

“Analizador Lógico con Cuatro Señales de Entrada realizado mediante Microcontroladores Avanzados y GLCD”

Autores:

Christian Jhonny Alvarado Falcones
Juan Pablo Orellana Valarezo

Director:

Ing. Carlos Valdivieso

Facultad de Ingeniería en el Electrónica y Telecomunicaciones
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-863. Guayaquil-Ecuador

Emails:

cjalvara@espol.edu.ec
jporellana@espol.edu.ec
cvaldivi@espol.edu.ec

RESUMEN

El presente trabajo corresponde al desarrollo de un analizador lógico, este fue construido por la implementación de rutinas especiales usando como base el PIC 16F887 junto a un dispositivo gráfico, este dispositivo permite mostrar en pantalla cuatro señales lógicas; estas señales ingresan con diferentes frecuencias. Para poder desarrollar este proyecto se debe utilizar el programa MikroC Pro for Pic, el cual nos permitió programar en C todas las funciones que debe desempeñar nuestro microcontrolador; funciones de conversión analógico-digital, para temporizadores de interrupciones y funciones para poder mostrar en la GLCD.

Los analizadores lógicos son muy parecidos a los osciloscopios ya que estos muestran señales en pantalla con la diferencia de que un analizador lógico permitirá mostrar señales digitales permitiendo el análisis y comparación de las frecuencias presentes en las señales que ingresan.

Palabras Claves: Analizador lógico, microcontrolador, convertidores A/D, interrupciones, GLCD.

Abstract

This work corresponds to the development of a logic analyzer, that was built by the implementation of special routines using a 16f887 microcontroller with a graphics device, this device allowed us to show on screen four logic signals, these signals may have different frequencies. MikroC Pro for Pic was used in the development of this project, the program was written in C language divided in functions such as: analog-digital converters, Timer interrupts and functions to display on the GLCD.

The Logic analyzers are very similar to the oscilloscopes showing signals on the screen with the difference that a logic analyzer will show only digital signals allowing frequency analysis and comparison of incoming signals .

Keywords: logical analyzer, microcontroller, ADC, interruptions, GLCD.

1. Antecedentes

Los Analizadores Lógicos se desarrollaron casi simultáneamente al mismo tiempo que los primeros microcontroladores que salieron al mercado. Los ingenieros que diseñaban sistemas basados en estos nuevos dispositivos pronto notaron que para la implementación de diseños con microcontroladores y chips digitales hacía falta una herramienta para visualizar más entradas de las que podían ofrecer los osciloscopios.

Los analizadores lógicos, con sus distintas entradas, eran la solución perfecta a este problema, dado que varias señales podían ser analizadas al mismo tiempo a diferencia del osciloscopio. Estos instrumentos han aumentado gradualmente tanto su velocidad de adquisición como el recuento de canales para mantenerse a la altura de los rápidos avances en la tecnología digital. El analizador lógico es una herramienta clave para el desarrollo de los sistemas digitales, aunque existen similitudes entre los osciloscopios y los analizadores lógicos [1].

2. Teoría

El desarrollo del proyecto se lo puede dividir en Software y Hardware. El software para el programa del PIC 16F887 es MIKRO C PRO FOR PIC de MIKROELECTRONICA y a su vez el software para cargar este programa en el PIC es el programa PICKIT 3, que permite cargar el archivo en hexadecimal que fue generado por MIKRO C al PIC desde un puerto USB de la PC.

Los componentes para desarrollar el proyecto, es decir el hardware usado son principalmente el PIC 16f887 que tiene 40 pines y además la pantalla táctil GLCD que nos permitirá visualizar las señales digitales [4].

2.1 Herramientas de Software

MikroC PRO para PIC es una herramienta muy útil y con todas las características para microcontroladores PIC de Microchip. Está diseñado para desarrollar, construir y depurar aplicaciones embebidas basadas PIC. Este entorno de desarrollo cuenta con una amplia variedad de características tales como IDE fácil de usar, muy compacta y eficiente código, hardware y software de bibliotecas, documentación completa, simulador de software, soporte de hardware depurador [3].

