



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

“Interfaz Gráfica con pantalla LCD para analizar señales de sensores externos desarrollada con microcontroladores avanzados y capacidad de comunicación serial.  
Fuente de energía: 4 pilas recargables AA”

**INFORME DE TESINA DE SEMINARIO**

Previa la obtención del Título de:

**INGENIERO EN ELECTRICIDAD**

**ESPECIALIDAD EN ELECTRÓNICA Y TELECOMUNICACIONES**

Presentado por:

Giovanny Stalin Tapia Velásquez

Tyrone Erick Cazar Aguilar

GUAYAQUIL – ECUADOR

AÑO 2010

## **AGRADECIMIENTO**

A Dios.

A los amigos.

A todas las personas que contribuyeron en el desarrollo de este trabajo.

A todos quienes apuestan por el desarrollo tecnológico en Ecuador.

Al Ing. Carlos Valdivieso, por su

Incondicional ayuda y consejos para

lograr terminar con éxito nuestro proyecto.

## DEDICATORIA

A Dios por ser nuestro padre y por  
guiarnos en la vida, a mi mami  
Noemi Aguilar, mi papi Carlos  
Cazar, mis hermanos, tíos, tías,  
primas, primos, mi sobrino Alessandro  
que es una nueva parte de mi vida y en  
especial a mi primo Elvis Armando Cazar  
Pico (+), que con su ayuda y demostración  
de afecto han  
contribuido a formar una excelente  
persona y cada día llenan de felicidad  
mi corazón.

*Tyrone Erick Cazar Aguilar*

A Dios que siempre me ha acompañado en todo momento, siendo su amor la fuente de energía para alcanzar mis metas.

A mi mami Cristina Lucia Velásquez Mariño, por su comprensión y ayuda incondicional, que siempre me ha inculcado perseverancia con valores éticos, permitiéndome iniciar mi vida profesional, a mis hermanos que siempre han apoyado con un granito de arena mis decisiones y retos, a mi princesita Briannita que con su sonrisa me llena de mucha emoción y alegría cada día, a mi tía Cecilia y en especial a mi primita bella que ha sido de gran ayuda en esta etapa de mi vida.

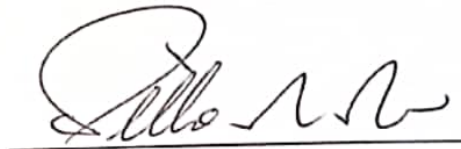
*Giovanny Stalin Tapia Velásquez*

# TRIBUNAL DE SUSTENTACION

A handwritten signature in black ink, appearing to read 'Carlos Valdivieso A.', written over a horizontal line.

Ing. Carlos Valdivieso A.

Profesor del Seminario de Graduación

A handwritten signature in black ink, appearing to read 'Hugo Villavicencio V.', written over a horizontal line.

Ing. Hugo Villavicencio V.

Delegado del Decano

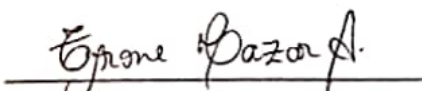
## DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este trabajo, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de exámenes y títulos profesionales de la ESPOL)



Giovanni Stalin Tapia Velásquez



Tyrone Erick Cazar Aguilar

## RESUMEN

El objetivo principal es diseñar una interfaz gráfica de buena presentación y fácil de usar, además que nos permita mostrar los datos obtenidos por los sensores externos a conectarse con nuestro proyecto, utilizando una pantalla GLCD táctil y herramientas de software como Mikrobasic Pro y Proteus.

El proyecto que a continuación se presenta consiste en el diseño y construcción de una interfaz gráfica con pantalla GLCD táctil, y está basado en el principio de mostrar por pantalla datos adquiridos por los diversos sensores, para luego su posterior análisis de cómo van variando las muestras con el tiempo.

La interfaz tiene dos etapas de secuencia, una para mostrar los diferentes menús que permitan una buena comprensión al usuario, y así poder escoger los datos del sensor externo deseado con solo tocar la pantalla táctil. La segunda consiste en la recepción de los datos que son enviados por los diversos proyectos mediante comunicación serial, para su posterior graficación y análisis.

Se utiliza la pantalla GLCD táctil 128\*64 RGB para mostrar los datos, el cual es controlado por el PIC-18F452. Además se utiliza un convertidor A/D para controlar la parte del funcionamiento del panel táctil.

