

# **DISEÑO E IMPLEMENTACIÓN DE UNA TARJETA DE CONTROL REPROGRAMABLE DESDE UN PC.**

Ronald Ponguillo Intriago<sup>1</sup> , Ludmila Gorenkova Labicova<sup>2</sup>

<sup>1</sup> Ingeniero Eléctrico en Electrónica 2003

<sup>2</sup> Director de Tesis. MSc. Telecomunicaciones, Universidad Electrotécnica de Telecomunicaciones, San Petersburg Rusia 1978. Profesora de ESPOL desde 1979

## **RESUMEN**

En el capítulo 1 se hace una introducción a lo que es el proyecto, mostrando el esquema de funcionamiento, donde se detalla las características de cada etapa. El capítulo 2 nos introduce en el diseño del hardware. El diseño del firmware se detalla en el capítulo 3, haciendo una exposición general de los requerimientos del programa de control, para luego pasar al detalle de cada subrutina que componen al mismo. El diseño del software se lo analiza en el capítulo 4, donde se justifica la elección del paquete de programación que se empleó. Posteriormente en el capítulo 5 se encuentran algunas fotos que muestran el trabajo final, también se muestran algunos resultados de la simulación del microcontrolador y de las pruebas hechas al software.

Al final se encontrará varios apéndices que dan algunos detalles de las herramientas utilizadas, como el formato hexadecimal de Intel en el apéndice A, la tarjeta de circuito impreso en el apéndice B y las características del microcontrolador utilizado en el apéndice C, dejando el manual del usuario en el apéndice D. Finalmente se muestra un glosario y alguna bibliografía que sirvió de consulta para este trabajo.

## **INTRODUCCIÓN**

Con el avance tecnológico han surgido variedades de circuitos integrados con capacidad de programación. Esta es la tendencia del mundo moderno en el que casi todo lo que existe es digital y casi todo lo digital es programable.

Entre los circuitos integrados programables uno que se ha destacado en los últimos años por su flexibilidad a la hora de usarlo es el microcontrolador, ya que lleva integrado un sinnúmero de recursos que reduce el tiempo de trabajo para el diseñador y el tamaño de los prototipos y productos finales. Su antecesor, el microprocesador, ha tenido su auge en las computadoras personales que han evolucionando con una velocidad asombrosa. Por otro lado el desarrollo del microcontrolador se debe en gran parte a la necesidad de comodidad del ser humano, que día a día requiere de cosas que le faciliten la vida y que la tecnología le puede ofrecer gracias a los sistemas automáticos controlados electrónicamente. Para facilitar el desarrollo de estos sistemas automáticos la electrónica moderna diseña y construye dispositivos semiconductores con capacidad de programación y reprogramación junto con las herramientas necesarias para en conjunto elaborar una solución específica.

Si tuviésemos que enumerar las diferentes aplicaciones que se puede dar a los microcontroladores nos tomaría mucho tiempo debido a que su campo de aplicación es bastante amplio y variado. Además como aquí si cabe mencionar que el límite lo pone la imaginación, solamente diré que casi en todas las casas hoy en día se encuentra microcontroladores, por ejemplo en los equipos de audio y video, en los productos de línea blanca, en los automóviles, en las puertas de garaje automáticas, en controles electrónicos de encendido apagado de bombas de agua, solo por citar algo.

Cuando se habla de microcontroladores es posible mencionar un sinnúmero de fabricantes como Intel, AMD, Motorola, Texas Instruments y no se puede dejar de mencionar a Microchip, que es una de las empresas que mas microcontroladores vende en el mundo anualmente, y su éxito lo debe en gran parte a que entrega gratuitamente toda la información necesaria para que las personas interesadas aprendan a utilizar sus productos. Además de dar mucha información con los detalles de los recursos internos y su programación, sus microcontroladores son muy fáciles de programar ya que basados en una arquitectura RISC poseen muy pocas instrucciones.

En la realidad de nuestro medio se hace cada vez imprescindible contar con elementos que automaticen ciertos procesos que nos generen además de cierta comodidad una reducción en el consumo de energía, como por ejemplo un sistema que controle el encendido y apagado de las luces, o el funcionamiento de una bomba que llene un tanque elevado. Estas son tareas de automatización que no demandan una gran inversión y en cambio nos podrían generar un gran beneficio.

En este proyecto de tesis se presenta un prototipo bastante versátil que intenta ser una alternativa para las situaciones antes mencionadas, en el mismo que se podrá generar diversas soluciones a problemas sencillos pero de mucha utilidad y que además podrán ser generadas por personas que no necesitarán ser unos peritos en la materia, sino, tener solo unas cuantas nociones básicas al respecto.