Proteus este software ofrece una amplia gama de prestaciones al realizar diferentes tipos de proyectos

electrónicos, en todas sus etapas: diseño, simulación, depuración y construcción, ya sea analógica, digital o combinada, usando diferentes componentes electrónicos, como también puertos existentes en el mercado, herramientas de medición como multímetros, osciloscopios, analizadores lógicos y demás.

Esta herramienta nos permite simular y depurar el funcionamiento de todo el sistema paso a paso, de tal manera que podremos observar el contenido de registros y diferentes posiciones de memoria (en circuitos digitales), que nos facilitara determinar si la respuesta del hardware es la correcta o en su defecto existen errores.

2.2 Herramientas de hardware

El pic 16f887 tiene las siguientes características:

- DIP 40 pines.
- Memoria FLASH para programa: 14KB
- Memoria RAM para datos: 368 bytes.
- Memoria EEPROM para datos: 256 bytes.
- Capacidad de Interrupciones.
- Stack de 8 niveles.
- Oscilador interno de 8 a 31 MHz.
- Modos de direccionamiento directo, indirecto y relativo.
- Hasta 36 pines de entrada/salida.
- Convertidor A/D de 14 canales, 10 bit.
- Tres timers/counters (8, 16 y 8 bits).
- 1 Módulo CCP (capture, compare y PWM).
- EUSART/SCI.
- In circuit serial programming (ICSP).
- Power On Reset, Power Up Timer, Oscillator Start-up Timer.
- Watchdog Timer, Brown-out detect.
- Code protection.
- Modo SLEEP para ahorro de energía[2].

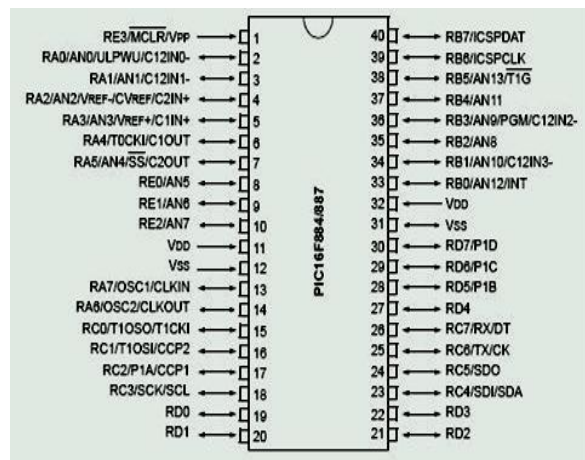


Figura 1: Configuración de pines PIC 16F887.

Las pantallas táctiles resistivas son más accesibles por su durabilidad ante elementos como polvo o agua, razón por la que son el tipo de pantallas más usado en la actualidad, aunque pero tienen una pérdida de aproximadamente el 25% del brillo debido a las múltiples capas necesarias. Otro inconveniente que tienen es que pueden ser dañadas por objetos afilados.

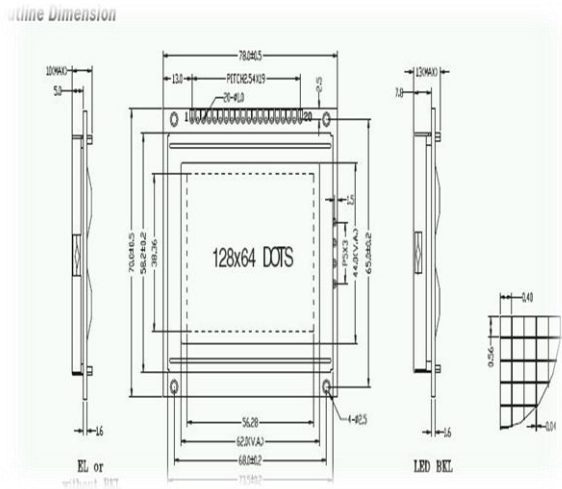


Figura 2: Dimensiones Pantalla GLCD [5].