# ÍNDICE GENERAL

AGRADECIMIENTO .....	II
DEDICATORIA .....	III
RESUMEN .....	VII
INTRODUCCIÓN .....	XII
CAPÍTULO 1 .....	1
1. DESCRIPCIÓN GENERAL DEL PROYECTO .....	1
1.1. Antecedentes .....	1
1.2. Descripción del Proyecto .....	2
1.3. Aplicaciones .....	3
1.3.1. Sistemas de Control y Automatización.....	3
1.4. Descripción del problema .....	3
1.5. Efectos.....	4
CAPÍTULO 2 .....	5
2. FUNDAMENTACIÓN TEÓRICA .....	5
2.1. Pantalla táctil .....	5
2.1.1. Principio de funcionamiento del panel táctil.....	7
2.2. Comunicación serial. ....	9
2.2.1. RS-232 (Estándar ANSI/EIA-232).....	10
2.2.2. Funcionamiento del conector DB-9.....	11
2.3. Herramientas de Software.....	1XXV
2.3.1. Mikrobasic Pro 3.2 .....	12



2.3.2. Proteus 7.5.....	13
2.4. PIC18F452 .....	14
CAPÍTULO 3 .....	16
3. DESCRIPCIÓN E IMPLEMENTACIÓN DEL PROYECTO .....	16
3.1. Prueba Inicial.....	16
3.1.1. Construcción.....	17
3.1.2. Determinación de las coordenadas X e Y del panel táctil.....	18
3.1.3. Conexión a un microcontrolador .....	19
3.2. Descripción del Proyecto Final .....	20
3.2.1. Diagrama de bloques del proyecto .....	21
3.3. Programación en Mikrobasic Pro 3.2 .....	22
CAPÍTULO 4 .....	27
4. SIMULACIÓN Y PRUEBAS .....	27
4.1. Simulación en proteus.....	27
4.2. Implementación en protoboard.....	35
4.3. Implementación de la Tarjeta Electrónica PCB. ....	35
CONCLUSIONES .....	
CONCLUSIONES .....	
RECOMENDACIONES .....	
BIBLIOGRAFÍA .....	
BIBLIOGRAFÍA .....	

# ÍNDICE DE FIGURAS

Fig. 1.1 Contacto entre el usuario y el menu en pantalla .....	2
Fig 1.2 Aplicación en el control industrial .....	3
Fig. 2.1 Pantalla táctil .....	6
Fig. 2.1.1 Contacto con el panel táctil .....	6
Fig. 2.1.2 Estructura interna del panel táctil .....	7
Fig. 2.1.3 Dimensiones de una pantalla GLCD táctil .....	7
Fig. 2.2 Pines del conector DB-9.....	11
Fig. 2.3 Configuración de pines del PIC18F452 .....	15
Fig. 3.1 Imagen inicial en la construcción .....	17
Fig. 3.1.1 Circuito convertidor A/D.....	18
Fig. 3.1.2 Conexión a pantalla táctil .....	19
Fig. 3.2 Diagrama de bloques del proyecto.....	22
Fig. 4.1 Potenciómetros para poder simular la parte del panel táctil.....	27
Fig. 4.1.1 Secuencia de inicialización del proyecto; <b>Error! Marcador no definido.</b> .....	28
Fig. 4.1.2 Simulación de la imagen representativa de la institución .....	29

Fig. 4.1.3 Simulación que muestra el nombre del Seminario de Graduación y del Profesor.....	30
Fig. 4.1.4 Simulación para mostrar en pantalla el diagrama de bloques del proyecto	31
Fig. 4.1.5 Simulación que muestra sensores que el usuario desee escoger.....	31
Fig. 4.1.6 Simulación del menú de las dos trayectorias del acelerómetro .....	32
Fig. 4.1.7 Simulación de las mediciones X vs. Y, ángulo vs. tiempo.....	33
Fig. 4.1.8 Simulación del eje de coordenadas X vs. Y .....	33
Fig. 4.1.9 Simulación de dos ángulos censados por la brújula magnética. ....	34
Fig. 4.2 Interfaz gráfica con pantalla GLCD táctil en protoboard.....	35
Fig. 4.3.1 Diseño de la tarjeta electrónica. ....	36
Fig. 4.3.2 Vista 3-D .....	36

# ÍNDICE DE TABLAS

Tabla 2.1.1 Configuración de pines de la pantalla GLCD táctil .....	8
Tabla 2.1.2 Parámetros técnicos de operación de la pantalla GLCD táctil.....	8
Tabla 2.3 Parámetros técnicos del PIC18F452.....	15

# INTRODUCCIÓN

El presente proyecto tiene como finalidad el diseño y construcción de una interfaz gráfica con pantalla GLCD táctil. La implementación se la realizará con un circuito que controle las diversas etapas de la interfaz, que incluye un convertidor A/D para el panel táctil. La pantalla GLCD táctil será controlada por un PIC 18F452 y se comunicará con sensores externos a través de comunicación serial.

