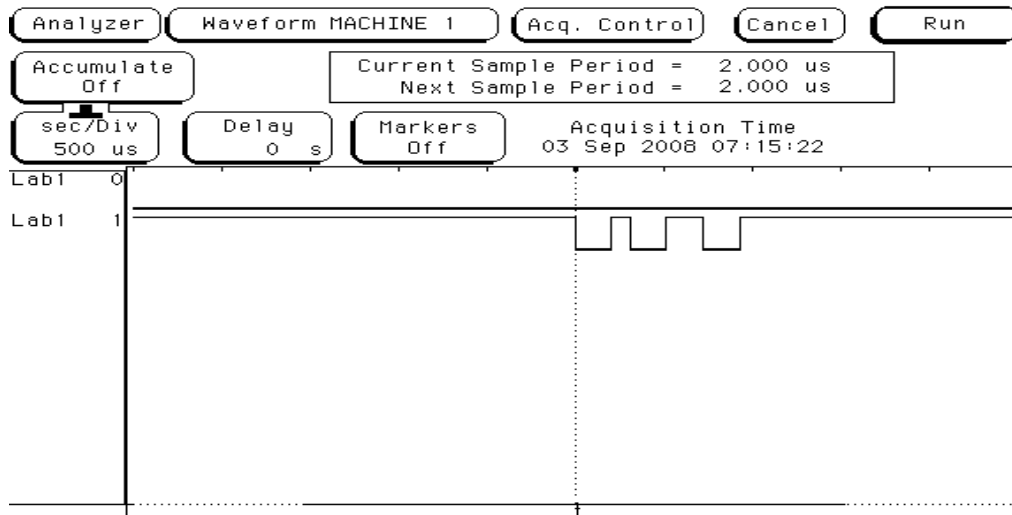
**Figura 6.5** Dato correspondiente a la primera posición del valor de tensión medida (3 en ASCII)



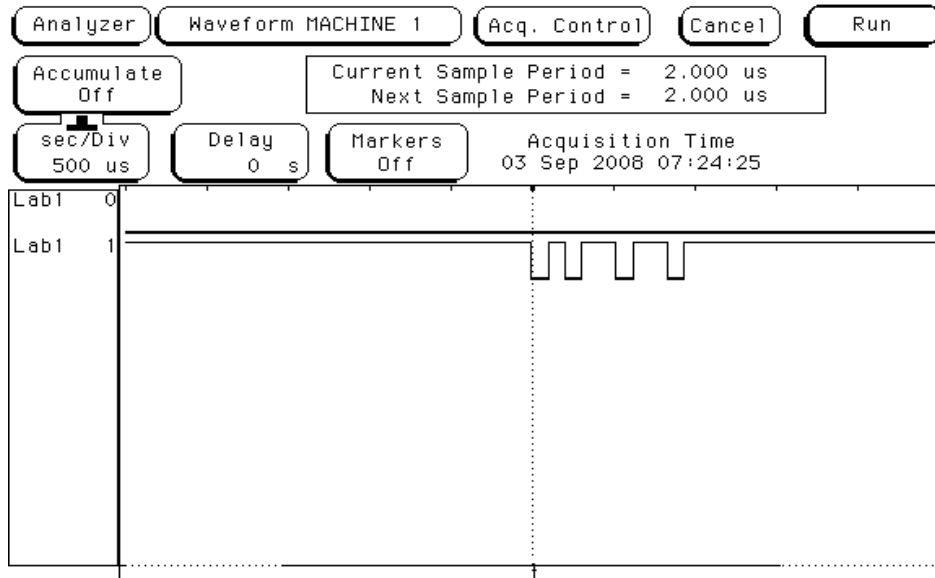
**Figura 6.6** Dato correspondiente a la segunda posición del valor de tensión media (0 en ASCII)



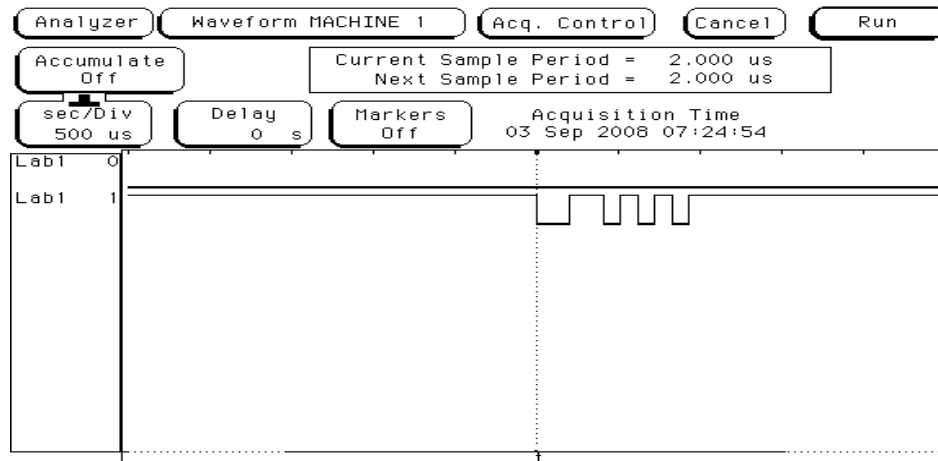
**Figura 6.7 Dato correspondiente a la tercera posición del valor de tensión medida (5 en ASCII)**



**Figura 6.8 Dato correspondiente a la cuarta posición del valor de tensión medida (2 en ASCII)**

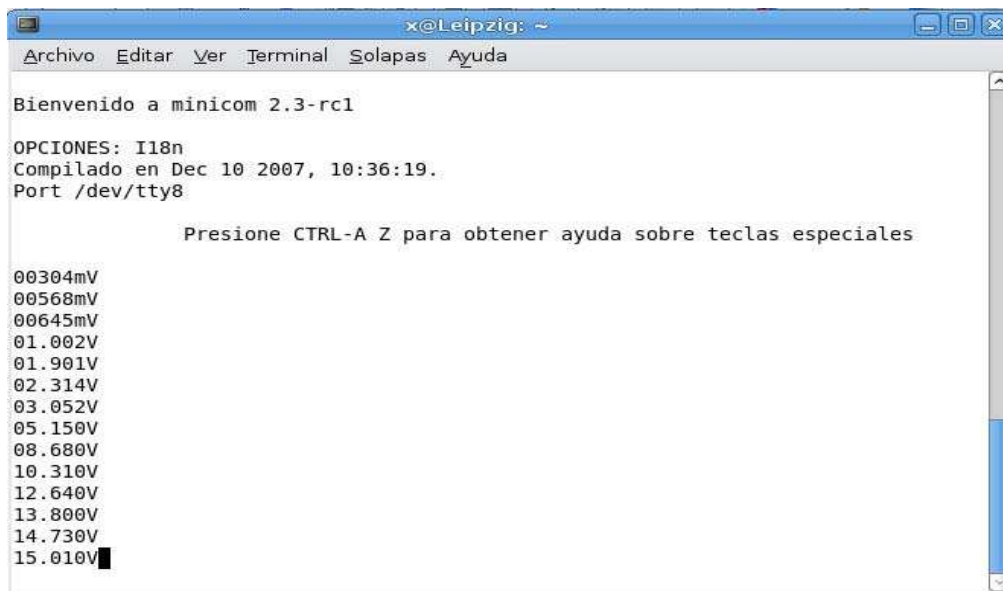


**Figura 6.9** Dato correspondiente al carácter “m” en ASCII para indicar la escala en que está dado el dato



**Figura 6.10** Dato correspondiente al carácter “V” en ASCII para indicar las unidades del dato dado

En la siguiente figura se muestran los datos de la tabla 6.1 que fueron enviados al computador



The image shows a terminal window titled 'x@Leipzig: ~'. The window contains the following text:

```
Archivo Editar Ver Terminal Solapas Ayuda

Bienvenido a minicom 2.3-rc1

OPCIONES: I18n
Compilado en Dec 10 2007, 10:36:19.
Port /dev/tty8

Presione CTRL-A Z para obtener ayuda sobre teclas especiales

00304mV
00568mV
00645mV
01.002V
01.901V
02.314V
03.052V
05.150V
08.680V
10.310V
12.640V
13.800V
14.730V
15.010V
```

**Figura 6.11 Datos de las distintas mediciones descritos en la tabla 6.1 enviados al PC**

### 6.1.2.2 Módulo de Control y submódulo de LCD

Los resultados experimentales del funcionamiento del conjunto de módulos Control-LCD son los más sencillos de obtener y evaluar. Para ello, únicamente basta con mirar la figura 6.12 donde se muestra la imagen del LCD desplegando un dato, que en este caso es el dato referente a la medición #6 de la tabla 6.1



**Figura 6.12** Despliegue en pantalla de una tensión medida con el voltímetro

### **6.1.2.3 Módulo de Control y submódulo de Amperímetro**

El submódulo de Amperímetro estuvo basado en la utilización del sensor para medición de corrientes ACS712ELCTR-20A-T de Allegro Microsystems. Dado su pequeño tamaño (4.9mm x 6mm x 1.75mm,SOIC-8), fue necesario soldarlo sobre una base SIP (Single In-line Package) para colocarlo sobre un protoboard, de manera que su empleo fuese mucho más sencillo. Una vez acoplado el sensor, se procedió a tomar mediciones de la tensión de salida del mismo y compararlas con los rangos establecidos por el fabricante para determinar el correcto funcionamiento del sensor

En la tabla 6.2 se muestran los valores de tensión de salida del sensor, tanto teóricos o esperados como experimentales para diferentes valores de corriente.



**Tabla 6.2: Tensión de salida del sensor ACS712ELCTR-20A-T para diferentes valores de corriente.**

Corriente (A)	Tensión Esperada (V)	Tensión Obtenida (V)	Error (%)
0	2.5	0.822	67.12
0.01	2.502	0.826	66.99
0.06	2.512	0.840	66.56
0.22	2.544	0.888	65.09
0.35	2.570	0.956	62.80
0.48	2.596	0.968	62.71
0.64	2.628	1.016	61.34
0.80	2.660	1.064	60.00
0.93	2.686	1.103	58.94
1.12	2.724	1.151	57.75
1.28	2.756	1.200	56.46
1.50	2.800	1.248	55.43
1.75	2.850	1.295	54.56
1.99	2.898	1.361	53.04
2.02	2.904	2.376	18.18

Después de obtener los datos que conforman la tabla 6.2, se procedió a verificar toda conexión del sensor de acuerdo a como lo especifica el fabricante y realizar una nueva medición dado que el comportamiento del sensor no era el esperado.

Para que el funcionamiento del sensor fuese considerado como correcto, los valores de tensión de salida proporcionales al valor de corriente deberían empezar en 2.5V al medirse cero amperios, y a partir de éste valor, aumentar en 185mV por cada aumento de 1 amperio.

En la tabla 6.3 se muestran los datos obtenidos correspondientes al valor de tensión de salida del sensor ACS712ELCTR-20A-T esperados y obtenidos,

para los mismos valores de corriente empleados en la tabla 6.2, después de verificar las conexiones indicadas por el fabricante en la hoja de datos

**Tabla 6.3: Tensión de salida del sensor ACS712ELCTR-20A-T para diferentes valores de corriente después de verificar las conexiones del sensor de corriente**

Corriente (A)	Tensión Esperada (V)	Tensión Obtenida (V)	Error (%)
0	2.5	0.829	66.84
0.01	2.502	0.840	66.43
0.06	2.512	0.843	66.44
0.22	2.544	0.852	66.51
0.35	2.570	0.956	62.80
0.48	2.596	0.964	62.87
0.64	2.628	0.980	62.71
0.80	2.660	0.998	62.48
0.93	2.686	1.050	60.91
1.12	2.724	1.114	59.10
1.28	2.756	1.190	56.82
1.50	2.800	1.207	56.89
1.75	2.850	1.438	49.54
1.99	2.898	1.776	38.72
2.02	2.904	2.200	24.24

La relación entre los valores obtenidos en la tabla 6.2 y en la tabla 6.3 muestra la situación permanente con la que siempre se contó en cada medición con respecto a las anteriores: los valores de tensión obtenidos en cada grupo de mediciones nunca fueron similares para los mismos valores de corriente sensados.

Al no contar con un sensor extra, y por tratarse de la implementación prototipo, la empresa sugirió descartar el uso del sensor temporalmente y únicamente implementar la medición de tensiones y su posterior conversión a valores de

corriente, partiendo del comportamiento del sensor indicado por el fabricante en la hoja de datos.

Con base en el párrafo anterior, se procedió a asumir un funcionamiento correcto del sensor y se tomaron datos con el fin de corroborar el buen funcionamiento de la parte del módulo de Amperímetro ligada con la conversión de tensiones.

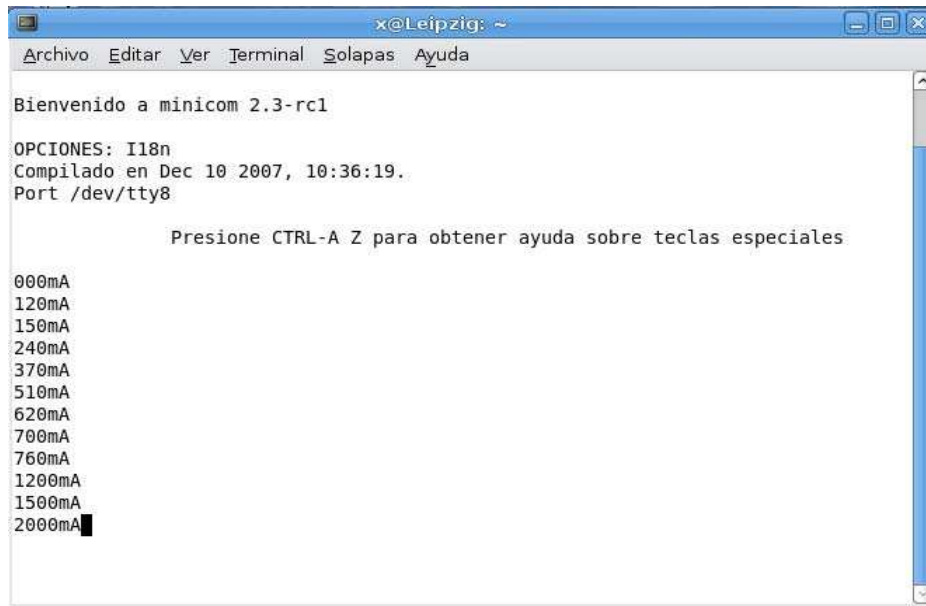
Para esta prueba se aplicó tensión a la terminal AMPMETER del microcontrolador donde normalmente se conectaría la tensión de salida del sensor de corriente y se varió en el rango de funcionamiento del sensor que se tenía previsto : 2.5V a 2.9V de acuerdo a los posibles valores de corriente que se podrían medir.

En la siguiente tabla se muestra el valor de tensión aplicado a la terminal de AMPMETER junto con el valor de corriente teórico, es decir que origina dicha tensión de salida en el sensor y por último el valor de corriente medido con el módulo (valor experimental).

**Tabla 6.4: Corriente registrada para distintos valores de tensión aplicada a la terminal AMPMETER**

<b>Tensión Aplicada (V)</b>	<b>Valor Teórico de Corriente (A)</b>	<b>Valor Experimental de Corriente (mA)</b>
2.500	0.0	0000
2.525	0.13	120
2.530	0.15	150
2.550	0.25	240
2.575	0.38	370
2.600	0.50	510
2.625	0.63	620
2.640	0.70	700
2.650	0.75	760
2.750	1.25	1200
2.850	1.75	1500
2.900	2	2000

En la siguiente figura se muestra la captura de pantalla de la aplicación Minicom, en donde se muestran los valores de corriente originados al aplicar diversas tensiones a la terminal AMPMETER de forma que se simulara el correcto funcionamiento del módulo de sensado de corriente.



```
x@Leipzig: ~
Archivo Editar Ver Terminal Solapas Ayuda

Bienvenido a minicom 2.3-rc1

OPCIONES: I18n
Compilado en Dec 10 2007, 10:36:19.
Port /dev/tty8

Presione CTRL-A Z para obtener ayuda sobre teclas especiales

000mA
120mA
150mA
240mA
370mA
510mA
620mA
700mA
760mA
1200mA
1500mA
2000mA
```

**Figura 6.13 Valores de corriente sensados con el módulo de amperímetro**

### **6.1.3 Módulo de JTAG**

Para obtener los resultados experimentales concluyentes sobre el funcionamiento de éste módulo, se actualizó el firmware de uno de los dispositivos que actualmente desarrolla RidgeRun, con una versión más actual e igualmente desarrollada por la empresa. Cabe mencionar que los pasos que se siguieron para realizar esta función, fueron los mismos que se siguen normalmente para realizar una actualización de firmware mediante el empleo de un dispositivo JTAG Olimex de origen comercial.