El PICKIT 3 es un programador fabricado por Microchip© para programar toda su línea de microcontroladores PIC's® desde los PIC10, PIC12, PIC14, PIC16, PIC18, PIC24, dsPIC30 y dsPIC33 (todos los micro-controladores con memoria Flash sin excepción). El programador fue diseñado para programar los microcontroladores en circuito (ICSP) lo que significa que puede programar los microcontroladores montados directamente en tu aplicación y/o protoboard sin necesidad de tener que sacarlo y meterlo cada vez que se modifica el programa.



Figura 3: Hardware y software PICKIT 3.

3. Descripción e implementación del proyecto

Este proyecto nos permite visualizar por medio de una pantalla hasta 4 señales digitales que coloquemos a la entrada. Las señales pueden variar en frecuencia y para poder mostrarlo en la GLCD usamos dos potenciómetros para cambiar el tiempo de muestreo por periodo.

Las 4 señales son ingresadas por los puertos RB4 - RB7, y el control de tiempo de muestreo lo realizamos internamente en el pic con una conversión A/D de los potenciómetros en los puertos RA1 y RA2, que nos permitirá cambiar valores en TIMER1, logrando así interrupciones a distintos tiempos para poder visualizar diferentes frecuencias de las señales de entrada.

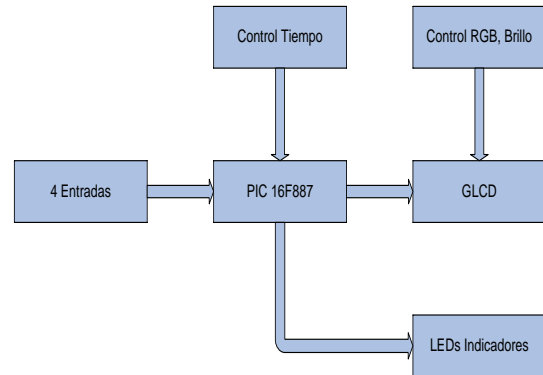


Figura 4 : Diagrama de bloques del Proyecto

3.1 Descripción uso convertidor A/D

Usamos un potenciómetro en cada puerto RA1 y RA2 del pic, ya que están configurados como entradas analógicas para poder usar la conversión A/D, al realizar la conversión se almacenan 10 bits, es decir un valor decimal entre 0 y 1023, dando así intervalos para lograr tener un control tipo switch entre los dos potenciómetros.

Se lo realizo de esta manera debido a que en total deberíamos representar 9 estados diferentes, lo que nos representaría desperdicio de recursos del pic si utilizáramos más puertos E/S, por lo que su funcionamiento esta dado por la siguiente tabla.

Tabla 1: Factor de conversión A/D Potenciómetro 1

Frecuencia (Hz)	Periodo (ms)	Valor Conversión A/D	Interval o Voltaje
5	0,20	0 - 341	1,66v
10	0,10	342 - 682	3,33v
20	0,05	683 - 1023	5,00v

Tabla 2: Factor conversión A/D Potenciómetro 2

Factor	Valor Conversión A/D	Intervalo Voltaje
1	0 – 341	1,66v
10	342 – 682	3,33v
100	683 – 1023	5,00v

Por ejemplo si queremos mostrar una señal de 1KHz o 1ms, el potenciómetro 1 debería tener un valor a su salida entre 1.66 y 3.33 voltios y el potenciómetro 2 debería tener un valor entre 3.33 y 5.00 voltios dando así $10 \times 100 = 1\text{KHz}$ o $0.1/100 = 1\text{ms}$.