En el primer capítulo, se da un detalle general del proyecto con su debida descripción, antecedentes, planteo del problema, aplicación en el campo industrial, causales que motivaron a la elaboración de dicho proyecto y efectos al realizarlo.

En el segundo capítulo, se describen las herramientas de hardware utilizadas, que incluyen equipos y materiales adicionales en la construcción de la interfaz gráfica. Además se detalla el software utilizado Mikrobasic Pro, que es la principal herramienta de programación del PIC 18F452, por lo que se da una breve descripción de las funciones utilizadas para desarrollar este proyecto. Además hemos utilizado Proteus versión 7.5, un potente software que nos permite la simulación del proyecto y así ir modificando sentencias hasta llegar a nuestro objetivo.

En el tercer capítulo, se describe el prototipo inicial, del cual surgió la idea del proyecto final, él cual es una versión mejorada de una tarea realizada durante las clases del seminario el cual consiste de un sensor de temperatura con muestreo en una pantalla GLCD.

En el cuarto y último capítulo encontramos todos los diagramas electrónicos para el diseño del circuito controlador de la pantalla táctil. Todos los valores de las pruebas efectuadas para la configuración correspondiente del panel táctil

# CAPÍTULO 1

## 1. DESCRIPCIÓN GENERAL DEL PROYECTO

### 1.1. Antecedentes

Antes de comenzar a recabar información acerca del tema, es importante definir el significado de “interfaz gráfica”, es un programa informático que actúa de interpretador, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

En el ritmo rápido del mundo de hoy, las largas colas de espera es uno de los mayores problemas que todavía hay que mejorar. Nuestro proyecto sería de gran aporte al aceleramiento de los tiempos de espera ya que brinda al usuario la facilidad de colocar sus pedidos de una manera ágil y veloz. Entre las aplicaciones de la interfaz gráfica están el autoservicio de terminales, tiendas de ocupado, servicio de comida rápida, los cajeros automáticos. Los clientes pueden rápidamente colocar sus órdenes, acelerando el proceso y así brindar un servicio de mejor calidad.



**Fig. 1.1 Contacto entre el usuario y el menu en pantalla**

## **1.2. Descripción del proyecto**

En el presente capítulo se explica el problema que se pretende solucionar a través del proyecto. Para este fin, es fundamental conocer de los diversos tipos de interfaz gráfica con pantallas GLCD táctiles que existen como medio de representación para el análisis de sensores externos.

Además tratamos sobre la nueva tecnología en el mercado de las pantallas táctiles para el manejo de los menús de usuarios. Así mismo se detallan las justificaciones por las cuales se decidió emprender el proyecto y el posible impacto que éste vaya a tener en el medio y demás efectos.

## **1.3. Aplicación**

La aplicación del proyecto puede ser en diversos campos como:

### **1.3.1. Sistemas de Control y Automatización**





**Fig. 1.2 Aplicación en el control industrial**

La interfaz de pantalla táctil es útil en sistemas que van desde el control de procesos industriales hasta la automatización del hogar. Al integrar el dispositivo de entrada con la pantalla, valioso espacio de trabajo puede ser salvado, además que con una interfaz gráfica los operadores pueden monitorear y controlar operaciones complejas en tiempo real, simplemente tocando la pantalla.

#### **1.4. Descripción del problema**

El problema planteado es buscar la manera de representar gráficamente en una pantalla el análisis de diferentes tipos de sensores externos.

Actualmente existen en el mercado diferentes tipos de pantallas táctiles GLCD que podrían compararse con nuestro proyecto, pero la diferencia es que nosotros tenemos la capacidad de comunicación serial y una fuente de energía de 4 pilas recargables AA, teniendo como diferencia la adquisición de energía para el funcionamiento del dispositivo.

#### **1.5. Efectos**

El tiempo en la actualidad representa grandes cantidades de dinero. Los sistemas de pantalla táctil son fáciles de usar que los empleados pueden realizar su trabajo más rápido, dando como resultado una mejor calidad de servicio al cliente y con esto logramos brindar nuestro producto a mayor cantidad de personas evitando las largas colas de espera. Con la ayuda de la tecnología y de nuestro proyecto optimizamos el tiempo, haciendo más eficaz las diversas aplicaciones de atención al cliente.