A continuación se describen los pasos requeridos para la actualización de firmware en uno de los dispositivos en desarrollo empleando el sistema operativo Linux y el programa OpenOCD, con el fin de que el lector comprenda el procedimiento y los resultados obtenidos.

## Pasos:

1. Abrir una terminal
2. Escribir en la terminal: “sudo openocd -f /etc/ft2232.cfg”

La instrucción *sudo* permite ejecutar el comando que acompaña en modo super usuario, es decir con permisos de ejecución. El comando `openocd -f /etc/ft2232.cfg` permite ejecutar el programa openOCD y tener como referencia el archivo de configuración `ft2232.cfg` ubicado en el directorio `/dev` , donde se describen los parámetros de configuración de la herramienta

3. Escribir en una segunda terminal: telnet “localhost 4444”
4. Cuando estos pasos han sido ejecutados y la herramienta para JTAG es funcional, en la segunda terminal abierta aparecerá la siguiente información

Open On-Chip Debugger

Lo cual significa que la herramienta de JTAG es funcional.

5. Seguidamente en una tercera terminal se escribe el comando: “rrocd --file bootloader”. “bootloader” es un archivo que contiene la lista de instrucciones que deben incluirse en la terminal donde la aplicación telnet se está ejecutando. A manera de ejemplo se muestra el siguiente conjunto de instrucciones obtenidas al emplear el comando mencionado:

6. `rrocd --file bootloader .`

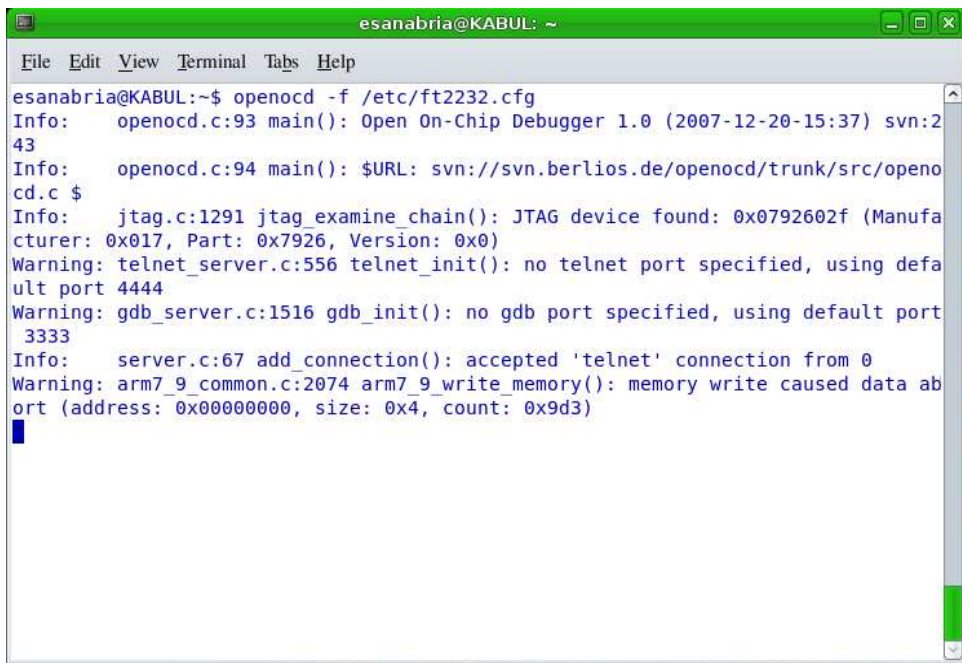
halt

**sleep 500**

```
wait_halt
sleep 500
load_image /home/user/manufacturing/liteon/images/vz5-liteon/setup 0x0 elf
resume 0x00002004
sleep 1000
halt
sleep 500
wait_halt
sleep 500
```

7. Las instrucciones generadas al ejecutar el comando en el paso #5 son introducidas en la terminal #2 donde se está ejecutando la aplicación telnet y finalmente después de unos segundos el nuevo firmware habrá sido descargado en el dispositivo en desarrollo.

En las figuras 6.14, 6.15 y 6.16 se muestran las capturas de pantalla para cada una de las terminales que se emplean en el proceso de descarga del firmware a uno de los dispositivos en desarrollo.



```
esanaabria@KABUL: ~
File Edit View Terminal Tabs Help
esanaabria@KABUL:~$ openocd -f /etc/ft2232.cfg
Info: openocd.c:93 main(): Open On-Chip Debugger 1.0 (2007-12-20-15:37) svn:2
43
Info: openocd.c:94 main(): $URL: svn://svn.berlios.de/openocd/trunk/src/openo
cd.c $
Info: jtag.c:1291 jtag_examine_chain(): JTAG device found: 0x0792602f (Manufa
cturer: 0x017, Part: 0x7926, Version: 0x0)
Warning: telnet_server.c:556 telnet_init(): no telnet port specified, using defa
ult port 4444
Warning: gdb_server.c:1516 gdb_init(): no gdb port specified, using default port
3333
Info: server.c:67 add_connection(): accepted 'telnet' connection from 0
Warning: arm7_9_common.c:2074 arm7_9_write_memory(): memory write caused data ab
ort (address: 0x00000000, size: 0x4, count: 0x9d3)
```

**Figura 6.14** Captura en pantalla de la primera terminal empleada

La figura 6.14 muestra la captura de pantalla de la primera terminal empleada para llevar a cabo el procedimiento de descarga de firmware a uno de los dispositivos en desarrollo. En esta figura claramente se observa que el dispositivo JTAG , perteneciente al módulo del mismo nombre, es detectado. En esta imagen se resumen los pasos 1 y 2 previamente descritos.



```
esanabria@KABUL: ~/Desktop/vc4-mrs.5412-3183/images
File Edit View Terminal Tabs Help
esanabria@KABUL:~/Desktop/vc4-mrs.5412-3183/images$ rrocd --file bootloader .

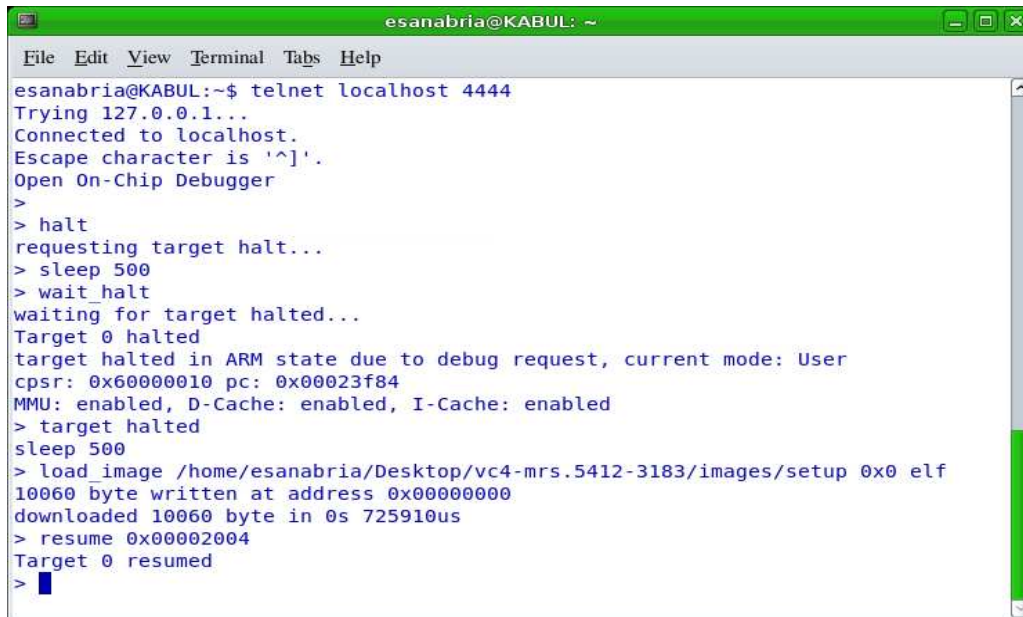
halt
sleep 500
wait halt
sleep 500
load_image /home/esanabria/Desktop/vc4-mrs.5412-3183/images/setup 0x0 elf
resume 0x00002004
sleep 1000
halt
sleep 500
wait halt
sleep 500

load_image /home/esanabria/Desktop/vc4-mrs.5412-3183/images/bootloader 0x0 elf
resume 0x02d0bc64
esanabria@KABUL:~/Desktop/vc4-mrs.5412-3183/images$
```

**Figura 6.15 Captura de pantalla de la segunda terminal empleada**

En la figura 6.15 se muestran las instrucciones que se obtienen posterior a ejecutar el comando “rrocd –file bootloader .”, las cuales serán introducidas en la terminal donde se tiene la aplicación de telnet abierta.





```
esanabria@KABUL: ~
File Edit View Terminal Tabs Help
esanabria@KABUL:~$ telnet localhost 4444
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Open On-Chip Debugger
>
> halt
requesting target halt...
> sleep 500
> wait_halt
waiting for target halted...
Target 0 halted
target halted in ARM state due to debug request, current mode: User
cpsr: 0x60000010 pc: 0x00023f84
MMU: enabled, D-Cache: enabled, I-Cache: enabled
> target halted
sleep 500
> load_image /home/esanabria/Desktop/vc4-mrs.5412-3183/images/setup 0x0 elf
10060 byte written at address 0x00000000
downloaded 10060 byte in 0s 725910us
> resume 0x00002004
Target 0 resumed
>
```

**Figura 6.16 Captura de pantalla de la tercera terminal empleada**

La figura 6.15 muestra la captura de la tercera terminal donde se ejecuta la aplicación telnet. Como se puede observar, una vez que aparece el texto : “Open On Chip Debugger”, las instrucciones generadas mediante el comando “rrocd - - file bootloader” (figura 6.15), son ingresadas en esta terminal por el usuario.

La finalización del proceso de descarga del firmware, se da cuando en la ventana terminal aparece el texto : “Target 0 resumed”, el cual se puede visualizar en la imagen 6.16.

#### **6.1.4 Módulo de Fuente CD**

La implementación del circuito prototipo de este módulo, no incluyó el decodificador 74HC238, dado que no fue posible adquirirlo. Sin embargo para la prueba de este módulo, cada una de las diferentes combinaciones responsables de seleccionar una tensión de salida predefinida fue seleccionada manualmente, mediante la polarización de la base de cada transistor ligado con una salida específica. De esta forma, las mediciones realizadas, se hicieron en primera instancia sin carga, y seguidamente se realizaron para varios valores de carga,

con tal de verificar el comportamiento de este módulo y asegurar su correcto funcionamiento para múltiples usos. Los resultados obtenidos se muestran en las siguientes tablas

**Tabla 6.5: Tensiones predefinidas de salida del módulo Fuente CD sin carga**

Valor teórico (V)	Valor Experimental (V)
1.5	1.504
1.8	1.792
2.5	2.515
3.0	3.067
3.3	3.304
3.6	3.630
5	5.010
[1.5-5.0]	[1.504-5.010]

**Tabla 6.6: Tensiones predefinidas de salida del módulo Fuente CD con carga RL=2.2KΩ**

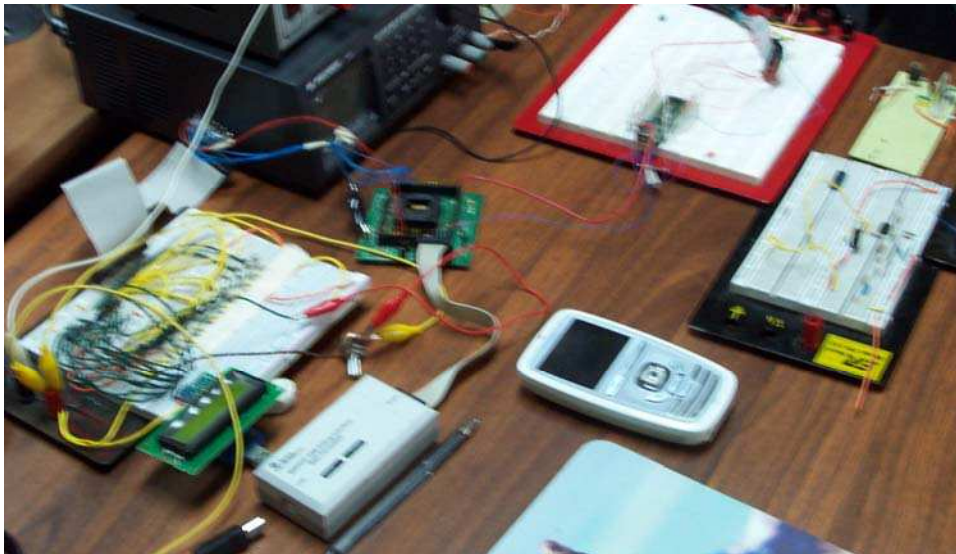
Valor teórico (V)	Valor Experimental (V)
1.5	1.504
1.8	1.792
2.5	2.515
3.0	3.067
3.3	3.305
3.6	3.360
5.0	4.99
[1.5-5.0]	[1.504-4.99]

**Tabla 6.7: Tensiones predefinidas de salida del módulo Fuente CD con carga  $R_L=220\Omega$**

Valor teórico (V)	Valor Experimental (V)
1.5	1.504
1.8	1.192
2.5	2.514
3.0	3.067
3.3	3.305
3.6	3.363
5	4.98
[1.5-5.0]	[1.504-4.98]

### 6.1.5 Prototipo Implementado

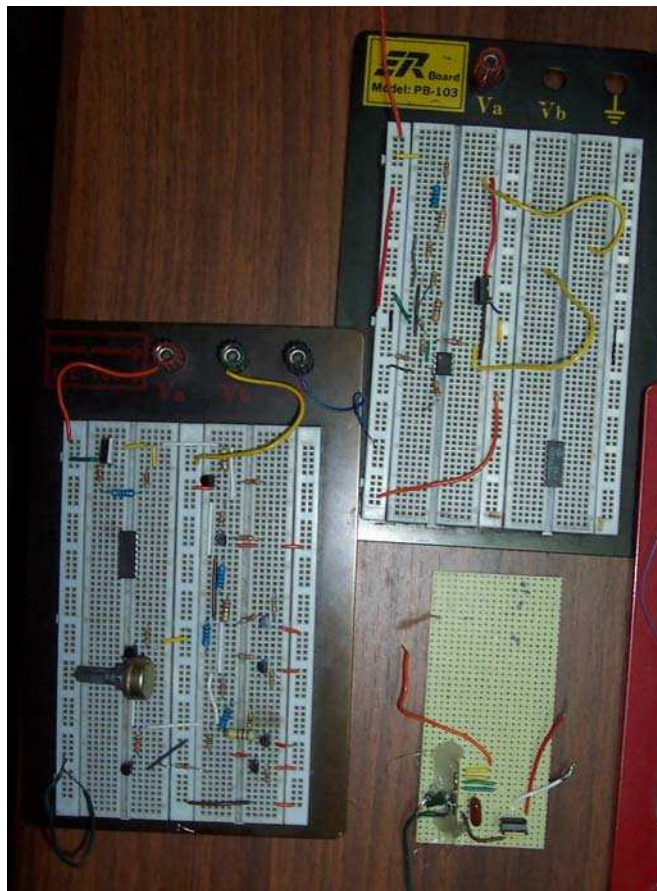
En la presente sección se muestran algunas imágenes del prototipo implementado.