3.2 Descripción uso timer1

Como es de conocimiento, para uso del temporizador en este caso TIMER1 a mas de configurar los bits para habilitar la interrupción se debe configurar los bits de valor inicial como son TMR1L y TMR1H ya que pueden alcanzar juntos un valor máximo en hexadecimal de $0xFFFF = 65535$ instrucciones, además a esto, el uso en este proyecto un cristal externo de 4MHz, es decir 1us cada instrucción, dando como total 65,535ms como tiempo máximo para generar la interrupciones sin usar el pre-escalador.

Configurando el registro T1CON, podremos cambiar el valor del pre-escalador, pudiendo así aumentar el tiempo máximo de estas interrupciones hasta en 8 veces, es decir a 524,28ms.

Tomando estas consideraciones podemos empezar a determinar diferentes valores de TMR1L, TMR1H y T1CON para generar interrupciones de acuerdo a la selección que el usuario de al equipo.

En este proyecto para tratar de generar lo más exacto posible las señales a la salida, lo que hacemos es tomar 4 muestras de un periodo completo de cada señal colocada a la entrada, por lo que si queremos tomar datos de una señal de 20Hz, es decir con periodo de 0,05s, generaremos interrupciones cada 12,5ms. Este principio básico es conocido como Teorema de Nyquist [8].

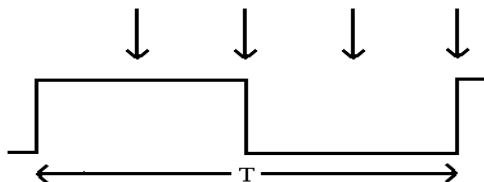


Figura 5: Interrupciones generadas por periodo

Este proceso fue posible realizarlo para mostrar señales desde 5Hz hasta los 20Hz, por limitaciones del equipo que demostraremos en el siguiente punto. Dada esta limitación para poder mostrar señales de más de 20Hz, ahora lo que haremos será tomar muestras saltando uno o varios periodos dependiendo de la señal que deseemos observar en la GLCD como mostraremos en la siguiente grafica.

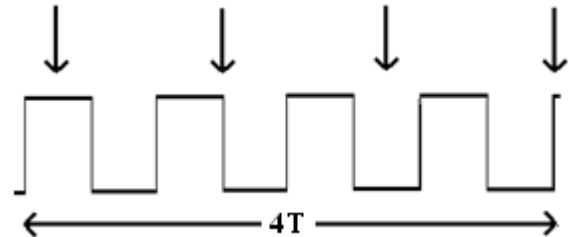


Figura 6: Interrupciones generadas saltando un periodo

Para el ejemplo de la figura anterior realizamos los cálculos debidos para obtener el tiempo al que debe generarse la interrupción y generaremos un periodo completo graficado en la GLCD a partir de 4 periodos en la señal de la entrada [7].

De tal manera que variando los valores a los que deseemos la interrupción lograríamos mostrar señales mayores a los 20Hz hasta una frecuencia máxima de 2KHz, teniendo así un rango de 5Hz hasta 2KHz en los 4 canales de entrada [6].

3.3 Gráficos en la GLCD

Para graficar en la GLCD usamos funciones de la propia librería de MikroC, principalmente lo que se hace es un barrido horizontal hacia la derecha, en cada interrupción se graficará dos pixeles en la GLCD por cada una de las señales de la entrada, censando el valor de esta señal, ya que los gráficos son en tiempo real, es decir, no se guarda información alguna sobre las señales de la entrada, se las grafica directamente, por lo que con lo explicado anteriormente tendremos que un periodo completo de la señal será equivalente a 8 pixeles y como la GLCD tiene 128 pixeles a lo ancho, se graficarán 16 periodos completos.

Graficamos con la función `Glcd_Dot(x,y,color)`, cada pixel dando como valores la coordenada en X, Y y además el color.