## **CAPÍTULO 2**

### **2. FUNDAMENTACIÓN TEÓRICA**

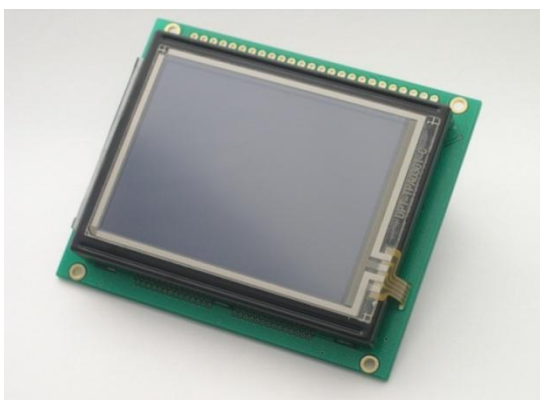
En el presente capítulo se muestran los fundamentos básicos de una interfaz gráfica a través de una pantalla táctil GLCD. Además se explican las técnicas,

capacidades de comunicación con los sensores externos, también se expone los campos de aplicación del proyecto, por último se describen los programas desarrollados para la comunicación serial y hardware implementados que se utilizan en la interfaz gráfica con pantalla táctil GLCD en el mercado.

## 2.1. Pantalla táctil

Una pantalla táctil es un fino panel autoadhesivo colocado sobre la pantalla de un LCD gráfico. Es muy sensible a la presión de manera que un suave toque provoca algunos cambios en la señal de salida. Hay diferentes tipos de paneles táctiles. El más sencillo de ellos es el panel táctil resistivo.

**Glcd+Panel táctil = Pantalla táctil**



**Fig. 2.1 Pantalla táctil**

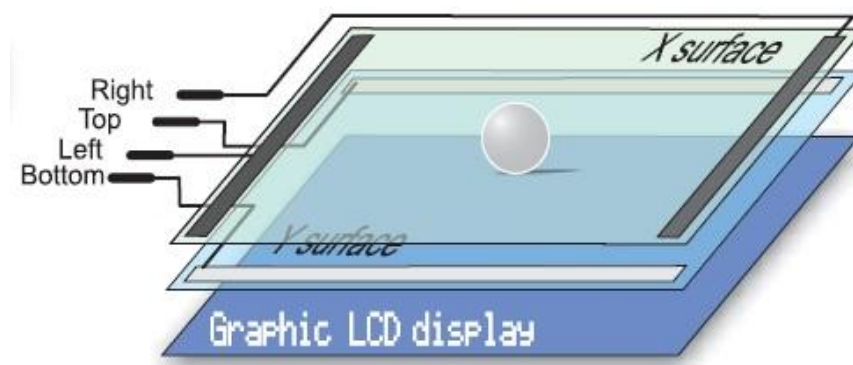
Las pantallas táctiles resistivas son por norma general más accesibles pero tienen una pérdida de aproximadamente el 25% del brillo debido a las múltiples capas necesarias. Otro inconveniente que tienen es que pueden ser dañadas por objetos afilados. Por el contrario no se ven afectadas por elementos externos como polvo

o agua, razón por la que son el tipo de pantallas táctiles más usado en la actualidad.



**Fig. 2.1.1 Contacto con el panel táctil**

### 2.1.1. Principio de funcionamiento del panel táctil



**Fig. 2.1.2 Estructura interna del panel táctil**

Un panel táctil resistivo está compuesto por dos láminas rígidas transparentes, formando una estructura "sándwich", que tienen una capa resistiva en sus caras internas. La resistencia de estas capas no excede normalmente de 1[Kohm]. Los lados opuestos de las láminas disponen de contactos para acceder a un cable plano.

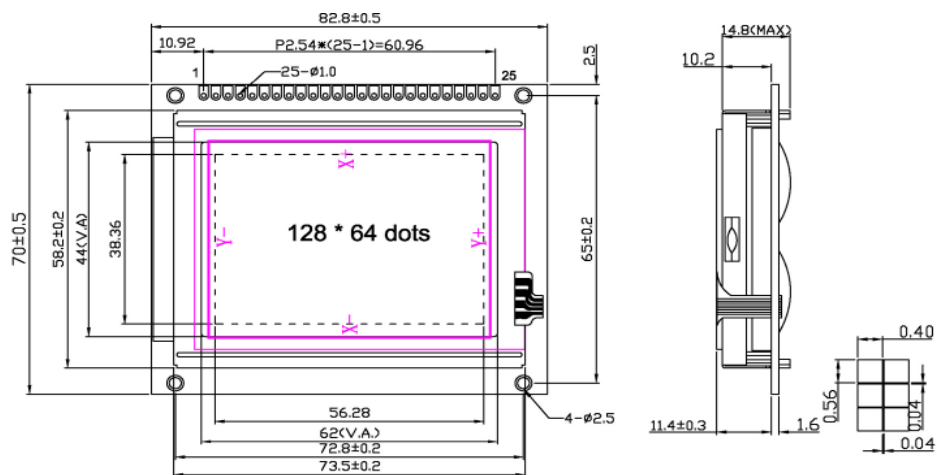


Fig.