**Figura 6.17 Sistema prototipo implementado**

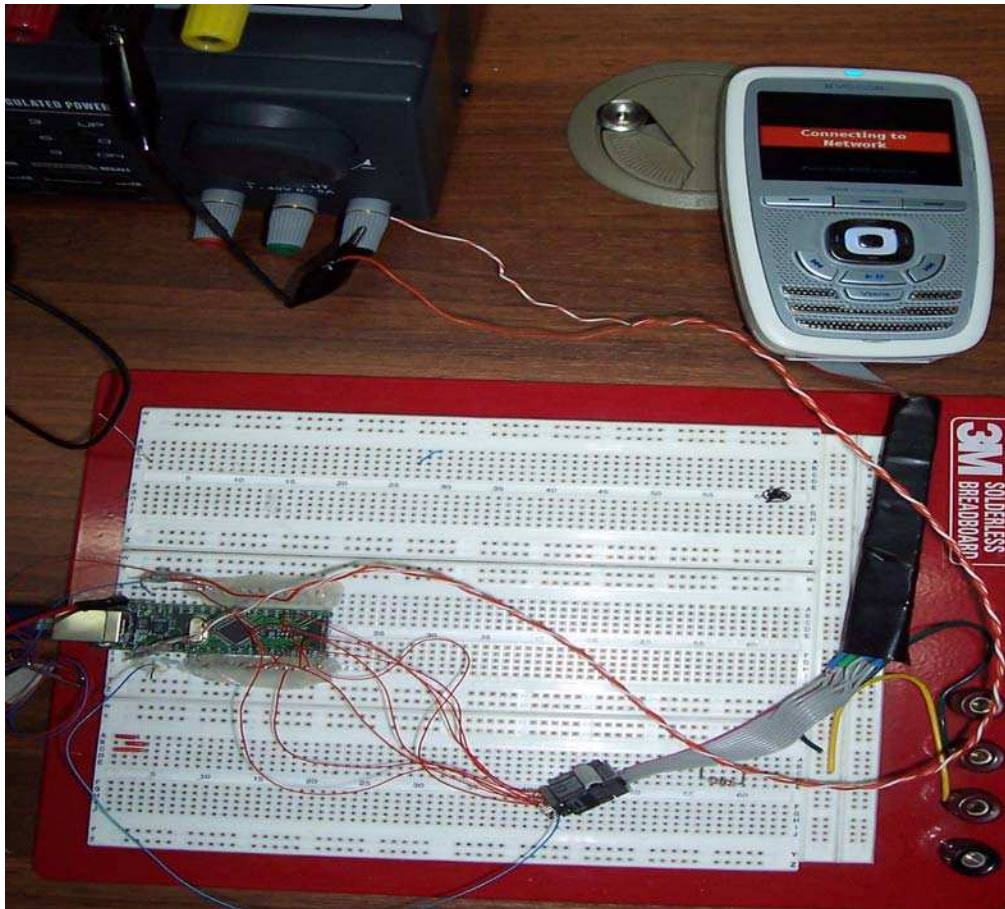
En la figura 6.17 se muestra el sistema prototipo implementado. El dispositivo que

se observa en el centro de la imagen corresponde al sistema en desarrollo con el cual se probó el sistema prototipo.



**Figura 6.18 Prototipos de fuente de CD, Amperímetro y Voltímetro**

En la figura 6.18 se observa en detalle los prototipos implementados correspondientes al módulo de fuente CD, el módulo del Amperímetro y el módulo del Voltímetro.



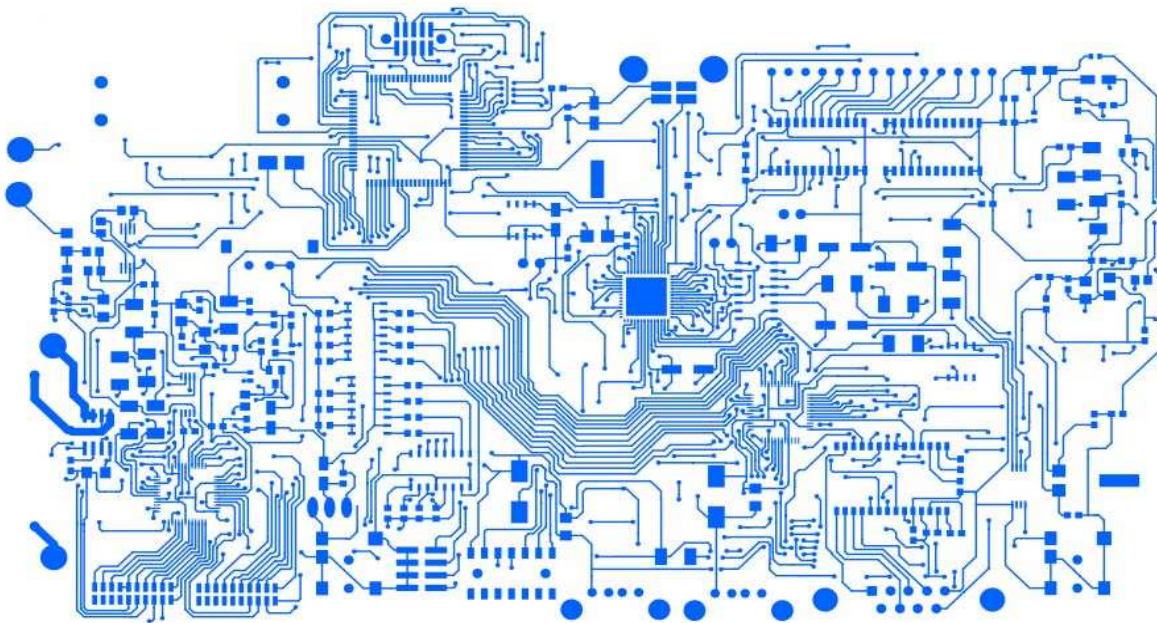
**Figura 6.19 Módulo de JTAG conectado a un dispositivo en desarrollo**

En la figura 6.19 se puede observar con detalle el módulo de Jtag siendo empleado con uno de los dispositivos en desarrollo en RidgeRun para actualizar el firmware del mismo.

### 6.1.6 Circuito Impreso

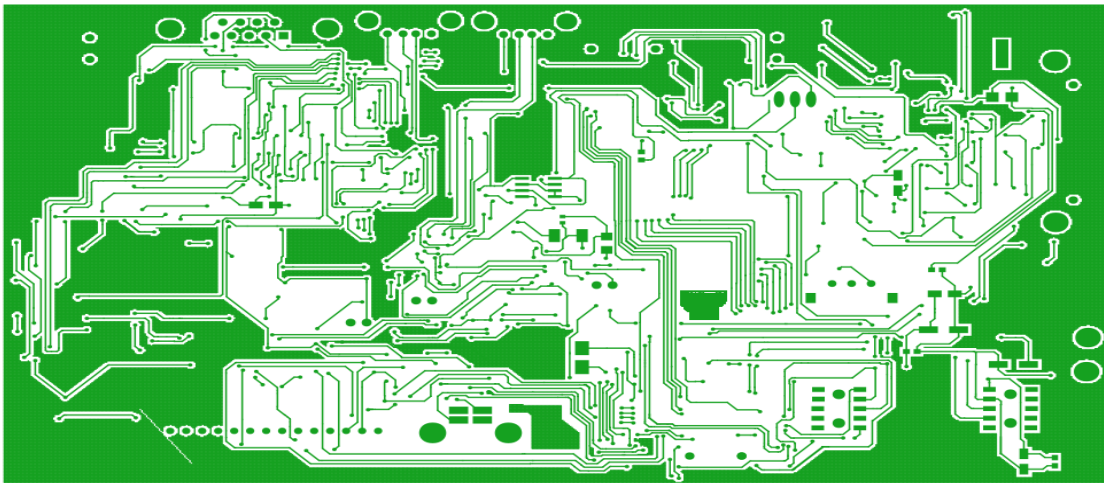
Uno de los objetivos específicos mencionados en el capítulo 2 consistía en desarrollar el esquemático para que posteriormente la empresa se encargara de realizar un circuito impreso basándose en el mismo, sin embargo el diseño del circuito impreso fue realizado durante el desarrollo del presente proyecto empleando el software libre Kicad. y se muestra en figuras.

**La figura 6.20 muestra el diseño del circuito impreso correspondiente a la capa donde se ubican los componentes.**



**Figura 6.20 Diseño del circuito impreso correspondiente a la capa de componentes**

Las pistas de mayor grosor que se aprecian a la izquierda de la figura, se hicieron intencionalmente de ese tamaño, dado que es en esa zona del impreso donde se ubicará el módulo de amperímetro, por lo que existe el riesgo de que corrientes de hasta 2 amperios pasen por allí. Al dejar las pistas mas gruesas, la resistencia eléctrica de la misma disminuye y por ende baja la probabilidad de que genere calor o cualquier problema eléctrico.



**Figura 6.21 Diseño del circuito impreso correspondiente a la capa de cobre**

En la figura 6.21 se observa el diseño del circuito impreso correspondiente a la capa de cobre.

En la siguiente tabla se muestran los tamaños del circuito impreso diseñado, el cual será el tamaño final que ocupará en el escritorio de un desarrollador.

**Tabla 6.8: Dimensiones finales del circuito impreso diseñado**

Medida	Valor
Largo	18cm
Ancho	10 cm
Area	180cm <sup>2</sup>

Para la realización del diseño del circuito impreso, se tomaron en cuenta factores muy influyentes como: encapsulados de circuitos integrados, tipos de conectores por emplear, precios, distribuidores de las piezas requeridas, distribución adecuada de todos los componentes para disminuir el tamaño del circuito impreso, tamaño de pistas, etc.

Dado que la herramienta desarrollada, en primera instancia se usará para suplir las necesidades internas de RidgeRun, y posteriormente será llevada al mercado comercial, en donde debe ser producida de forma industrial, se realizó una cotización de los costos para producir dicha herramienta a nivel comercial.

El costo total de producción de la misma es de \$66.401 Los detalles de la cotización se pueden consultar en la sección de anexos.

## **6.2 Análisis de Resultados**

### **6.2.1 Módulo USB y Módulo Serial**

Basándose en las figuras 6.3 y 6.4 donde se muestran los datos serializados que han sido enviados por el usuario a través del puerto USB, así como también en la figura 6.11 en donde se muestran los datos recibidos por el computador vía el puerto USB después de ser convertidos desde el protocolo serial, se puede concluir que no hay errores de funcionamiento asociados con este módulo, puesto que de ser así, los datos no serían recibidos o bien no serían desplegados correctamente. En otras palabras el funcionamiento del módulo es el esperado.

### **6.2.2 Módulo de Control**

De igual forma como se indicó en la sección de resultados experimentales, el funcionamiento del módulo de Control debe ser evaluado teniendo como base el funcionamiento de los módulos que se encuentran asociados.

Sin embargo, antes de continuar con la evaluación de los restantes módulos ligados al módulo de Control, se debe mencionar que para que el código



desarrollado en el lenguaje de programación C fuese funcional, y pudiese ser empleado por el microcontrolador que integra este módulo, fue necesario incluir segmentos de código en el lenguaje de programación Ensamblador para MSP430, con el fin de solventar problemas de manejo de pila que a su vez repercutían en la correcta ejecución de los distintos procedimientos que conformaban el programa en general

Mediante el empleo del lenguaje Ensamblador, este problema fue solucionado de forma satisfactoria, haciendo posible que el programa en ejecución siempre pudiese tener la referencia de la siguiente instrucción por ejecutar.

#### **6.2.2.1 Módulo de Control y submódulo de Voltímetro**

Después de enviar en reiteradas ocasiones el comando “a” “d” o , para leer las diferentes tensiones aplicadas al circuito acondicionador de señal del módulo de Voltímetro, se obtuvieron los resultados de la tabla 6.1. Como se puede observar en dicha tabla, los datos contenidos en la columna de la derecha corresponden a los datos adquiridos mediante el empleo del voltímetro desarrollado, mientras que los datos de la izquierda corresponden a datos adquiridos por un multímetro de origen comercial. El dato correspondiente a la comparación mediante porcentaje de error se encuentra en la misma tabla. Para esta comparación, se tomaron como teóricos los datos medidos con el multímetro comercial, y como experimentales, los datos adquiridos por el sistema desarrollado.

Como se podrá observar, la diferencia porcentual entre ambos tipos de datos, no supera el 1.61%. Ahora bien, tomando en cuenta que se emplea un convertidor analógico digital de 12 bits para realizar las mediciones de tensión, la incertidumbre del dispositivo es de 0.0008 V que redondeado se puede tomar como 0.001V, razón por la cual se le atribuye un alto grado de confiabilidad al sistema. Las diferencias porcentuales entre los datos de la tabla 6.1 se deben básicamente a que no se emplearon resistores con valores precisos en la implementación.

La empresa considera que la diferencia porcentual entre ambos datos no constituye un problema a la hora de realizar mediciones de tensión, por lo que el funcionamiento del dispositivo desarrollado es el esperado, y por tanto puede ser empleado para labores de desarrollo.

La aprobación del funcionamiento del módulo de Control y el submódulo de Voltímetro, también puede ser apoyada en la figura 6.11, donde se muestran los datos adquiridos en forma experimental de la tabla 6.1, que a su vez fueron enviados al computador como parte del funcionamiento de éste módulo, para que el usuario los pueda visualizar.

#### **6.2.2.2 Módulo de Control y submódulo de Amperímetro**

Como se puede observar en las tablas 6.2 y 6.3, los valores de tensión de salida en el sensor de corriente, no coincidieron con los valores sugeridos por el fabricante. Después de numerosos intentos de corrección del valor, mediante modificaciones en el circuito, se concluyó que el circuito integrado se encontraba dañado como consecuencia del proceso de soldadura al que fue sometido para fijarlo en una base o extensión para prototipado SIP (Single-In line Package)

Dada esta circunstancia y lo complicado que resultaba adquirir a corto plazo un sensor de corriente con similares características, la empresa sugirió para impedir cualquier tipo de retraso, excluir el sensor del sistema prototipo y partir de la suposición de que el sensor funcionaba en forma correcta para así lograr desarrollar el módulo de amperímetro con una funcionalidad correcta.

Como el comportamiento del sensor era conocido, su simulación estuvo basada en la aplicación de tensiones de diferente valor, directamente al pin del microcontrolador donde eventualmente iría conectado el sensor de corriente.

Las tensiones aplicadas al pin AMPMETER se mantuvieron dentro del rango de tensión en el cual el sensor podía variar la tensión de salida ante cambios en la corriente sensada. Con este sencillo procedimiento, se pudo simular el comportamiento del sensor y generar los datos experimentales que se muestran en la tabla 6.4 y en la figura 6.13, con lo cual se pudo comprobar que el comportamiento del módulo era el esperado y por tanto catalogar el módulo de

amperímetro como funcional.

Con respecto a la incertidumbre del amperímetro, ésta es de  $\pm 14\text{mA}$ . Este valor se obtuvo tomando en cuenta el error asociado en la tensión de salida del sensor de corriente que es de 1.5% y relacionándolo directamente con la corriente medida.

Dado este valor de incertidumbre, la empresa indica que el dispositivo es funcional, debido a que el objetivo primordial de la inclusión del amperímetro en la estación de trabajo, es monitorear corriente de consumo en los dispositivos que por lo general supera los 300mA

### **6.2.2.3 Módulo de Control y submódulo de LCD**

Como se puede observar en la figura 6.12, la pantalla del prototipo muestra la información que es adquirida.

El despliegue de información en pantalla, se da debido a que el hardware y software involucrados en esta sección funcionan de manera efectiva.

### **6.2.3 Módulo de JTAG**

Para comprobar el correcto funcionamiento del módulo JTAG, basta con observar la figura 6.16, específicamente las últimas líneas en donde se muestra, que el archivo llamado "setup", correspondiente a la imagen del firmware por actualizar, ubicado en la dirección "/home/esanabria/Desktop/vc4-mrs.5412-3183/images/" del computador en uso es descargado en la dirección 0x00000000 en 725910  $\mu\text{S}$ ., y seguidamente el dispositivo cuyo firmware fue actualizado, continua su funcionamiento normal una vez que Con respecto al funcionamiento del dispositivo en desarrollo al cual se le actualizó el firmware, éste inmediatamente finalizado el procedimiento ejecutado con el módulo de Jtag, su sistema se reinicia y el nuevo firmware puede ser almacenado en su memoria flash.

#### **6.2.4. Módulo de Fuente de CD**

Los datos contenidos en las columnas de la derecha de las tablas 6.5,6.6 y 6.7 muestran los valores de tensión experimentales obtenidos al seleccionar las diferentes configuraciones predefinidas de resistencias para variar la tensión de salida.

Como se puede observar en las mencionadas tablas, el empleo de diferentes cargas, en este caso: circuito abierto,  $2.2\text{K}\Omega$  y  $220\Omega$  no afecta el funcionamiento del circuito de la figura 5.14. Esto se debe a que el regulador ajustable LM317 es sumamente estable, y de acuerdo a la ecuación 5.1, la tensión de salida de éste depende únicamente de la configuración de resistencias  $R_1$  y  $R_2$  y nunca de el valor de la resistencia de carga.

La diferencia de tensiones experimentales, con respecto a las tensiones teóricas, se debe a que en el circuito prototipo se emplearon resistencias de valores cercanos a los especificados en al figura 5.14.

Con respecto a la corriente que éste módulo o fuente es capaz de entregar, el valor máximo de la misma es de 1.5A. Este valor es considerado por RidgeRun como óptimo, dado que el empleo que se le dará a este módulo, será único y exclusivo para energizar dispositivos empotrados, cuyo consumo de corriente nunca sobrepasa los 750mA.

Por otra parte, el margen de error en la tensión de salida de este módulo, está directamente relacionado con la precisión en los resistores que se empleen, dado que influyen directamente sobre la misma.