La verdad del caso es que no se puede hacer una comparación en cuanto a prestaciones entre esta tarjeta y un PLC, ya que el PLC es mucho mas completo, pero para el tipo de aplicaciones que se pueden desarrollar con la tarjeta presentada en ésta tesis, resulta mas conveniente usar esta última por las siguientes razones:

El PLC más pequeño y básico no deja de costar alrededor de cien dólares, mientras que el costo de armar la tarjeta basada en microcontroladores, como se puede ver en el capítulo 5 no sobrepasa los sesenta y cinco dólares.

Por otro lado el PLC posee su software para programación de las aplicaciones de usuario al igual que lo hace el software TESIS\_RAPI con la tarjeta hecha con microcontrolador, pero la diferencia de costos es enorme, ya que el software TESIS\_RAPI por ser parte de una tesis no tiene precio y en cambio un software para programar PLC podría estar costando hasta varios miles de dólares, inversión que no se justifica para desarrollar aplicaciones pequeñas como las planteadas aquí.

## **CONTENIDO**

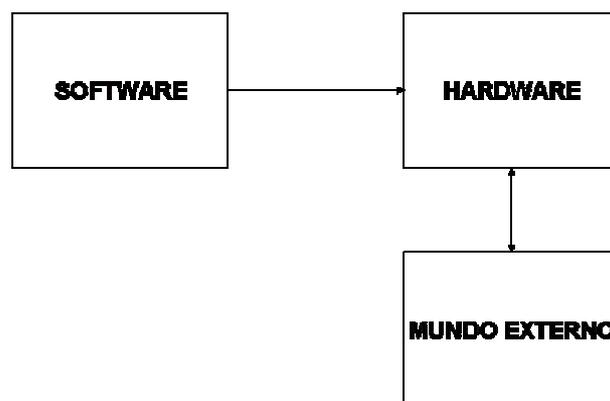
### **ESQUEMA DE FUNCIONAMIENTO**

Básicamente lo que se quiere es un dispositivo que sea capaz de realizar procesos de encendido/apagado de acuerdo a una secuencia programada que puede estar en función de los estados que tengan las entradas en determinado momento. Además se debe tener la capacidad de cambiar el programa directamente desde una computadora personal sin ayuda de un programador especializado adicional.

Un programa de computadora debe realizar las tareas de edición, compilación y transferencia de datos de los programas de usuario hasta la tarjeta prototipo.

Dentro de las posibles aplicaciones que se dará a este proyecto podrían estar incluidos cualquier proceso de control que demande una cantidad de cuatro entradas digitales del tipo todo o nada, ocho salida con relés que pueden manejar una corriente nominal de 5 amperios, temporización de hasta 255 segundos con pasos de un segundo

### **DIAGRAMA DE BLOQUES DEL SISTEMA**



**Figura 1 Diagrama de Bloques del Sistema**

El diagrama de bloques mostrado representa cada una de las etapas que comprenden el proyecto, y como se puede ver consta de tres bloques que se detallan a continuación.

El bloque del software lo constituye un programa para PC que corre bajo el sistema operativo Windows y tiene un ambiente visual agradable que mantiene muchas de las características de los programas que funcionan bajo Windows, como los menús comunes para abrir o guardar archivos que el usuario requiera para su trabajo, también una barra con las herramientas más usadas y las opciones para salir del programa.

El programa es capaz de comunicarse con el hardware para transferir hasta el microcontrolador el código necesario para realizar la tarea que estime el usuario; estas tareas pueden ser codificadas dentro del editor que posee el software y compiladas en el mismo entorno.

En lo que respecta al hardware se implementó el firmware de tal manera que posee funciones básicas como una rutina que le permite al microcontrolador comunicarse serialmente con la PC para poder recibir los datos que requieren ser grabados en la memoria de programa. Además tiene la capacidad de hacer un control en tiempo real que le permite temporizar eventos.

Otra de las características que se consideraron, fue darle la capacidad de poder escribir en la memoria de programa, tomando en cuenta que no se debe sobrescribir en la región donde residen las funciones del firmware, para no provocar conflictos en el funcionamiento.

En el hardware se colocaron también las respectivas interfaces para acoplar los niveles de voltaje RS232 de la PC con los del microcontrolador, por otro lado, se estableció la circuitería para manejar cada relé en las salidas.