### 2.1.3 Dimensiones de una pantalla GLCD táctil

PIN	SIGNAL	PIN	SIGNAL	PIN	SIGNAL
1	/CS1	10	DB1	19	LED_R+
2	/CS2	11	DB2	20	LED_G+
3	VSS	12	DB3	21	LED_B+
4	VDD	13	DB4	22	Y-
5	V0	14	DB5	23	X-
6	D/I	15	DB6	24	Y+
7	R/W	16	DB7	25	X+
8	E	17	/RST		
9	DB0	18	VEE		

Tabla 2.1.2 Configuración de pines de la pantalla GLCD táctil

TOLERANCES UNLESS OTHERWISE STATED

XX.X  $\pm$ 0.20    X.X  $\pm$ 0.10

**PERFORMANCE FEATURES**

<b>LCD TYPE</b>	<b>STN,Transflective</b>
<b>Viewing Angle</b>	<b>6:00</b>
<b>Backlight</b>	<b>LED Sidelight(R, G, B)</b>
<b>Operation temperature</b>	<b>-20° C ~ +70° C</b>
<b>Storage temperature</b>	<b>-25° C ~ +80° C</b>
<b>Controller</b>	<b>S6B0107 S6B0108</b>
<b>Operation Voltage of LCM</b>	<b>+5.0 V</b>

**Tabla 2.1.2 Parámetros técnicos de operación de la pantalla GLCD táctil**

## 2.2. Comunicación serial

La comunicación serial es un protocolo muy común (no hay que confundirlo con el Bus Serial de Comunicación, o USB) para comunicación entre dispositivos que se incluye de manera estándar en prácticamente cualquier computadora. La mayoría de las computadoras incluyen dos puertos seriales RS-232. La comunicación serial es también un protocolo común utilizado por varios dispositivos para instrumentación; existen varios dispositivos compatibles con GPIB que incluyen un puerto RS-232. Además, la comunicación serial puede ser utilizada para adquisición de datos si se usa en conjunto con un dispositivo remoto de muestreo.

El concepto de comunicación serial es sencillo. El puerto serial envía y recibe bytes de información un bit a la vez. Aun y cuando esto es más lento que la comunicación en paralelo, que permite la transmisión de un byte completo por

vez, este método de comunicación es más sencillo y puede alcanzar mayores distancias. Por ejemplo, la especificación IEEE 488 para la comunicación en paralelo determina que el largo del cable para el equipo no puede ser mayor a 20 metros, con no más de 2 metros entre cualesquier dos dispositivos; por el otro lado, utilizando comunicación serial el largo del cable puede llegar a los 1200 metros.

Típicamente, la comunicación serial se utiliza para transmitir datos en formato ASCII. Para realizar la comunicación se utilizan 3 líneas de transmisión: (1) Tierra (o referencia), (2) Transmitir, (3) Recibir. Debido a que la transmisión es asincrónica, es posible enviar datos por una línea mientras se reciben datos por otra. Existen otras líneas disponibles para realizar handshaking, o intercambio de pulsos de sincronización, pero no son requeridas. Las características más importantes de la comunicación serial son la velocidad de transmisión, los bits de datos, los bits de parada, y la paridad. Para que dos puertos se puedan comunicar, es necesario que las características sean iguales.

### **2.2.1. RS-232 (Estándar ANSI/EIA-232):**

Es el conector serial hallado en las PCs IBM y compatibles. Es utilizado para una gran variedad de propósitos, como conectar un ratón, impresora o modem, así como instrumentación industrial. Gracias a las mejoras que se han ido desarrollando en las líneas de transmisión y en los cables, existen aplicaciones en las que se aumenta el desempeño de RS-232 en lo que respecta a la distancia y velocidad del estándar. RS-232 está limitado a comunicaciones de punto a

punto entre los dispositivos y el puerto serial de la computadora. El hardware de RS-232 se puede utilizar para comunicaciones seriales en distancias de hasta 50 pies.

### 2.2.2. Funcionamiento del conector DB-9

Conector externo de la computadora y expuesto del cable.

Funciones de los pines en RS-232:

Datos: TXD – pin 3

RXD - pin 2

Handshake: RTS – pin 7

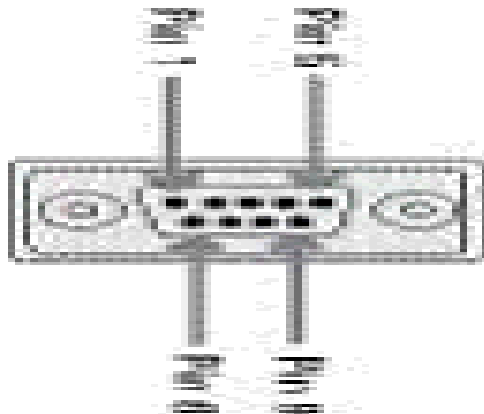
CTS – pin 8

DSR – pin 6

DCD – pin 1

DTR-pin4

Tierra: GND – pin 5



**Fig. 2.2 Pines del conector DB-9**



## **2.3. Herramientas de Software**

El proyecto utilizó dos tipos de software, el primero tiene como fin la programación del PIC 18F452 y el segundo tiene como objetivo la simulación completa del proyecto.