#### **6.2.5. Margen de Error del Sistema**

El margen de error del sistema está dado por los módulos en donde la función principal esté relacionada con alguna variable física, que en este caso lo serían el módulo de voltímetro, amperímetro y fuente de CD. Los demás módulos no son

tomados en cuenta, para esta evaluación dado que en caso de que se presente un error en dichos módulos, automáticamente estos dejarían de ser funcionales. Por ende el margen de error del voltímetro es de  $\pm 0.001V$ , el del amperímetro es de  $0.014mA$  y el de la fuente de CD queda sujeto a la precisión de los resistores que sean empleados.

## **CAPÍTULO 7. Conclusiones y recomendaciones**

### **7.1 Conclusiones**

- Se diseñó e implementó a nivel de prototipo de forma satisfactoria para RidgeRun Engineering Limitada un sistema multiherramienta para ser usado en el desarrollo de sistemas empotrados.
- Se comprobó mediante la puesta en práctica que el sistema diseñado es útil para el desarrollo de sistemas empotrados que se da en RidgeRun Engineering.
- El software que se desarrollo permitió configurar y emplear el sistema desarrollado.
- Se generó una documentación exhaustiva que incluye un diagrama esquemático completo de sistema que le permite a la empresa realizar modificaciones, o bien crear un circuito impreso.

### **7.2 Recomendaciones**

1. Todo sistema con miras a mejorarse en el futuro debe ser estandarizado en cuanto a conectores se refiere, para que en el futuro el no contar con los modelos requeridos no sea sinónimo de que el sistema se vuelva inmejorable
2. Previo a la realización de cualquier diseño electrónico, se debe investigar los posibles componentes a usar, sus encapsulados, su estado de producción (descontinuado, en producción, etc) y los precios de los mismos al comprar grandes lotes de componentes.
3. Al desarrollar software o hardware, en la medida de lo posible se debe tratar de que éste sea multiplataforma, es decir que el sistema operativo con el cual se relacione, no limite el funcionamiento del mismo.

## 8 BIBLIOGRAFIA

[1] Texas Instruments. MSP430x4xx Family User's Guide. Application Note Slau056g [en línea]. Texas Instruments Incorporated, 2007. Disponible en <http://focus.ti.com/lit/ug/slau056g/slau056g.pdf> [Consulta 10 de febrero de 2008].

[2] Underwood, Steven (2003). MSPGCC. A port of the GNU tools to the Texas Instruments MSP430 microcontrollers. Disponible en [www.eecs.harvard.edu/~konrad/projects/motetrack/mspgcc-manual-20031127.pdf](http://www.eecs.harvard.edu/~konrad/projects/motetrack/mspgcc-manual-20031127.pdf) . [Consulta 13 de marzo de 2008].

[3] Future Technology devices Internation Ltd. FT2232D Dual USB UART/FIFO I.C. Future Technology Devices Ltd. 2006. Disponible en [www.ftdichip.com/Documents/DataSheets/DS\\_FT2232D.pdf](http://www.ftdichip.com/Documents/DataSheets/DS_FT2232D.pdf). [Consulta 25 de febrero de 2008.]

[4] USB Protocol (s.f) Disponible en [http://www2.rad.com/networks/2000/usb/maintxt.htm#USB\\_Protocol](http://www2.rad.com/networks/2000/usb/maintxt.htm#USB_Protocol) [Consulta 3 de marzo de 2008]

[5] Axelson, Jan. USB Complete. 3ed. USA: Lakeview Research, 2005.

[6] SPI. [en línea]. Disponible en <http://es.wikipedia.org/wiki/SPI>. [Consulta 13 de abril de 2008].

[7] **SPI. Serial Peripheral Interface** (s.f.) Disponible en <http://www.mct.net/faq/spi.html>

[8]. Bennetts, R.G. **The Boundary Scan Tutorial. ASSET InterTech Inc.** [en línea] . Disponible en [www.asset-intertech.com/pdfs/boundaryscan\\_tutorial.pdf](http://www.asset-intertech.com/pdfs/boundaryscan_tutorial.pdf) [Consulta: 12 de marzo de 2008].

[9] ACS712. Allegro Microsystems Inc. [en línea]. Disponible en <[www.allegromicro.com/en/Products/Part\\_Numbers/0712/0712.pdf](http://www.allegromicro.com/en/Products/Part_Numbers/0712/0712.pdf)> [Consulta: 10 de marzo de 2008]

[10] Ley de Ohm. (s.f) Disponible en < [http://www.unicrom.com/Tut\\_leyohm.asp](http://www.unicrom.com/Tut_leyohm.asp)> [Consulta: 13 de abril.]

[11] Material de Clase por el profesor Ing. Victorino Rojas Madrigal. Curso Circuitos Eléctricos en Corriente Continua, Primer semestre 2002, Escuela de Ingeniería Electrónica, ITCR.

[12] Romera, Luis F. El Efecto Hall .2004. Disponible en <<http://perso.wanadoo.es/frs88/tpn/desc/hall.htm>> . [Consulta: 13 de abril de 2008.]

[13] Texas Instruments. MSP430x4xx Device Datasheet. Note SLAS380B [en línea]. Texas Instruments Incorporated, 2007. Disponible en <<http://www.ti.com/lit/gpn/msp430fg439>> [Consulta 10 de febrero de 2008].

[14] Abel, Peter. Lenguaje Ensamblador y programación para IBM PC y compatibles. México .Prentice Hall Hispanoamericana, 1996.

[15] Kernighan, B. & Ritchie, D. El lenguaje de Programación C. México. Prentice Hall Hispanoamericana , 1989.



## 9 APENDICES

### Apéndice A1. Glosario

**Acondicionador de señal:** dispositivo que convierte un tipo señal electrónica en otro tipo. Su uso principal es convertir una señal que resulta difícil de ser leída por un instrumento convencional en un formato más fácil de leer.

**Alto nivel:** Permite a los programadores escribir instrucciones en un lenguaje más familiar para ellos y que contiene notaciones matemáticas comúnmente utilizadas

**ARM:** Familia de microprocesadores RISC (Reduced Instruction Set Computer) diseñados por Acorn Computers y desarrollados por Advanced RISC Machines Ltda.

**Ascii:**American Standard Code for Information Exchange , estándar para representar caracteres y símbolos en forma electrónica.

**Bajo Nivel:** Lenguaje de programación que el ordenador puede entender a la hora de ejecutar programas, lo que aumenta su velocidad de ejecución, pues no necesita un intérprete que traduzca cada línea de instrucciones.

**Código Libre:** también conocido como código abierto. Denominación para aquellas aplicaciones de software que tienen su código fuente liberado por lo que puede ser distribuido libremente.

**DID:**Device Identification. Código que identifica a un dispositivo USB específico.

**FPGA:** Field Programmable Gate Array. Dispositivo semiconductor que posee bloques lógicos de propósito general comunicados por conexiones programables manipulables por los usuarios para implementar aplicaciones específicas..

**Gate:** Terminal de un transistor de efecto de campo (FET), la cual es capaz de controlar el paso de corriente de la terminal Source a la terminal Drain.

**GPIO:** General Purpose Input/Output. Este término es usado para catalogar a aquellos pines de un microcontrolador o microprocesador que no tienen una función específica asignada y pueden ser usados como entradas o salidas de datos.

**Ensamblador:** programa de bajo nivel que toma instrucciones básicas de el computador y las convierte en un patrón de bits que el procesador puede usar para ejecutar operaciones básicas.

**Impedancia:** Es una magnitud que se compone de la suma de una componente resistiva debida a las resistencias, y una componente reactiva debido a las bobinas y capacitores, que establece la relación entre la tensión y la intensidad de corriente.

**JTAG:** Protocolo serial empleado para desarrollo de sistemas empotrados y prueba de circuitos impresos.

**Led:** Light emitting diode o diodo emisor de luz.

**Lenguaje C:** Lenguaje de programación de alto nivel

**LVTTTL:**Low Voltage TTL. .Tecnología de construcción de circuitos electrónicos digitales cuya tensión de alimentación se halla entre 3.0V y 3.6V

**Maestro SPI:** dispositivo que genera la señal de reloj y controla la activación y desactivación de otros elementos SPI llamados esclavos. Por definición este dispositivo siempre es el responsable de iniciar cualquier transmisión de datos.

**OpenOCD:** Software abierto desarrollado para emplear el protocolo JTAG en procesadores ARM

**PID:** Physical Interface Device: Código que identifica la interfaz física de un dispositivo USB.

**Pila:** Estructura de datos que permite almacenar y recuperar datos. El modo de acceso o entrada y salida de datos se hace en forma LIFO (Last In-first Out), último en entrar, primero en salir

**Polling:** Operación de consultas repetitivas y separadas por un intervalo de tiempo constante, a un dispositivo de hardware para enviarle o recibirle información de este.

**Script:** conjunto de instrucciones agrupadas en un mismo archivo. Permite la automatización de pequeñas tareas mediante la ejecución del archivo que

contiene las instrucciones.

**Sistemas Empotrados:** Sistema informático (hardware + software) desarrollado para ejecutar una aplicación específica.

**Software propietario:** también llamado software no libre. Software en el que los usuarios tienen limitaciones para modificarlo o redistribuirlo sin permiso del autor.

**SPI:** Serial Peripheral Interface. Es un estándar de comunicación creado por Motorola en donde un dispositivo se puede comunicar con otros mediante la relación maestro-esclavo. ( el dispositivo maestro siempre inicia la transmisión de datos)

**USB:** Universal Serial Bus. Protocolo que permite agregar periféricos al computador sin la necesidad de reiniciar el sistema.

**TTL:** Transistor-Transistor logic .Tecnología de construcción de circuitos electrónicos digitales cuya tensión de alimentación se halla entre 4.75V y 5.25V

**VID:** Vendor Identification.Código que identifica al vendedor de un dispositivo USB.

## Apéndice A 2. Software desarrollado para el microcontrolador.

```
#include <msp430xG43x.h>
#include <signal.h>
#include <io.h>
#include <math.h>
#include </opt/cdk4msp/msp430/include/stdio.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
//*****LCD VARIABLES*****
int LCD_CTRL=0x00;
int i; //
int lenght=15;
long h;
result[100];
// *****END LCD VAR SEGMENT*****
//*****//ADC and TX VARIABLES *****
char dato=1;
long vale=1;
long value_c=1;
int aux=1;
int i=0;
long u=1;
long dos=1;
long tres=1;
long cuatro=1;
long cinco=1;
long seis=1;
long siete=1;
long ocho=1;
int pos1=1;
int pos2=1;
int pos3=1;
int pos4=1;
long convert1=1;
long convert2=1;
long convert3=1;
long convert4=1;
long convert5=1;
long pos1_c=1;
long pos2_c=1;
//*****Delay procedure*****
void delay_d(unsigned int p){ // delay procedure
asm ("mov r0,r13"); //moves the PC to R13
```

```

asm ("add #4,r14"); //calculates the next PC address
asm ("push r14"); // stores the new PC value in the stack
while (p--){
nop();
nop();

if (p==0){
asm ("pop r14"); // take out the PC value stored
asm ("mov r14, r0"); // updates the new PC address with the value already calculated
}
}
}
//*****End Delay *****
//*****Begin send byte*****
void send_byte_LCD (int mode, int byte){
asm ("add #10,r13"); //moves the PC to R13
asm ("push r13");// stores the new PC value in the stack
LCD_CTRL= (LCD_CTRL |0x10); //on rs
P5OUT= LCD_CTRL; //out rs
P4OUT= byte; // out E
asm ("mov r0,r14");
delay_d(473); //wait
LCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD
P5OUT= LCD_CTRL;//out enable
asm ("mov r0,r14");
delay_d(473);//wait
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
P5OUT=LCD_CTRL; // off enable on port
asm ("mov #0x262,r1");
asm ("pop r13");
asm ("mov r13, r0");
}
//*****End send byte *****
//*****Jump line *****
void SLin (void){
asm ("mov r0,r15");//moves the PC to R13
asm ("add #4,r14");//calculates the next PC address
asm ("push r14");// stores the new PC value in the stack
LCD_CTRL=(LCD_CTRL & 0xEF);// turn off RS
P5OUT=LCD_CTRL;
asm ("mov r0,r14");
delay_d(10);
LCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD
P5OUT=LCD_CTRL; // turn on Enable LCD
asm ("mov r0,r14");

```

```

delay_d(23);
P4OUT=0xC0; // Change line command
asm ("mov r0,r14");
delay_d(35);
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
P5OUT=LCD_CTRL; // off enable on port
asm ("mov r0,r14");
delay_d(28); //wait 170 u
asm ("mov #0x27e, r1");// 0x27e is the next address where PC must go
}
//*****End Jump Line *****
//*****CLEAN LCD *****
void clean_LCD(){
asm ("mov r0,r15");
asm ("add #4,r14");
asm ("push r14");
int t=0x04;
while (t-- ) {
LCD_CTRL=(LCD_CTRL & 0xEF); // turn off RS
P5OUT=LCD_CTRL;
asm ("mov r0,r14");
delay_d(10);
LCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD
P5OUT=LCD_CTRL; // turn on Enable LCD
asm ("mov r0,r14");
delay_d(23);
P4OUT=0x01; // Clear LCD
asm ("mov r0,r14");
delay_d(35);
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
P5OUT=LCD_CTRL; // off enable on port
asm ("mov r0,r14");
delay_d(28); //wait 170 us
} //end while
asm ("mov #0x27a, r1");
asm ("pop r14");
asm ("mov r14 ,r0");
} // end clean_LCD

//*****END CLEAN LCD PROCEDURE*****
//*****Display_Off_LCD*****
void LCD_off(void){
asm ("mov r0,r15");
asm ("add #4,r14");
asm ("push r14");

```

```

LCD_CTRL=(LCD_CTRL & 0xEF);// turn off RS
P5OUT=LCD_CTRL;
asm ("mov r0,r14");
delay_d(10);
LCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD
P5OUT=LCD_CTRL; // turn on Enable LCD
asm ("mov r0,r14");
delay_d(23);
P4OUT=0x0B; // Turn Off LCD
asm ("mov r0,r14");
delay_d(35);
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
P5OUT=LCD_CTRL; // off enable on port
asm ("mov r0,r14");
delay_d(28); //wait 170 us

asm ("pop r14");
asm ("mov r14, r0");
}
//*****End Display_off_LCD*****
//*****Display_On_LCD*****

void LCD_on(){
asm ("mov r0,r15");
asm ("add #4,r14");
asm ("push r14");
LCD_CTRL=(LCD_CTRL & 0xEF);// turn off RS
P5OUT=LCD_CTRL;
asm ("mov r0,r14");
delay_d(10);
LCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD
P5OUT=LCD_CTRL; // turn on Enable LCD
asm ("mov r0,r14");
delay_d(23);
P4OUT=0x0E; // Turn Off LCD
asm ("mov r0,r14");
delay_d(35);
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
P5OUT=LCD_CTRL; // off enable on port
asm ("mov r0,r14");
delay_d(28); //wait 170 us
asm ("pop r14");
asm ("mov r14, r0");
asm ("mov #0x27e, r1");
asm ("pop r14");

```

```

asm ("mov r14, r0");
}
//*****End_Display_On_LCD*****
//*****CURSOR to Home HOME*****
void cursor_home(){ //cursor in home position
asm ("mov r0,r15");
asm ("add #4,r14");
asm ("push r14");
LCD_CTRL=(LCD_CTRL & 0xEF); // turn off RS
P5OUT=LCD_CTRL;
LCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD
P5OUT=LCD_CTRL; // turn on Enable LCD
asm ("mov r0,r14");
delay_d(10);
P4OUT=0x02; // Clear LCD
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
asm ("mov r0,r14");
delay_d(10); // to make an enable larger
P5OUT=LCD_CTRL; // off enable on port
asm ("mov r0,r14");
delay_d(23); //wait 170 us
asm ("mov #0, r14");
asm ("mov #0x26e, r1");
asm ("pop r14");
asm ("mov r14, r0");
} // end cursor_home
//*****END CURSOR TO HOME
//*****INIT_LCD*****
void init_LCD(){
asm ("mov r0,r15");
asm ("add #4,r14");
asm ("push r14");
LCD_CTRL=0x04; // Load LCD_CTRL variable with enable condition
P5OUT=LCD_CTRL; // Put on port Enable
P5OUT=0x00; // Turn off enable
P5OUT=LCD_CTRL; //turn on ENABLE_LCD
P4OUT=0x3C; // Init command, interface length 8 bits, 2 Lines , 5x10 (Character Font)
asm ("mov r0,r14");
delay_d(35); // keep enable on for 203 msLCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD
P5OUT=LCD_CTRL; // turn on Enable LCD
P4OUT=0x01; // Clear LCD
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
asm ("mov r0,r14");
delay_d(10); // to make an enable larger
P5OUT=LCD_CTRL; // off enable on port