## **CRITERIOS PARA LA SELECCIÓN DEL PIC**

Para poder elegir el PIC que se iba a utilizar se debió tomar en consideración varios aspectos, basados en los requerimientos de recursos, consideraciones

de precio versus prestaciones, disponibilidad en el mercado y proyecciones para futuras ampliaciones. A continuación hago un bosquejo de los principales criterios para esta selección:

1. Se necesitan 12 pines de E/S, 4 para entradas y 8 para salidas.
2. Requerimos 2 pines para comunicación serial con el PC, uno para transmisión y el otro para recepción.
3. Se debe tener la capacidad para generar temporización de un segundo manejada con interrupciones.
4. Es indispensable que el dispositivo tenga memoria de programa tipo FLASH o EEPROM, para poder grabar y borrar desde el software.
5. Debe poseer suficiente memoria para programa y datos

En el primer punto se menciona que la tarjeta tiene cuatro entradas y ocho salidas, para ello necesito 12 pines.

Para poder comunicar la tarjeta con la PC serialmente a través del puerto RS232 de éste último, necesito bien sea emular su protocolo en el firmware del PIC o utilizar un dispositivo que establezca dicha comunicación serial. Este elemento es un USART (Universal Synchronous Asynchronous Receiver Transmitter – Receptor Transmisor Sincrónico Asincrónico Universal), y es mas conveniente utilizar el recurso en hardware que emular su funcionamiento por dos razones fundamentales, la primera es que realizar estas funciones con código es más complicado y la segunda y mas importante es que dicho código crearía dependencia con las subrutinas a las que les da servicio y ello conlleva a consumo innecesario de ciclos de instrucción, lo que haría mas lento y menos eficiente el proceso.

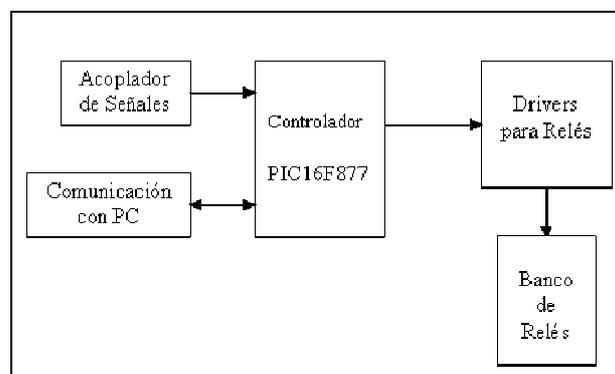
En el tercer punto se analiza la capacidad de generar temporizaciones de un segundo lo que sería completamente ineficiente hacerlo a base de retardos usando bucles dentro del programa, además esto no tendría la precisión que puede darnos un temporizador integrado que se pueda manejar con interrupciones.

La base del proyecto es la capacidad que tiene este para reprogramarse sin necesidad de un hardware de programación, para lo cual el PIC seleccionado debe tener un tipo de memoria de programa que pueda ser escrito y borrado con un código implementado en la misma. Esta capacidad la ofrecen las memorias FLASH y EEPROM, esto es lo que se establece en el punto cuatro. Por último en el punto cinco se plantea la necesidad de disponer de suficiente memoria para las aplicaciones del usuario. A partir de todas éstas características se ha elegido al PIC16F877, considerando además que se encuentra con mucha facilidad en el mercado local.

### DIAGRAMA DE BLOQUES DE LA TARJETA

En el siguiente diagrama de bloques se muestra cada una de las partes que constituyen el hardware del proyecto. Como bloque central está el controlador que ha sido implementado con el PIC16F877, en el que se han incluido subrutinas para el manejo de las entradas digitales, otra para el manejo de la comunicación serial con la PC. Se incluye también el código que permite manipular las salidas y alguna rutina que da la opción de producir temporizaciones desde un segundo hasta 255 segundos, para aplicaciones en las que se requiera manejar intervalos de tiempo.

El bloque Acoplador de Señales corresponde a cuatro entradas digitales implementadas sobre el puerto A del microcontrolador. La comunicación con la PC se logra a través del USART integrado en el PIC y un circuito integrado MAX232 debido a que las señales del microcontrolador no son compatibles con los niveles que maneja la PC. Para manejar las salidas de la tarjeta se ha dispuesto de un circuito integrado ULN2803.



**Fig. 2 Diagrama de Bloques de la Tarjeta**

## 5.2.2 PRUEBAS HECHAS EN EL SOFTWARE

Con lo que respecta al software se lo ha instalado y corrido en varias máquinas con diferentes versiones de Windows. Estas pruebas se resumen en la siguiente tabla.