### **2.3.1. Mikrobasic Pro 3.2**

Es una herramienta de desarrollo que permite realizar proyectos para microcontroladores. Proporciona una solución fácil para aplicaciones de sistemas embebidos, sin comprometer el rendimiento o el control, además desarrolla rápidamente y despliega aplicaciones complejas.

Originalmente concebido como una herramienta fácil de usar, BASIC se generalizó en microcomputadoras en la década de 1980, y sigue siendo popular hoy en día. BASIC permite un desarrollo mucho más rápido y fácil de las solicitudes de consentimiento fundamentado previo en comparación con el lenguaje del microchip Ensamblador MPASM. Al escribir el código de MCU, con frecuencia los programadores hacen frente a las mismas funciones, tales como la comunicación serial, la impresión en la pantalla LCD, la generación de señales PWM, etc. Con el fin de facilitar la programación, se basa en una serie de rutinas built-in, destinados para resolver estos problemas.

En cuanto a la ejecución y el tamaño del programa en cuestión, MPASM tiene una pequeña ventaja en relación con BASIC. Por esta razón hay una opción de

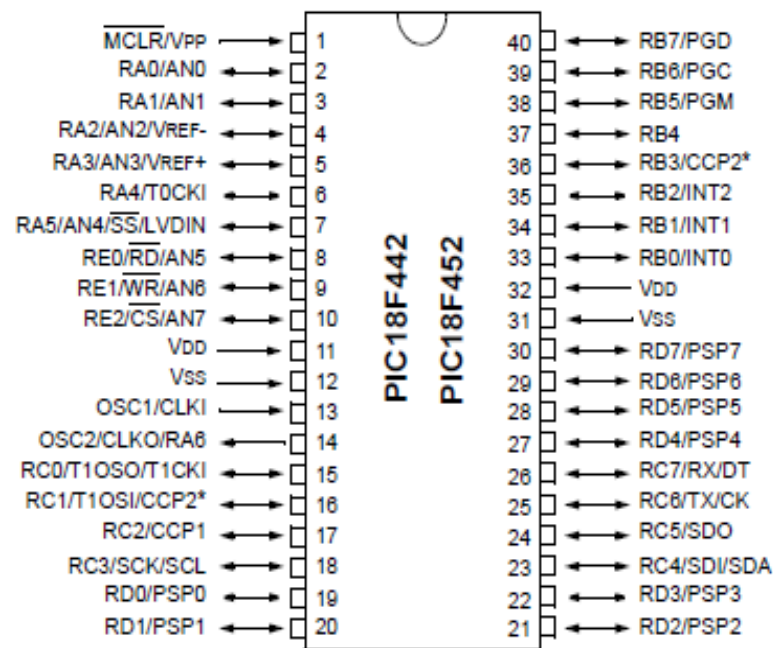
combinar BASIC y el código ensamblador. El ensamblador se utiliza comúnmente para las partes del programa en el que el tiempo de ejecución es crítico o que los comandos se ejecuten gran número de veces. Los microcontroladores modernos, como los PIC, ejecutan instrucciones en un solo ciclo de reloj. Si el reloj del microcontrolador es 4MHz, una instrucción en lenguaje ensamblador requiere  $250\text{ns} \times 4 = 1\mu\text{s}$ . Como cada comando básico es técnicamente una secuencia de instrucciones en ensamblador, el tiempo exacto necesario para su ejecución se puede calcular simplemente sumando los tiempos de ejecución de las instrucciones.

### **2.3.2. Proteus 7.5**

Es un paquete de software para el diseño de circuitos electrónicos que incluye captura de los esquemas, simulación analógica y digital combinada, además posee una herramienta ARES que se utiliza para el diseño de circuitos impresos. Proteus es un entorno integrado diseñado para la realización completa de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño, simulación, depuración y construcción. El paquete está compuesto por dos programas: ISIS, para la captura y simulación de circuitos; y ARES, para el diseño de PCB's. También permite simular y depurar el funcionamiento de todo el sistema ejecutando el software paso a paso, insertando puntos de ruptura (breakpoints, que también pueden ser generados por el hardware), mirando el contenido de registros y posiciones de memoria, etc. y comprobando si la respuesta del hardware es la correcta. También se simulan herramientas electrónicas, como osciloscopios, analizadores lógicos, voltímetros, etc.