```



```

asm ("mov r0,r14");
delay_d(23); //wait 170 us
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
P5OUT=LCD_CTRL; //Off enable on port
asm ("mov r0,r14");
delay_d(868); // wait 5 milisenconds
LCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD
P5OUT=LCD_CTRL; // turn on Enable LCD
P4OUT=0x3C; //Init command, interface lenght 8 bits, 2 Lines , 5x10 (Character Font)
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
asm ("mov r0,r14");
delay_d(10); // to make an enable larger
P5OUT=LCD_CTRL; // off enable on port
asm ("mov r0,r14");
delay_d(23); //wait 170 us
LCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD
P5OUT=LCD_CTRL; // turn on Enable LCD
P4OUT= 0x0C;//0x0E;//0x0F;//Enable Display, turn display on, turn cursor on, cursor blink ( setup to check lcd
status)
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
asm ("mov r0,r14");
delay_d(10); // to make an enable larger
P5OUT=LCD_CTRL; // off enable on port
asm ("mov r0,r14");
delay_d(23); //wait 170 us
LCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD
P5OUT=LCD_CTRL; // turn on Enable LCD
P4OUT= 0x06;// S0et cursor move direction, Increment the cursor after each byte written to display
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
asm ("mov r0,r14");
delay_d(10); // to make an enable larger
P5OUT=LCD_CTRL; // off enable on port
asm ("mov r0,r14");
delay_d(23); //wait 170 us
LCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD

```

```

P5OUT=LCD_CTRL; // turn on Enable LCD
P4OUT=0x1C;// Move cursor Display shift on and shift right direction
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
asm ("mov r0,r14");
delay_d(10); // to make an enable larger
P5OUT=LCD_CTRL; // off enable on port
asm ("mov r0,r14");
delay_d(23); //wait 170 us
asm ("mov r0,r14");
LCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD
P5OUT=LCD_CTRL; // turn on Enable LCD
P4OUT=0x02;//Return cursor and LCD to Home position
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
asm ("mov r0,r14");
delay_d(10); // to make an enable larger
P5OUT=LCD_CTRL; // off enable on port
asm ("mov #0x27e, r1");
asm ("pop r14");
asm ("mov r14, r0");
}
//***** End INIT*****
//*****Position procedure for cursor on LCD*****
void PosLCD( int xPosx){
//int xPosx;
asm ("mov r0,r15");
asm ("add #4,r14");
asm ("push r14");
xPosx= (xPosx | 0x80);
asm ("mov r0,r14");
send_byte_LCD(0,'C');
asm ("mov #0x27e, r1");

asm ("pop r14");
asm ("mov r14, r0");
}
// *****End position procedure for cursor on LCD*****
//***** send message to LCD*****
void message_LCD(char a){
asm ("mov r0,r15");
asm ("add #4,r14");
asm ("push r14");
switch (a){
case '&': PosLCD(0x40); break; // if character is &, jump line
default: send_byte_LCD(0x01,a);break;//send the character to the LCD
}
}

```

```

asm ("mov #0x27e, r1");
asm ("pop r14");
asm ("mov r14, r0");
} // *****end end message to LCD*****
//*****procedure to calculate the voltage value*****
void volt(){
u=vale*3052;
dos= (u/4095); //4095 is max value from adc
if (dos <1000){
tres=dos/100; //break down complete value in separated digits
cuatro=dos/10;
cinco=cuatro-(tres*10);
seis=dos-(cuatro*10);
siete=0;
ocho=0; //convert every digit to Ascii
pos2= (0x030 | tres);
pos3= (0x030 | cinco); //convert every digit to Ascii
pos4= (0x030 | seis); //convert every digit to Ascii
TXBUF0=0x0A; //Transmit data through serial interface
TXBUF0=pos1; //Transmit data through serial interface
TXBUF0=pos2; //Transmit data through serial interface
TXBUF0=pos3; //Transmit data through serial interface
TXBUF0=0x06D;
TXBUF0=0x056;
}
if (dos >1000){
tres=dos/1000; //break down complete value in separated digits
cuatro=(dos/100);
cinco=cuatro-(tres*10);
seis=(dos/10);
siete=seis-(cuatro*10);
ocho= dos-(seis*10);
pos1= (0x030 | tres); //break down complete value in separated digits
pos2= (0x030 | cinco); //break down complete value in separated digits
pos3= (0x030 | siete); //break down complete value in separated digits
pos4= (0x030 | ocho);

TXBUF0=0x0A;
TXBUF0=pos1; //Transmit data through serial interface
TXBUF0=0x2E; //Transmit data through serial interface
TXBUF0=pos2 ;//cuatro;
TXBUF0=pos3; //Transmit data through serial interface
TXBUF0=pos4; //Transmit data through serial interface
TXBUF0=0x056;
}
}

```

```

LCD_CTRL=(LCD_CTRL & 0xEF); // turn off RS
P5OUT=LCD_CTRL;
LCD_CTRL=(LCD_CTRL | 0x04); //On enable LCD
P5OUT=LCD_CTRL; // turn on Enable LCD
asm ("mov r0,r14");
P4OUT=0x02; // Clear LCD
LCD_CTRL=(LCD_CTRL & 0xFB); // turn off Enable
asm ("mov r0,r14");
P5OUT=LCD_CTRL; // off enable on port
asm ("mov &0x290, r0");
}
//*****end procedure to calculate the voltage value*****
// *****procedure to calculate the voltage value with scale correction*****
void volt_10 {
u=vale*3052;
dos= (u/4095); //4095 is max value from adc
tres=dos/1000;
cuatro=(dos/100); //break down complete value in separated digit
cinco=cuatro-(tres*10);
seis=(dos/10);
siete=seis-(cuatro*10);
ocho= dos-(seis*10);
pos1= (0x030 | tres);
pos2= (0x030 | cinco); //convert value to Ascii
pos3= (0x030 |siete);
pos4= (0x030 |ocho);
TXBUF0=0x0A;
TXBUF0=pos1; //Transmit data through serial interface
TXBUF0=pos2 ;
TXBUF0=0x2E; // . to indicate scale
TXBUF0=pos3;
TXBUF0=pos4;
TXBUF0=0x056;
} //
// *****end procedure to calculate the voltage value with scale correction*****
//*****procedure to calculate current value*****
void current() {
u=value_c*3052;
convert=(u/4095); //3
if(convert1>=2700){
convert1=(convert-2500)*100; //(3000-2500)*10= 5000

convert2=convert1/2; //2500
convert3=convert2/1000; //2
convert4= convert2-(convert3*1000); //2500-2000= 500

```

```

convert5= convert4/100;//5
pos1_c= (0x030 | convert3); //convert data to Ascii
pos2_c= (0x030 | convert4);
TXBUF0=0x0A;
TXBUF0=pos1_c;
TXBUF0=pos2_c;
TXBUF0=0x030; // transmit data through serial interface
TXBUF0=0x030;
TXBUF0=0x060;// transmit data through serial interface
TXBUF0=0x041;
} //if >=2700
if(convert1<2700){
convert1=(convert-2.5)*10000; //(2650-2500)*10= 1500
convert2=convert1/2; //750
convert3=convert2/100; //7
convert4= convert2-(convert3*100); //2500-2000= 50
convert5= convert4/10;//5
pos1_c= (0x030 | convert3); //convert data to Ascii
pos2_c= (0x030 | convert5);
TXBUF0=0x0A;
TXBUF0=pos1_c;
TXBUF0=pos2_c;
TXBUF0=0x030;
TXBUF0=0x060;// transmit data through serial interface
TXBUF0=0x041;
} //if <2700
} //*****end procedure to calculate current value *****
void main(void) //***** principal procedure*****
{
    WDTCTL = WDTPW + WDTHOLD;          // Stop WDT
//*****usart configuration*****/

    asm ("mov #0x280,r1");
    FLL_CTL0 |= XCAP18PF;                // Configure load caps
    P2SEL |= 0x30;                       // P2.4,5 = USART0 TXD/RXD
    P4SEL=0x00;
    ME1 |= UTXE0 + URXE0;                // Enable USART0 TXD/RXD
    UCTL0 |= CHAR;                       // 8-bit character
    UTCTL0 |= SSEL1;                     // UCLK = SMCLK
    UBR00 = 0x6D;                        // 1MHz 9600
    UBR10 = 0x00;                        //
    UMCTL0 = 0x03;                       // Modulation
    UCTL0 &= ~SWRST;                     // Initialize USART state machine
    IE1 |= URXIE0;                       // Enable USART0 RX interrupt
    P2DIR |= 0x10;                       // P2.4 output direction

```

```

P4DIR=0xFF;
P5DIR =0xFF;
P5SEL= 0x00;
//*****End Usart initialization*****
asm ("mov r0,r14");
init_LCD(); // init LCD procedure
//*****ADC initialization*****
P6SEL |= 0x0F;           // Enable A/D channel A0
ADC12CTL0 = ADC12ON+SHT0_2+MSC; // Turn on ADC12, set sampling time
ADC12CTL1 = SHP+CONSEQ_1; // Use sampling timer
ADC12MCTL0=INCH_0;
ADC12MCTL1=INCH_1+EOS;
ADC12CTL0 |= ENC;       // Enable conversions
//***** END ADC initialization*****//
while (1){
  _DINT(); // Disable all interruptions
  asm ("mov &0x0204,r15");
  if (dato=='a'){ // if "a" is received, enable voltmeter without scale correction
    ADC12CTL0 |= ADC12SC;
    vale=ADC12MEM0;
    asm ("mov r0,r14");
    clean_LCD();
    asm ("mov r0,r14");
    asm ("add #12,r14");
    asm ("mov r14,&0x0290");
    volt(); // call vltage measurement procedure
    dato= ' '; //clean data
    asm ("mov r0,r13");
    send_byte_LCD(0x01,pos1);//send data to LCD
    send_byte_LCD(0x01,pos2);
    send_byte_LCD(0x01,pos3);
    send_byte_LCD(0x01,0x030);//send data to LCD
    send_byte_LCD(0x01,0x06D);//send data to LCD
    send_byte_LCD(0x01,0x056);//send data to LCD
  }/'a'
  if (dato=='d'){ // if "d" is received, enable voltmeter to measure values greater than 3.3V
    ADC12CTL0 |= ADC12SC;
    vale=ADC12MEM0;
    asm ("mov r0,r14");
    clean_LCD(); // Clean LCD
    asm ("mov r0,r14");
    asm ("add #12,r14");
    asm ("mov r14,&0x0290");
    volt_10();
    dato= ' '; //clean data

```

```

asm ("mov r0,r13");
dato= ' '; //clean data
asm ("mov r0,r13");
send_byte_LCD(0x01,pos1);//send data to LCD
send_byte_LCD(0x01,pos2);//send data to LCD
send_byte_LCD(0x01,pos3);//send data to LCD

send_byte_LCD(0x01,0x056);//send data to LCD
} // 'd'
asm ("mov &0x0204,r15");
if (dato== 'e'){ // if "e" is received enable current meter
asm ("mov r0,r14");
clean_LCD();
ADC12CTL0 |= ADC12SC;
value_c=ADC12MEM1;
asm ("mov r0,r14");
dato=' '; // clean data
current();
asm ("mov r0,r13");
send_byte_LCD(0x01,pos1);// send data to LCD
send_byte_LCD(0x01,pos2);//send data to LCD
send_byte_LCD(0x01,pos3);//send data to LCD
send_byte_LCD(0x01,0x2E);//send data to LCD
send_byte_LCD(0x01,0x030);//send data to LCD
send_byte_LCD(0x01,0x06D);//send data to LCD
send_byte_LCD(0x01,0x056);//send data to LCD
asm ("mov r0,r14");
} // 'e'
_EINT();//Enable interruptions
} // end while
}
interrupt (UART0RX_VECTOR) usart0_rxISR (void) // procedure triggered every time serial interrupt is
activated
//while ((IFG1 & UTXIFG0)); // USART0 TX buffer ready?
_DINT();
dato=RXBUF0;
asm ("mov #0x00008796,r0");

}

```

### **Apéndice A 3 . Script desarrollado para realizar multiples lecturas con el voltímetro.**

Nota: El rango de mediciones con el presente script está enfocado al rango de 0V a 3V.

```
#!/bin/sh
clear
while true;
do echo -n 'a' >/dev/ttyUSB2 ;
// option -n send character without change line instruction
sleep 1;
// time in seconds between every voltage measurement.
done
```

### **Apéndice A 4 . Script desarrollado para realizar múltiples lecturas con el voltímetro.**

Nota: El rango de mediciones con el presente script está enfocado al rango de 1V a 15V.

```
#!/bin/sh

clear

while true;

do echo -n 'd' >/dev/ttyUSB2 ;
// option -n send character without change line instruction
sleep 1;
// time in between every voltage measurement.
Done
```

### **Apéndice A 5. Script desarrollado para realizar múltiples lecturas con el amperímetro.**

```
#!/bin/sh

clear

while true;

do echo -n 'e' >/dev/ttyUSB2 ;
// option -n send character without change line instruction
sleep 1;
// time in between every voltage measurement.
done
```



## Apéndice A 6. Código para generar el archivo de configuración “dlp2232.cfg” de OpenOCD para que reconozca el dispositivo JTAG desarrollado.

Nota: este código fue basado en el código desarrollado por Todd Fischer para dispositivos Olimex con circuitos integrados ft2232c

Los valores propios del sistema: VID, PID, DID así como el dato “device description” fueron obtenidos mediante el comando : **lsusb** en Linux

```
#daemon configuration
telnet_port 4444
gdb_port 3333
#interface
interface ft2232
#ft2232_device_desc "USB-JTAG A"
ft2232_device_desc "USB <-> Serial Cable"
ft2232_layout usbjtag
ft2232_vid_pid 0x0403 0x6010
jtag_speed 10

#use combined on interfaces or targets that can't set TRST/SRST separately
reset_config trst_and_srst srst_pulls_trst
#jtag scan chain
#format L IRC IRCM IDCODE (Length, IR Capture, IR Capture Mask, IDCODE)
#jtag_device 4 0x1 0xf 0xe
jtag_device 8 0x1 0x0 0x0
jtag_device 4 0x1 0x0 0xe
#target configuration
daemon_startup reset
# From Todd Notes
#target <type> <endianness> <reset mode> <chainpos> <variant>
target arm926ejs little run_and_init 1 arm926ejs
#target_script 0 reset omap5912_osk.init
run_and_halt_time 0 30
```

**Apéndice A 7. Lista Materiales y cotización de la producción del dispositivo desarrollado**



## Apéndice A. 8 Información del Proyecto

### Información del estudiante:

**Nombre:** Erick Sanabria Calvo

**Cédula:** 3-383-611                      **Carné ITCR:** 200018468

**Dirección de su residencia en época lectiva:** Oreamuno, San Rafael. Del BCR 200 m este y 75 sur. Cuarta casa mano derecha.

**Dirección de su residencia en época no lectiva:** Oreamuno, San Rafael. Del BCR 200 m este y 75 sur. Cuarta casa a mano derecha.

**Teléfono en época lectiva:** 2591-5389 **Teléfono época no lectiva:** 2591-5389

**Email:** erick.sanabria@ridgerun.com                      **Fax:** -

### Información del proyecto:

**Nombre del Proyecto:** Diseño y construcción de una estación de trabajo para sistemas empotrados

**Área del Proyecto:** Electrónica Digital

### Información de la empresa:

**Nombre:** RidgeRun Limitada

**Zona:** San José, Curridabat.

**Dirección:** Frente al BAC San José de Curridabat. Edificio Plaza Domus, Piso 2, local #10

**Teléfono:** 22259596                      **Fax:** 22259596                      **Apartado:** -

**Actividad Principal:** Desarrollo y soporte de sistemas empotrados.

### Información del encargado en la empresa:

**Nombre:** Jorge Manuel Rivera Gutiérrez

**Puesto que ocupa:** Ingeniero desarrollador de sistemas empotrados.

**Departamento:** Soporte electrónico

**Profesión:** Ingeniero en Electrónica                      **Grado académico:** Licenciado

**Teléfono:** 88224785                      **Email:** jorge.rivera@ridgerun.com

## **Apéndice A 9 Breve resumen del circuito, esquemático final y circuito capas del circuito impreso**

**Nombre: USB HUB**

**Hoja: 1/9**

El Hub USB se incluyó en el diseño con el fin de que la estación de trabajo para sistemas empotrados, le permitiese al usuario contar con más puertos USB de los que posee el computador.