**Tabla I Resumen de las Pruebas hechas al Software**

<b>CARACTERISTICAS DEL PROCESADOR</b>	<b>SISTEMA OPERATIVO</b>	<b>OBSERVACIONES</b>
AMD Atlon XP 2GHz	Windows XP	Funcionamiento OK condicionado <sup>a</sup>
Intel Pentium MMX 166MHz	Windows Me	Funcionamiento OK
Intel Pentium III 448MHz	Windows XP	Funcionamiento OK condicionado <sup>a</sup>
Intel Pentium 4 2GHz	Windows XP	Funcionamiento OK condicionado <sup>a</sup>
Intel Pentium 4 1.7GHz	Windows Me	Funcionamiento OK
Intel Pentium 133MHz	Windows 98	Funcionamiento OK

<sup>a</sup> Cuando se instala el programa TESIS\_RAPI, se copian algunas aplicaciones de Microchip como el MPASMWIN, MPLINK,\_MPLINK, MP2COD, MP2HEX en la carpeta C:\ASM\_RAPI. Estas aplicaciones fueron desarrolladas para correr bajo el sistema operativo MS-DOS, por lo tanto tiene restricciones al momento de correr en Windows XP. Así, si el administrador no ha habilitado la compatibilidad con versiones anteriores de Windows, corresponde al usuario hacerlo.

Es muy sencillo, si se instala TESIS\_RAPI sobre Windows XP, ubíquese en la carpeta C:\ASM\_RAPI y seleccione los archivos mencionados arriba; entonces haga clic con el botón derecho del ratón y habilite la opción compatibilidad con Windows 98, Windows Me.

## CONCLUSIONES Y RECOMENDACIONES

A pesar que el proyecto fue concebido y construido originalmente para trabajar con entradas digitales, existe la posibilidad de implementar entradas analógicas también. Este tema no ha sido considerado en el desarrollo de ésta tesis debido a que ya se había hecho la propuesta y el alcance del mismo. En el transcurso de la realización del proyecto recibí varias sugerencias acerca de si se podía implementar dicho tipo de entradas en el prototipo ya construido, sin necesidad de elaborar otra tarjeta con las modificaciones del caso, así que emprendí el análisis de esta situación.

Después de analizar el caso pude llegar a la siguiente conclusión:

Como el PIC que se ha utilizado en este trabajo posee un convertidor analógico a digital con ocho canales, y además las cuatro entradas digitales que se implementaron coinciden con cuatro de los ocho canales analógicos, si se puede implementar también entradas analógicas en la tarjeta.

Cabe indicar que como es obvio que si se usa un pin determinado para entrada digital, este no podrá usarse como entrada analógica y viceversa; a menos que se implemente la subrutina y el circuito de multiplexación adecuados. De esta manera se puede tener hasta cuatro entradas analógicas si se programa la opción de usar referencia de voltaje interna.

Otra cosa que hay que tener en cuenta es que en el firmware del microcontrolador no se ha implementado la subrutina necesaria para darle esta capacidad a la tarjeta, ni el software tiene la posibilidad de recibir e interpretar los datos digitalizados. Pero aún así el usuario podría tranquilamente hacer alguna aplicación que implique la conversión de datos de analógico a digital y visualización por algún medio que podría ser un monitor de computadora.

Otra sugerencia que yo haría a todos aquellos lectores de este trabajo es que le saquen provecho a cada una de las subrutinas elaboradas aquí y sobre todo se fijan en la forma utilizada para programar el PIC, que presenta cada subrutina para manipulación de recursos en un archivo ASM separado y que

es mucho mas eficiente a la hora de hacer una depuración que escribir todo el código sobre un solo archivo. También se podría intentar hacer que el proyecto corra en otras plataformas como por ejemplo Linux, haciendo una versión del software en Java, si no se quiere programar en C, o en otra aplicación dedicada para entorno Linux. En lo que respecta al hardware, éste es un buen punto de partida para desarrollar una tarjeta que tenga capacidad de trabajar en red, sea esta de tipo industrial o por que no hacerla interactuar con Internet.

## **BIBLIOGRAFÍA**

1. R. Ponguillo, "Diseño e Implementación de una Tarjeta Inteligente basada en Microcontroladores, Reprogramable desde un PC por medio de un Software hecho en Visual Basic". (Tesis, Facultad de Ingeniería Eléctrica, Escuela Superior Politécnica del Litoral, 2003)
2. J. Ma. Angulo, Microcontroladores PIC. Diseño Práctico de Aplicaciones (McGRAW-HILL, 1998), pp.93-150
3. L. Suárez, Visual Basic - Guía del Estudiante.( San Sebastián de los Reyes, Madrid, 1998), pp. 87-94