## 2.4. PIC18F452

Es un microcontrolador CMOS FLASH de 8 bits de arquitectura RISC capaz de operar con frecuencias de reloj hasta los 40 MHz, fácil de programar. Posee internamente un oscilador de 4 MHz y un circuito de Power-On Reset que eliminan la necesidad de componentes externos y expanden a 16 el número de pines que pueden ser utilizados como líneas I/O (entrada/salida; Input/ Output) de propósito general. Proporciona una memoria de datos EEPROM de 256x8 (128 Bytes), una memoria de programa FLASH de 2048x14 (2K con 14 bits por localidad), una memoria de datos RAM de propósito general de 1,536, dos módulos CCP (captura/comparación/PWM) de 10-bit, un USART, 3 comparadores análogos, una referencia de voltaje programable y tres temporizadores.



**Fig. 2.3 Configuración de pines del PIC18F452**

Parameter Name	Value
Program Memory Type	Flash
Program Memory (KB)	32
CPU Speed (MIPS)	10
RAM Bytes	1,536
Data EEPROM (bytes)	256
Digital Communication Peripherals	1-A/E/USART, 1-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	2 CCP
Timers	1 x 8-bit, 3 x 16-bit
ADC	8 ch, 10-bit
Temperature Range (C)	-40 to 125
Operating Voltage Range (V)	2 to 5.5
Pin Count	40

**Tabla 2.3 Parámetros técnicos del PIC18F452**

# CAPÍTULO 3

## 3. DESCRIPCIÓN E IMPLEMENTACIÓN DEL PROYECTO

### 3.1. Prueba inicial

Para la realización del proyecto tomamos como consideración dividirlo por fases para así ir diseñando y simulando paso a paso las diferentes partes del mismo. La primera etapa consistía en realizar un programa en el cual nos permitiera mostrar por pantalla una gráfica, en este caso colocamos como imagen el logo de la ESPOL. La segunda etapa consistía en encargarnos de manejar el flujo de datos de la comunicación serial, sincronizar los tiempos de transmisión y recepción de los datos. La última etapa consistía en la elaboración e implementación del circuito del touch panel.

#### 3.1.1. Construcción

La construcción del circuito de nuestro proyecto consistió en el cableado del protoboard entre el microcontrolador y la pantalla táctil GLCD con la utilización de 4 potenciómetros, 3 para el manejo de los colores de RGB (rojo, verde, azul), al ir regulando los potenciómetros podemos manipular la amplia gama de colores, a su vez el otro potenciómetro se encarga de la regulación del contraste

dando como resultado mejor calidad a la apreciación de las imágenes presentadas en la pantalla. En esta primera parte del proyecto solo se realizó la circuitería para verificar que la pantalla esté en buen estado.



**Fig. 3.1 Imagen inicial en la construcción**

La siguiente parte de la construcción fue el circuito de un convertidor A/D, para el funcionamiento del panel táctil, utilizando 3 transistores NPN y 2 PNP, 10 resistencias y dos capacitores