En el presente circuito se emplearon 2 puertos USB para conectar en forma permanente los módulos desarrollados de la estación de trabajo al computador. Los restantes 2 puertos no se utilizaron en una conexión específica, para que el usuario los pudiese emplear libremente conectando cualquier dispositivo que requiriese.

En caso de ser necesario, es sistema tiene la capacidad de apagar alguno de los puertos en caso de que este presente un cortocircuito o el periférico conectado sobrecargue el sistema.

**Nombre: Fuente Externa de Alimentación ( External Power Supply)**

**Hoja : 2/9**

La fuente externa de alimentación se incluyó en el sistema con el fin de alimentar los dispositivos electrónicos con los que se está trabajando.

Este módulo está diseñado para trabajar únicamente con sistemas empotrados, por lo que sus valores máximos de tensión y corriente son 5.25V y 1.5A respectivamente.

Los valores de tensión están predefinidos y son seleccionables por el usuario mediante el movimiento de jumpers.

Las tensiones de salida posibles son: 1.5V, 1.8V, 2.5V, 3.0V, 3.3V, 3.6V, 5.0V.

Además de estos posibles valores, el usuario puede seleccionar vía jumper la modalidad de tensión de salida ajustable, en donde mediante el ajuste de un potenciómetro, la tensión de salida puede variar en el rango de 1.5V a 5.0V.

En caso de cortocircuito o sobredemanda de corriente, el circuito integrado empleado (LM317) cuenta con protección que inmediatamente apaga el circuito.

**Nombre: Módulo JTAG y Expansión****Hoja : 3/9**

Este circuito se encuentra basado en el circuito integrado FT2232D y se encuentra conectado directamente al puerto 2 del HUB USB (ver hoja 1/9).

El puerto A de este circuito se encuentra conectado directamente al conector P5 o conector JTAG, a través del cual se ejecutan todas las acciones que requieren este protocolo. El módulo de JTAG es usado principalmente en RidgeRun para actualizar el firmware de los dispositivos en desarrollo.

El puerto B se encuentra conectado al conector P6 que no tiene una función específica asignada, sino que se dejó previsto como conector multipropósito para agregar periféricos al sistema. Los periféricos que se pueden agregar pueden emplear protocolos serie o paralelos ya que este circuito es compatible con estos.

**Nombre: Módulo Serial****Hoja : 4/9**

El circuito que abarca ambos módulos, también está basado en un circuito FT2232D, conectado al puerto 1 del HUB USB (ver Hoja 1/9).

El puerto A de este circuito integrado fue empleado para establecer la comunicación entre el Computador y el microcontrolador empleado (MSP430) que controla tanto el LCD como el voltímetro y el amperímetro.

Por otra parte, mediante el puerto B se le agrega a la estación de desarrollo diseñada, la capacidad de entablar comunicaciones seriales. La comunicación serial se da entre el computador y cualquier dispositivo conectado al conector J1. Si la empresa lo requiriese, puede eventualmente modificar el software del microcontrolador para emplear este puerto con algún dispositivo, dado que eléctricamente las conexiones fueron previstas.

**Nombre: Módulo de Control****Hoja: 5/9**

El circuito del módulo de control se encuentra basado en un microcontrolador MSP430FG439 de Texas Instruments.

Los módulos de voltímetro, amperímetro y LCD dependen en su totalidad de este circuito dado que es quien los controla mediante la generación de señales de control.

La comunicación de este módulo con el computador se hace a través de el módulo serial.

Al circuito de control se agregaron dos conectores: P9 y P10.

El conector P9 es un conector al cual se le cablearon todos los pines bidireccionales sobrantes del microcontrolador, para así tener centralizados en un sólo conector, pines multipropósito que en un futuro podrían ser empleados para realizar depuraciones de software cargado en el microcontrolador.

El conector P10 es un conector mediante el cual se pueden agregar dispositivos SPI al sistema si la empresa o el usuario lo requiriese en un futuro. Todas las señales de control que eventualmente requeriría un periférico SPI se encuentran centralizadas en este conector. En caso de que el orden de los pines no coincida con el periférico que se desea agregar, se puede desarrollar un cable para tal fin.

**Nombre: Módulo de LCD****Hoja: 6/9**

El circuito de este módulo está constituido por un LCD y unos circuitos cambiadores de nivel, dado que la interfaz del microcontrolador funciona con 3.3V como máximo y el LCD funciona con tensiones de 5V.

**Nombre: Módulo de Ethernet****Hoja: 7/9**

El módulo de Ethernet está basado en el circuito integrado ENC28J60 de Olimex, por lo que realmente se muestra en este circuito son las conexiones para incorporar este módulo. RidgeRun sugirió que el módulo de Ethernet fuese

adicional al sistema, pues la incorporación del mismo incrementaba el presupuesto de construcción de la estación de sistemas empotrados. Otra razón por la cual se deja el módulo de Ethernet como opcional, es que la empresa en un futuro puede decidir cambiar el módulo por otro de menor precio y de otro fabricante.

**Nombre: Amperímetro**

**Hoja : 8/9**

La medición de corriente se realiza mediante el empleo del circuito integrado ACS712ELCTR-20T de Allegro MicroSystems.

Este circuito integrado es un sensor de corriente de efecto HALL que tiene la capacidad de variar su tensión de salida en forma lineal respecto a la magnitud de la corriente que fluye a través de él.

La tensión de salida proporcional al valor de corriente que pasa a través de él, es sensada por el convertidor analógico- digital, del microcontrolador y este dato es convertido mediante una ecuación matemática en un valor de corriente similar al que se está sensando.

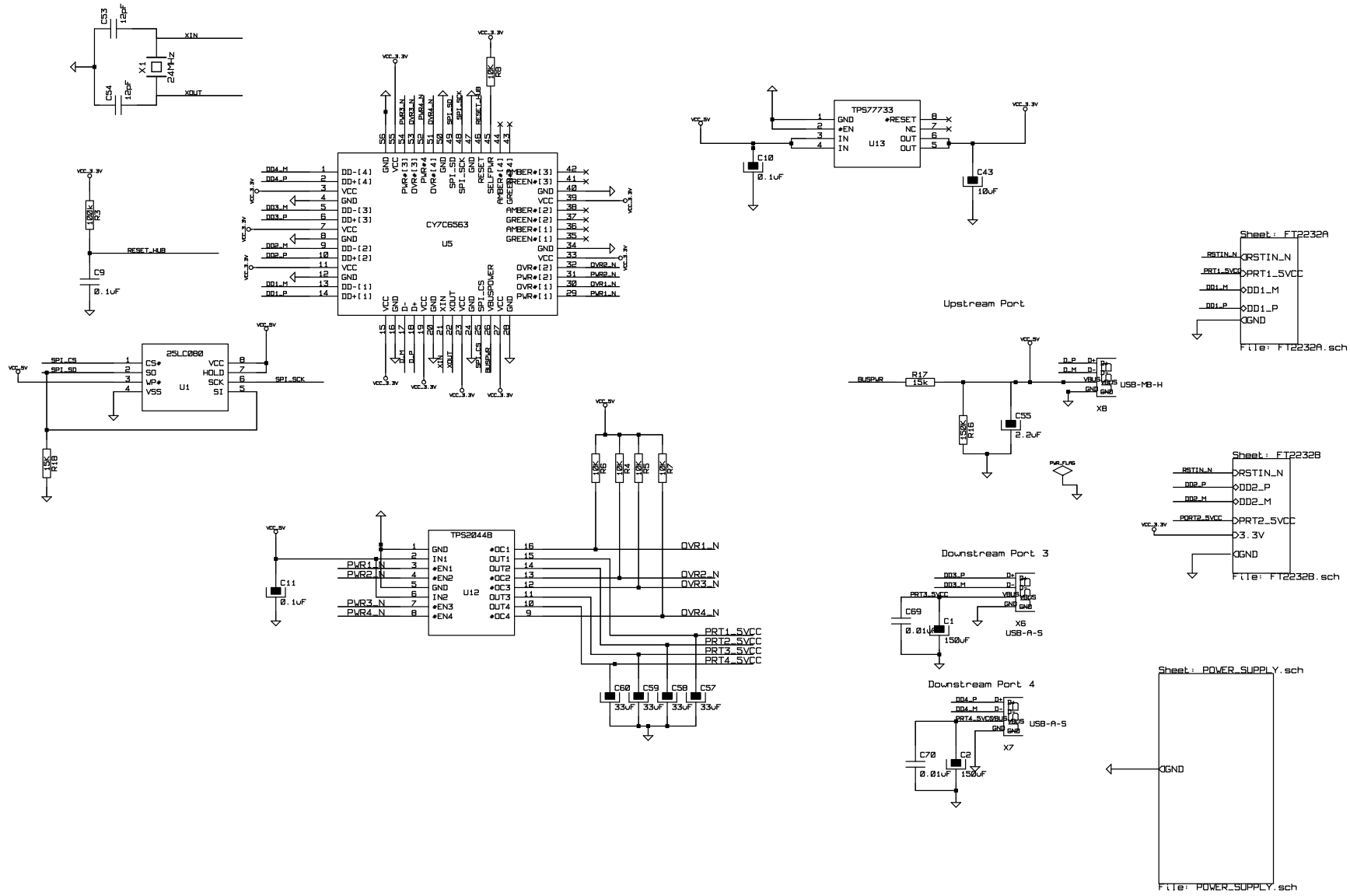
**Nombre: Voltímetro**

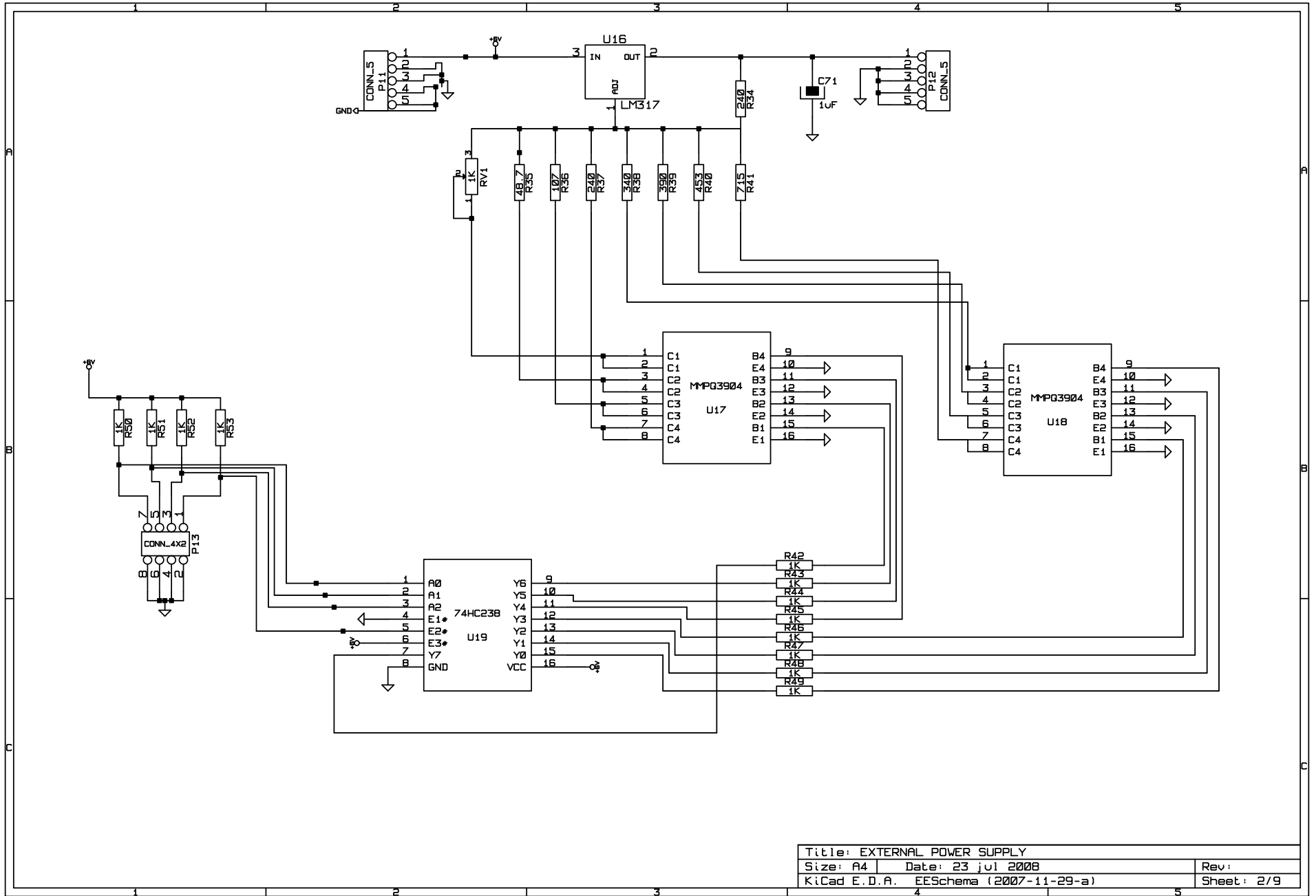
**Hoja: 9/9**

Dado que las tensiones que se pretenden medir se encuentran en el rango de los 0V a los 30V, y la tolerancia del microcontrolador en cuanto a las tensiones de entrada se encuentra entre los 0 y lo 3.3V, se emplea un circuito que funciona como divisor de tensión por un factor de 10, para que atenúe las señales de entrada y de esta forma el microcontrolador no sea dañado.

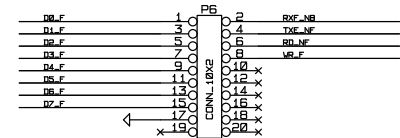
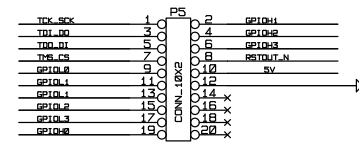
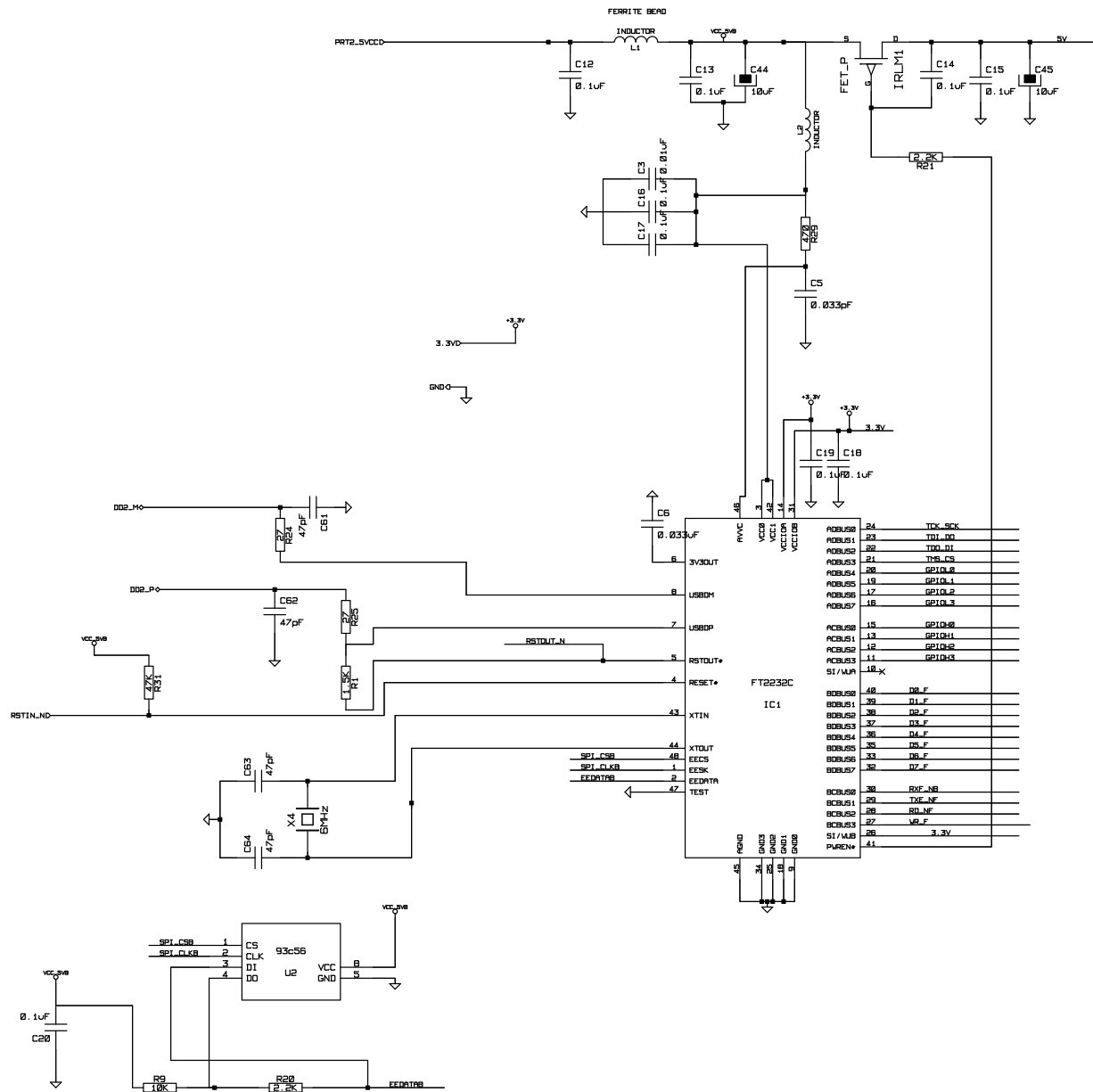
Se debe mencionar que el software desarrollado toma en cuenta el funcionamiento de éste módulo y realiza las correcciones pertinentes para mostrarle al usuario un resultado adecuado.