4. CONCLUSIONES:

1.- Este proyecto ha sido de mucha importancia para ampliar nuestros conocimientos sobre Microcontroladores, principalmente el uso de interrupciones ya sean éstas externas o como

temporizadores, además el uso de la conversión analógico – digital A/D y el uso de la librería propia de MikroC para graficar en una GLCD.

2.- El analizador lógico nos permite observar varias señales al mismo tiempo a diferencia de un osciloscopio que nos permite observar una o máximo dos señales dependiendo del fabricante, estos analizadores lógicos podemos implementarlos con más señales a la entrada, pero para este proyecto tendría el inconveniente de que aumentaría un poco más las limitaciones por lo que lo ideal fue hacerlo para cuatro señales digitales.

3.- Se desarrollo un analizador lógico usando el PIC16F887 a diferencia de otros microcontroladores más avanzados que por poseer mayor cantidad de memoria se podría grabar en una memoria EPROM para mejor manejo. Con este pic logramos sacar de una forma directa las señales a la GLCD sin ser guardadas en memoria.

4.- La limitación para mostrar señales mayores a 20Hz, en la que, solo se pueden observar hasta 10 señales de menor frecuencia en intervalos iguales, puede ser resuelta saltando más de 100 periodos en cada interrupción, pero esta solución trae consigo el inconveniente que las interrupciones tardarán mucho tiempo más en ejecutarse, tomando en cuenta el tiempo máximo de interrupción es de 524,28ms por 64 veces que grafica a lo ancho de la GLCD tomaría 33,55s, en el caso más largo, lo que es un tiempo demasiado extenso por lo que se prefirió dejarlo con la limitación.

5.-En la realización de este proyecto fue primordial el uso de las interrupciones, que fue el caso al que tuvimos que profundizar para el apropiado uso con las diferentes señales digitales que se puedan presentar a la entrada del dispositivo, específicamente el uso del Timer1. Es de vital importancia la sincronización de las interrupciones con las señales ya que un pequeño margen de error hará que la imagen mostrada en la GLCD no corresponda a la señal de la entrada.

5. Referencias

- [1]. Electronics Project Design, Introducción a Analizadores Lógicos.
<http://www.electronics-project-design.com/LogicAnalyzer.html>, 16 se septiembre del 2010
- [2]. Microchip, Datasheet PIC 16F887.
<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en026561>, 16 se septiembre del 2010
- [3]. MikroElectronica, mikroC PRO for PIC.
<http://www.mikroe.com/eng/products/view/7/mikroc-pro-for-pic/>, 16 se septiembre del 2010
- [4]. MikroElectronica, Manual mikroC PRO for PIC.
http://www.mikroe.com/eng/downloads/get/30/mikroc_pic_pro_manual_v100.pdf , 20 septiembre del 2010.
- [5]. MikroElectronica, Datasheet GLCD 128x64.
http://www.mikroe.com/eng/downloads/get/1010/glcd_128x64_specs.pdf, 20 septiembre del 2010.
- [6]. MikroElectronica, PIC Microcontrollers - Programming in C.
<http://www.mikroe.com/eng/products/view/285/book-pic-microcontrollers-programming-in-c/>, 27 septiembre del 2010.
- [7]. BitScope Designs, Analizadores Lógicos.
<http://www.bitscope.com/software/dso/>, 27 septiembre del 2010.
- [8]. UBM Electronics, Diseño Analizador Lógico.
<http://www.eetimes.com/design/programmable-logic/4015108/A-How-To-tutorial-on-logic-analyzer-basics-for-digital-design>, 27 septiembre del 2010.
- [9]. Jelsoft Enterprises Ltd., Osciloscopios vs Analizadores Lógicos

<http://www.motherboardpoint.com/logic-analyzer-vs-oscilloscope-t96637.html>, 18 de octubre del 2010

[10]. OpAmp Electronics, Tutorial Conversión Analógica a Digital. http://www.opamp-electronics.com/tutorials/digital_theory_ch_013.htm, 18 de octubre del 2010