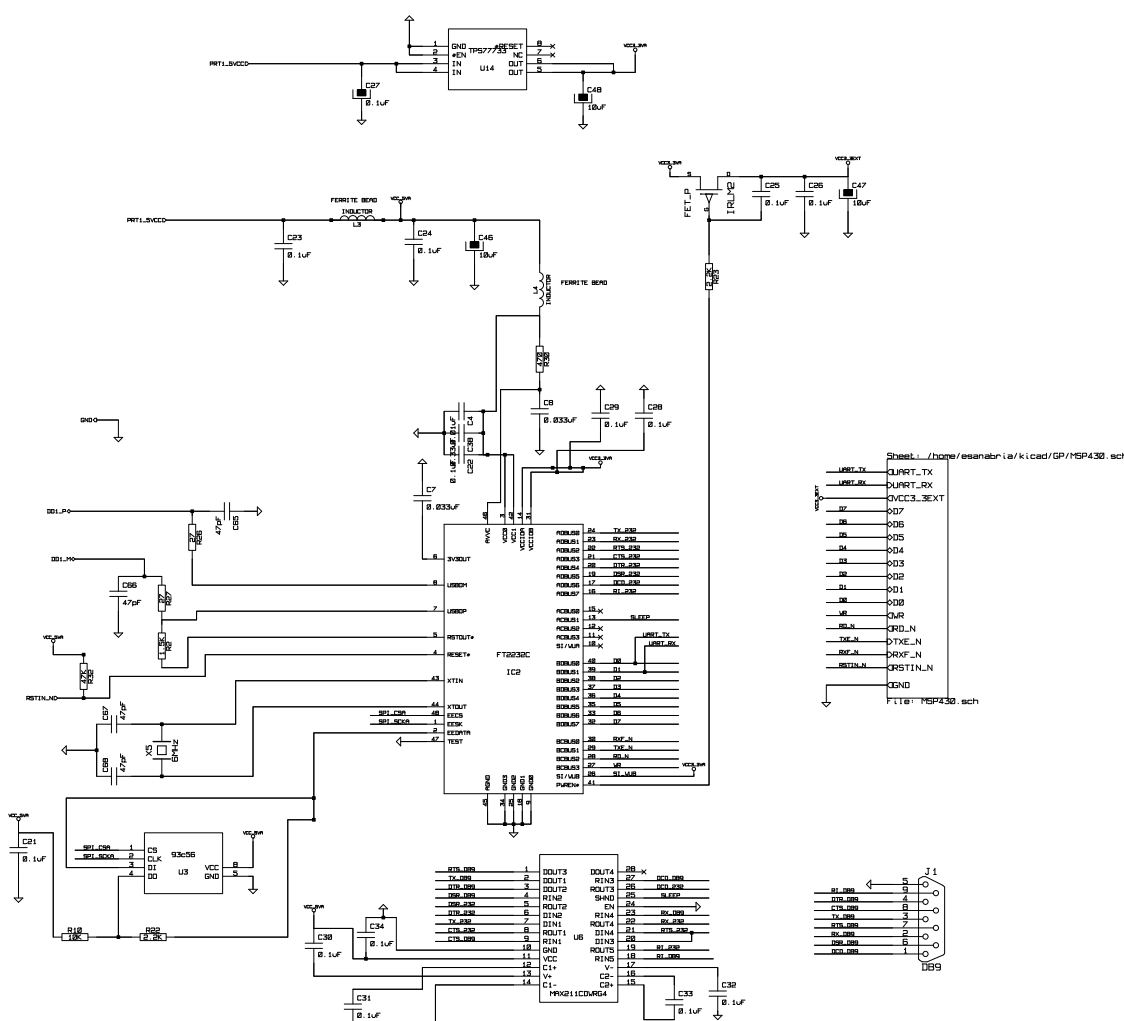






Title: EXTERNAL POWER SUPPLY		
Size: A4	Date: 23 jul 2008	Rev:
KiCad E.D.A. EESchema (2007-11-29-a)		Sheet: 2/9

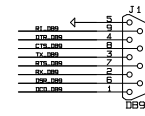


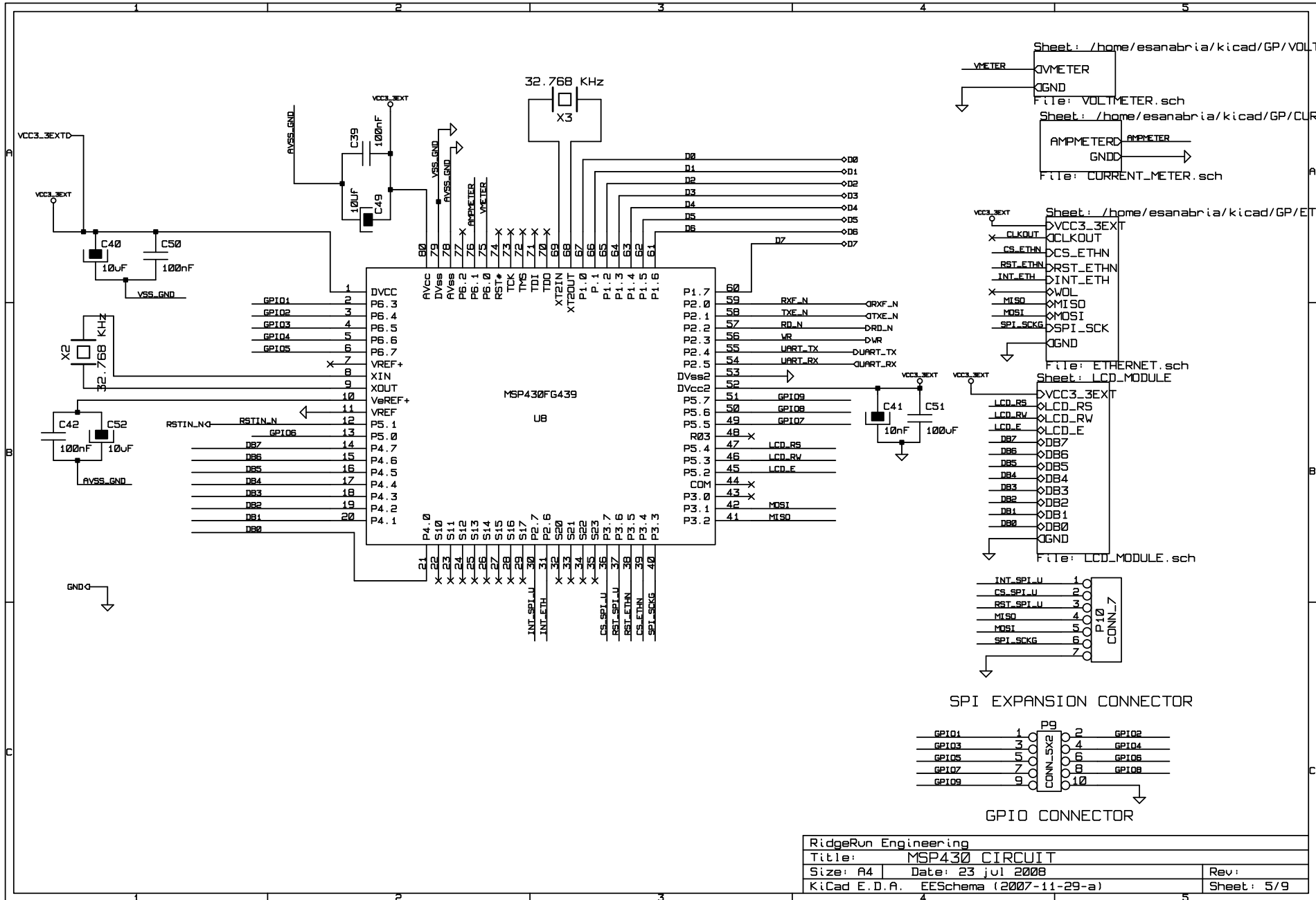


Sheet: /home/esanabr/ie/k/led/GP/MSP430.sch

UART_TX	UART_TX
UART_RX	UART_RX
VCC3_3EXT	VCC3_3EXT
D7	D7
D6	D6
D5	D5
D4	D4
D3	D3
D2	D2
D1	D1
D0	D0
CS	CS
RD_N	RD_N
TXE_N	TXE_N
DRX_N	DRX_N
RSTIN_N	RSTIN_N
GND	GND

FT232RL.MSP430.sch





Sheet: /home/esanabria/kicad/GP/VOLTMETER.

File: VOLTMETER.sch

Sheet: /home/esanabria/kicad/GP/CURRENT\_M

File: CURRENT\_METER.sch

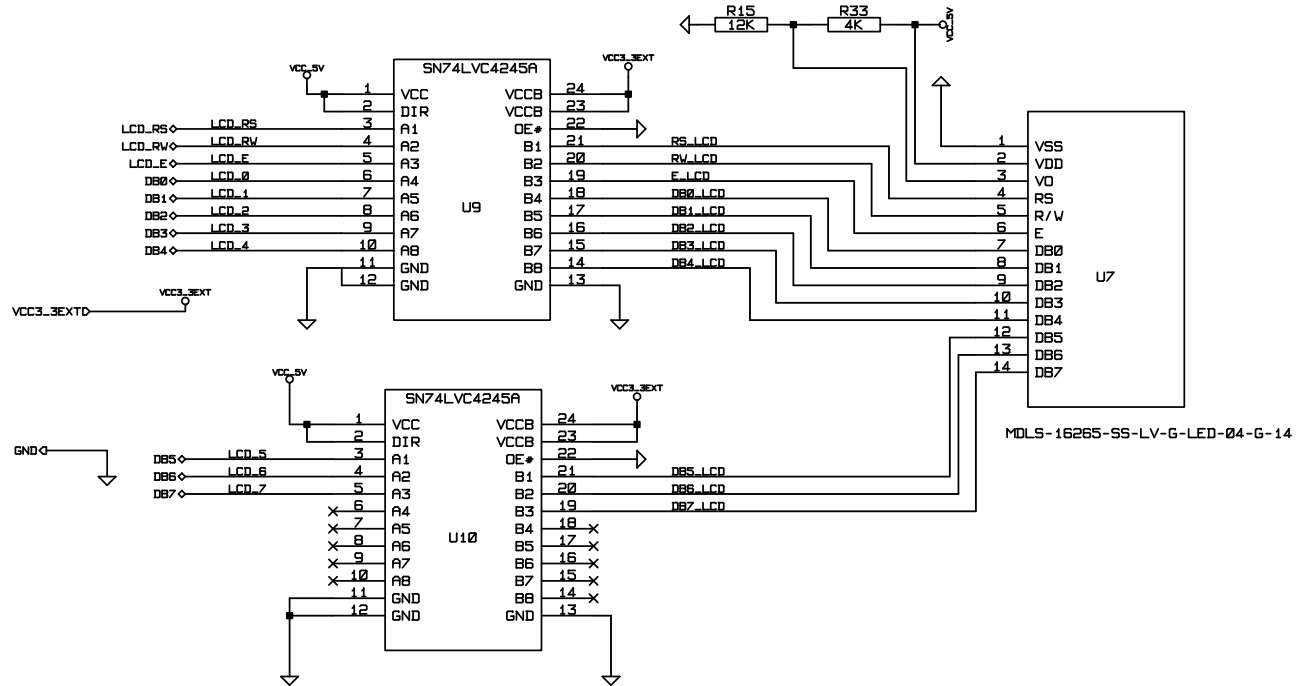
Sheet: /home/esanabria/kicad/GP/ETHERNET.

File: ETHERNET.sch

Sheet: LCD\_MODULE

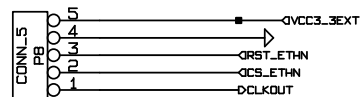
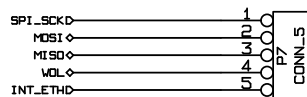
File: LCD\_MODULE.sch

RidgeRun Engineering			
Title: MSP430 CIRCUIT			
Size: A4	Date: 23 jul 2008	Rev:	
KiCad E.D.A.	EESchema (2007-11-29-a)	Sheet: 5/9	



RidgeRun Engineering		
Title: LCD CONNECTION		
Size: A4	Date: 23 Jul 2008	Rev:
KiCad E.D.A. EESchema (2007-11-29-a)		Sheet: 6/9

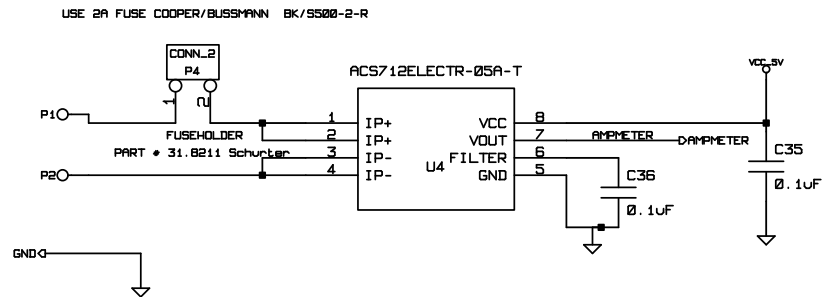
GND



SAMTEC CONNECTORS  
M16-105-02-T-5H

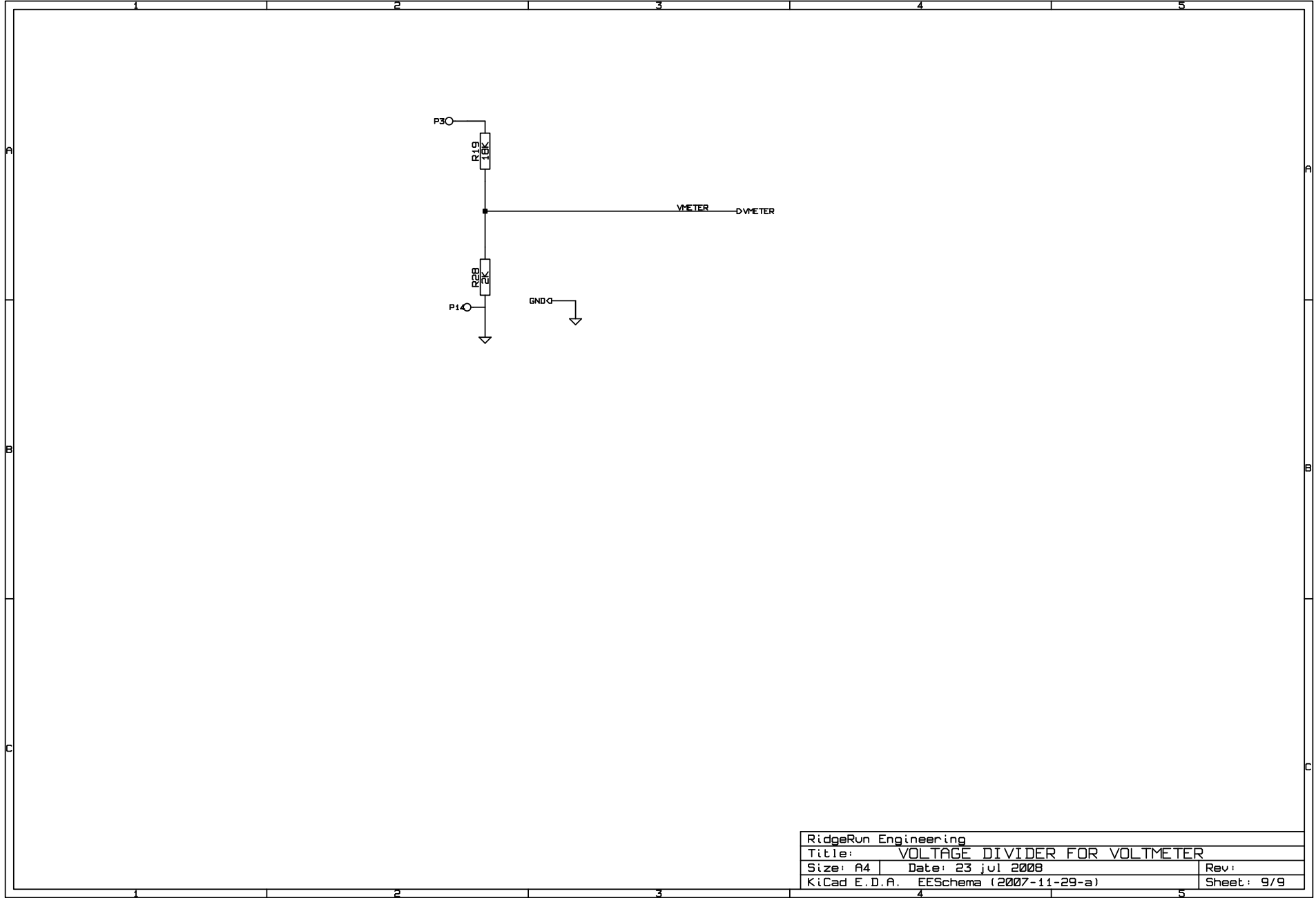
CONNECT SPI ETHERNET MODULE : OLIMEX ENC28J80

RidgeRun Engineering		
Title: CONNECTION FOR SPI MODULE		
Size: A4	Date: 23 jul 2008	Rev:
KiCad E.D.A. EESchema (2007-11-29-a)		Sheet: 7/9



RidgeRun Engineering		
Title: CURRENT SENSOR		
Size: A4	Date: 23 jul 2008	Rev:
KiCad E.D.A.	EESchema (2007-11-29-a)	Sheet: 8/9



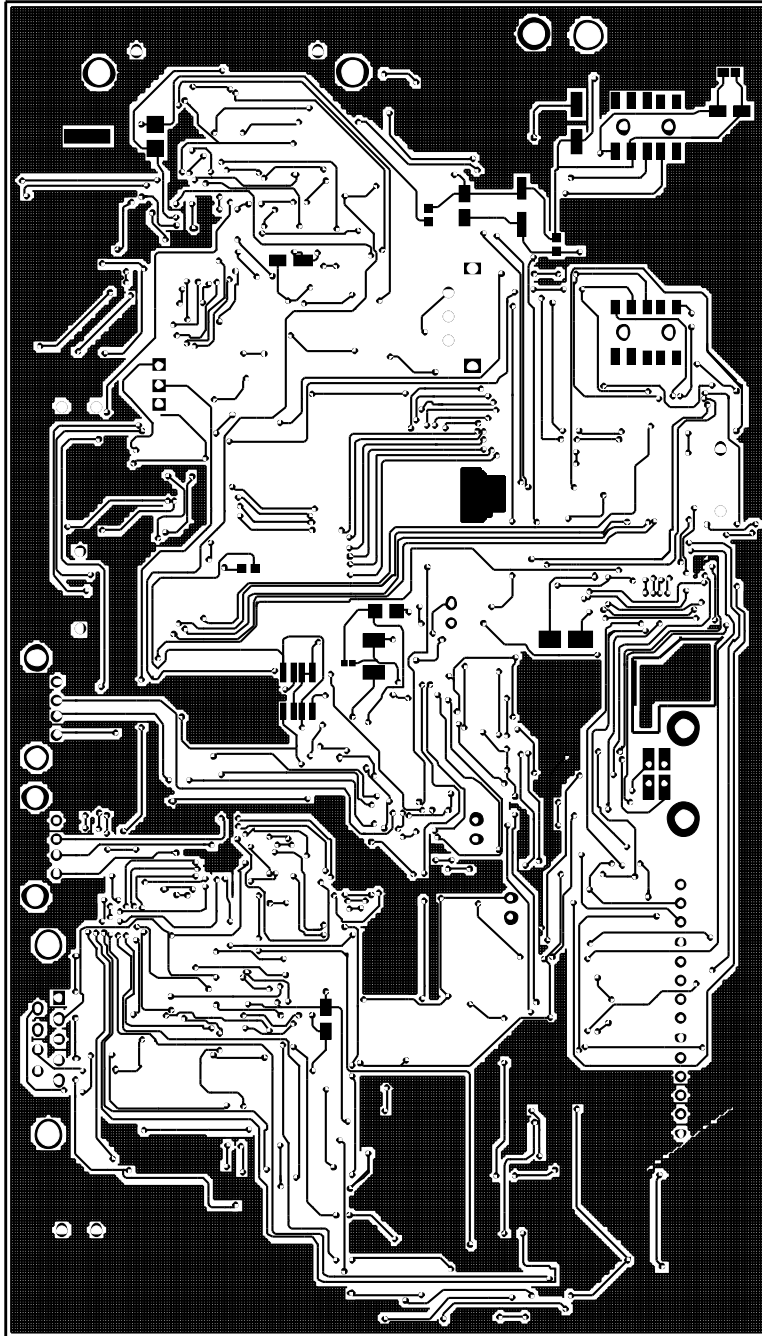


RidgeRun Engineering		
Title: VOLTAGE DIVIDER FOR VOLTMETER		
Size: A4	Date: 23 jul 2008	Rev:
KiCad E.D.A.	EESchema (2007-11-29-a)	Sheet: 9/9

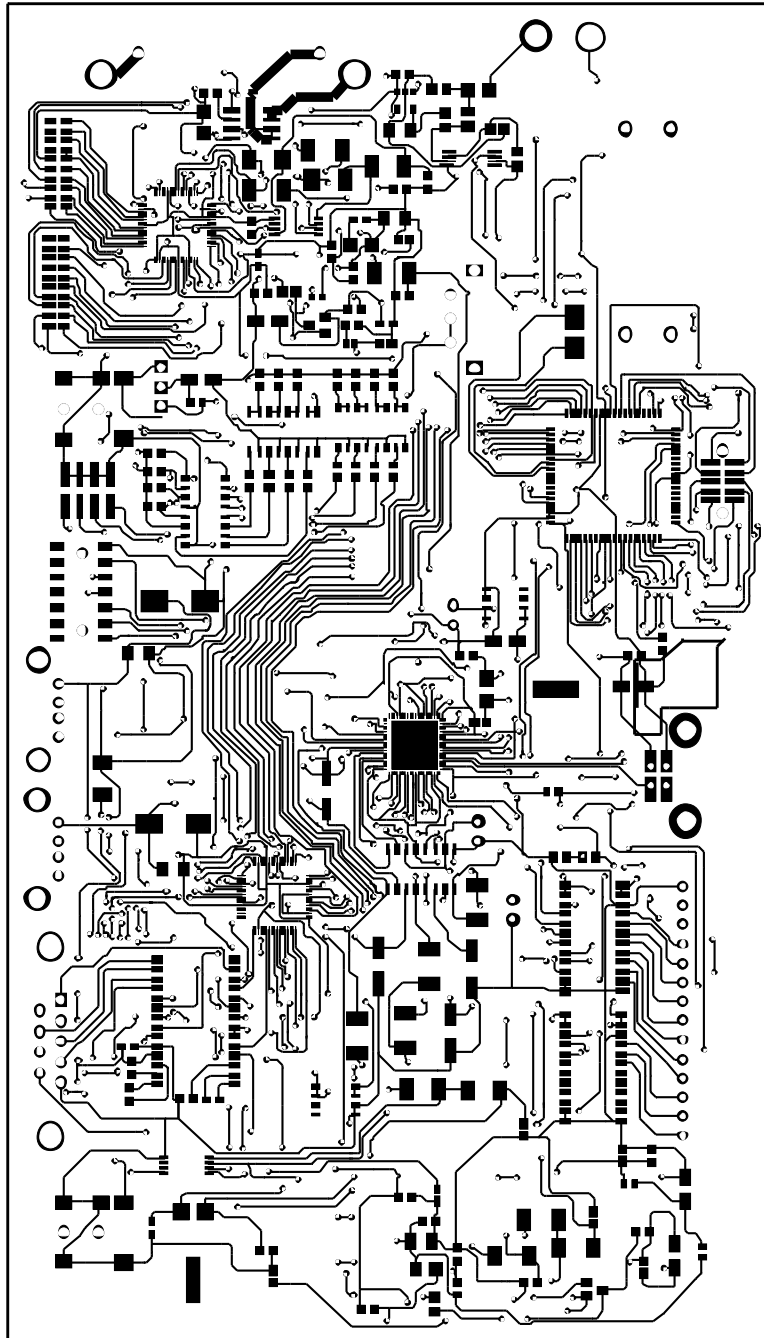
Las imágenes que se muestran a continuación corresponden las capas del circuito impreso.

Nota: El orden en que son mostradas estas imágenes corresponde a:

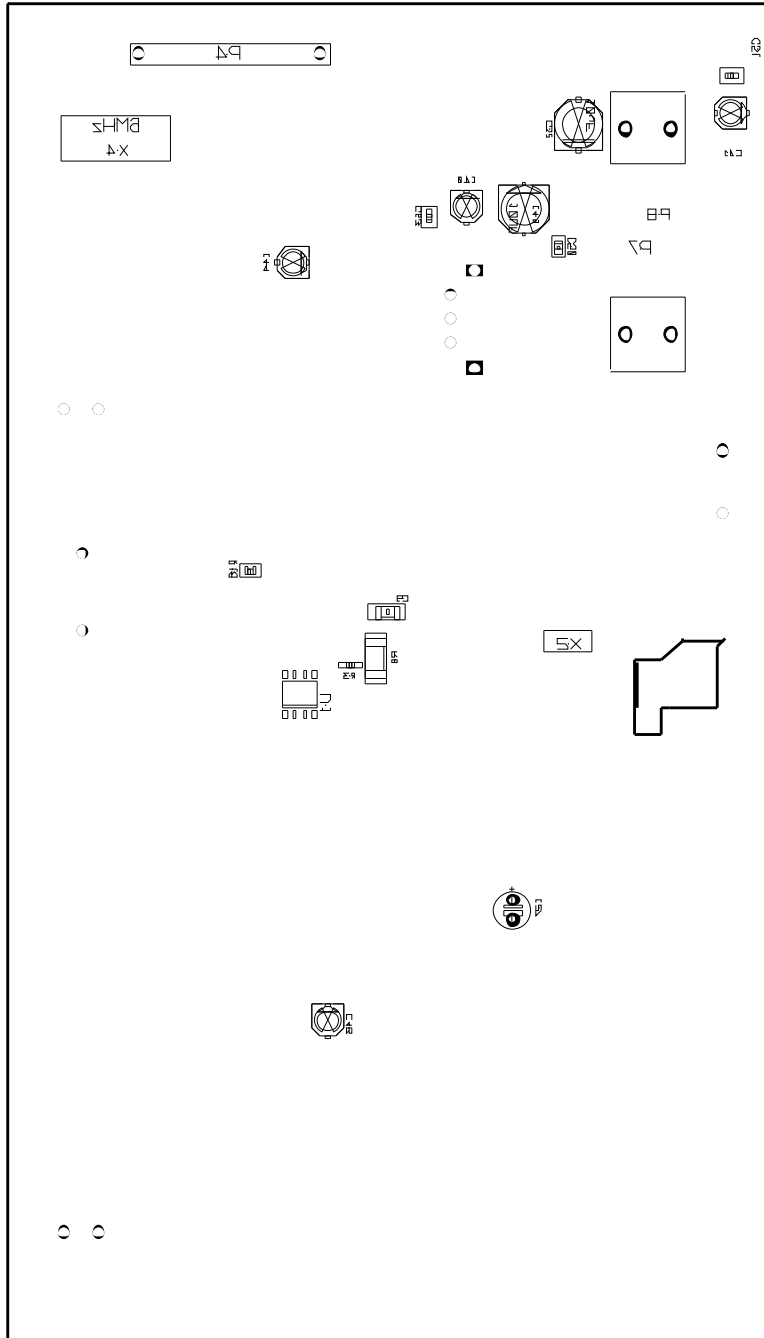
1. Capa de Cobre
2. Capa de Pistas
3. Silkscreen superior o capa de guía para colocar componentes en la parte superior
4. Silkscreen inferior o capa de guía para colocar componentes en la parte inferior



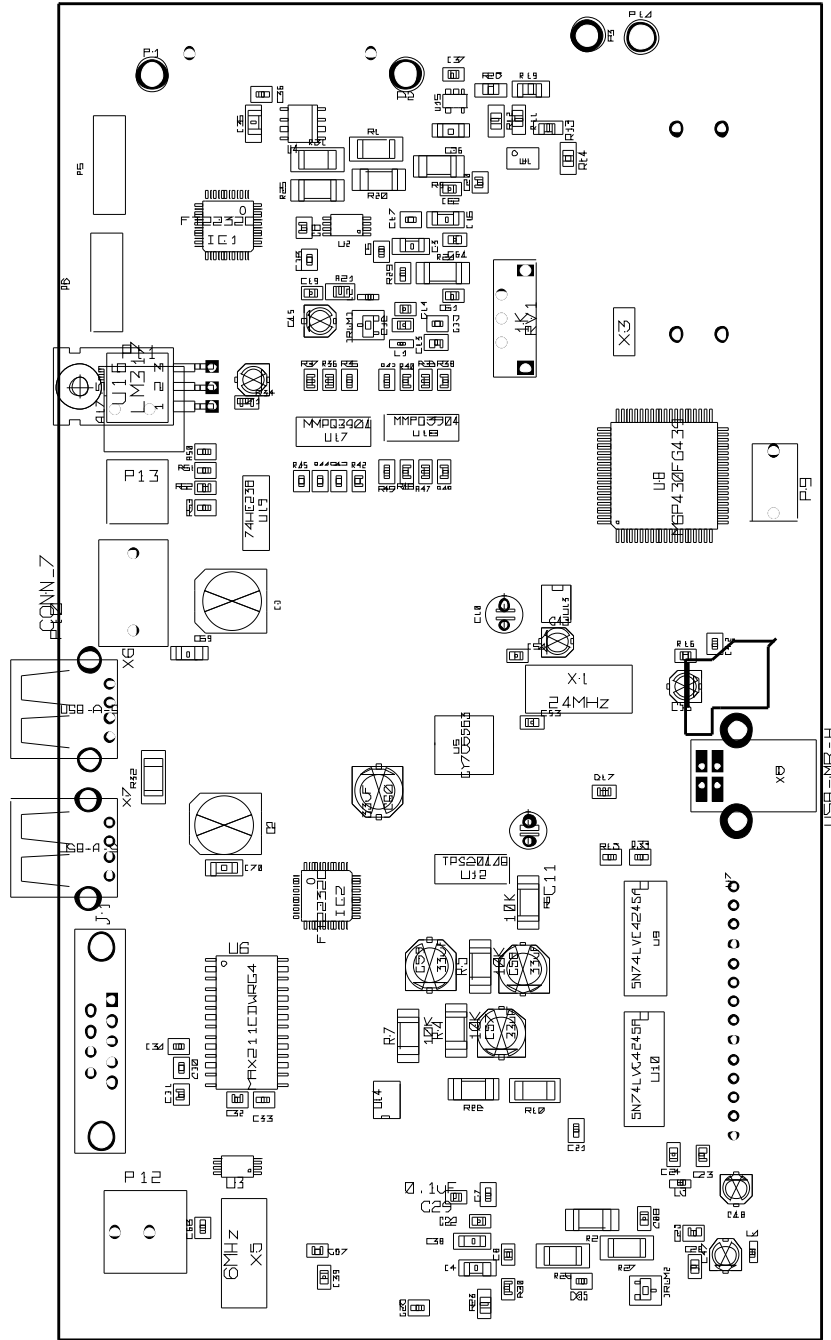
RIOGERUN	
Title: GP	
Size: user	Date: 23 ju) 2008
KiCad E.O.P. PCBNEW 12007-11-29-a1	Rev: 1
	Sheet: 1/1



RIOGERUN	
Title: GP	
Size: user	Date: 23 ju) 2008
KiCad E.O.P. PCBNEW 12007-11-29-a1	Rev: 1
	Sheet: 1/1



RIOGERUN		
Title: GP		
Size: user	Date: 23 ju 2008	Rev: 1
KiCad E.O.P.	PCBNEW 12007-11-29-a1	Sheet: 1/1



RIOGERUN	
Title: GP	
Size: USB	Date: 23 ju) 2008
KiCad E.O.P. PCBNEW 12007-11-29-a1	Rev: 1
	Sheet: 1/1