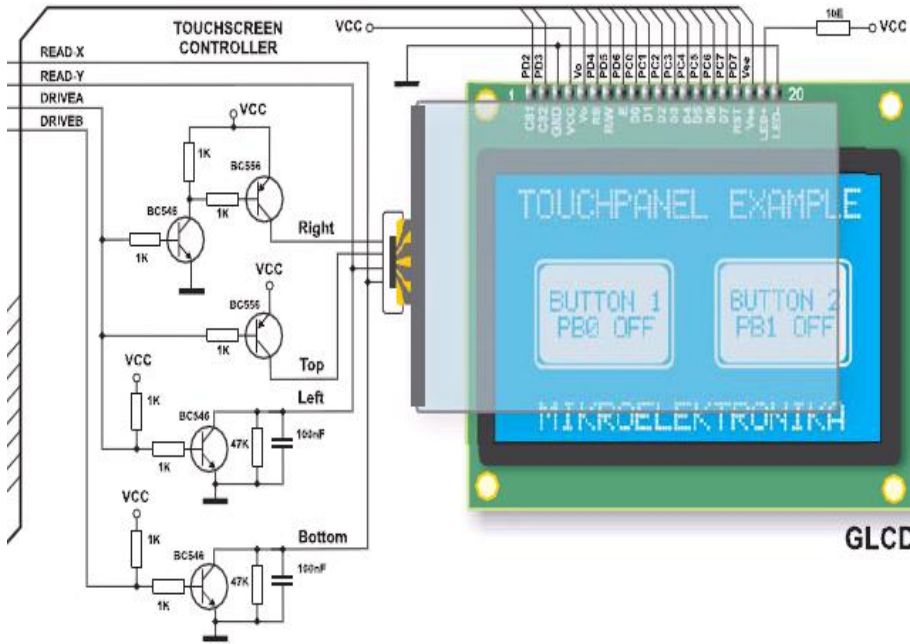


Fig. 3.1.1 Circuito convertidor A/D

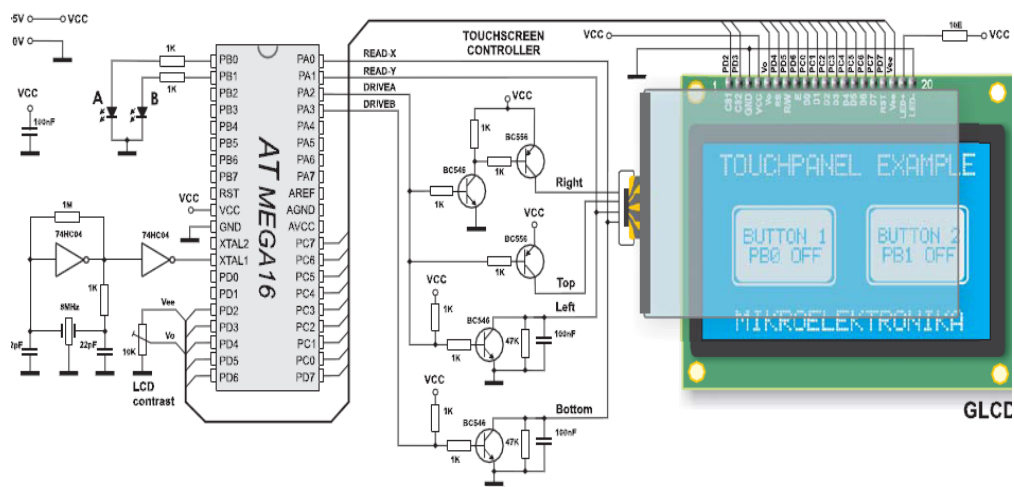
3.1.2. Determinación de las coordenadas X e Y del panel táctil

El procedimiento para determinar las coordenadas de la posición del panel que ha sido presionada, puede dividirse en dos pasos.

El primero es la determinación de la coordenada X y el segundo el de la coordenada Y de la región presionada. Para determinar la coordenada X, es preciso conectar el contacto izquierdo de la superficie X a tierra y el contacto derecho a la fuente de alimentación (+5V). Esto permite obtener un divisor de tensión al presionar el panel táctil. El valor de la tensión obtenida en el divisor se puede leer en el contacto inferior de la superficie Y. La tensión variará en el rango de 0V a la tensión suministrada por la fuente de alimentación y depende de la coordenada X. Si el punto está próximo al contacto izquierdo de la superficie X, la tensión estará próxima a 0 V. Para la determinación de la

coordenada Y, es preciso conectar el contacto inferior de la superficie Y a tierra, mientras que el contacto superior se conectará a la fuente de alimentación. En este caso, la lectura de la tensión se hará en el contacto izquierdo de la superficie X.

### 3.1.3. Conexión a un microcontrolador



**Fig. 3.1.2 Conexión a pantalla táctil**

Para conectar un panel táctil al microcontrolador, es preciso crear un circuito que se encargue del control del panel táctil. Por medio de este circuito el microcontrolador conecta los contactos adecuados del panel táctil a tierra y a la tensión de alimentación para determinar las coordenada X e Y. El contacto inferior de la superficie Y y el contacto izquierdo de la superficie X están conectados al convertidor A/D del microcontrolador. Las coordenadas X e Y se determinan midiendo la tensión en los respectivos contactos. En el software consiste en mostrar un menú en una pantalla LCD gráfica y leer los valores del convertidor A/D que representan realmente las coordenadas X e Y de la



posición. Una vez determinadas las coordenadas, es posible decidir qué es lo que deseamos que haga el microcontrolador.

### **3.2. Descripción del proyecto final**

Luego de la construcción del circuito y la programación del microcontrolador estamos listos para que funcione el prototipo, inicialmente notamos ciertos errores en la visualización de la toma de muestras, por motivo de que la primera prueba de comunicación con un sensor externo la tuvimos que realizar utilizando otro microcontrolador que simule un sensor de distancia.

Después de varias pruebas realizamos el código final que consiste en la presentación de varias imágenes. El microcontrolador se encarga de hacer trabajar a los demás dispositivos como es la pantalla táctil, la cual va mostrando imágenes y menús. El menú se realizó de tal manera que mantenga una imagen hasta que el usuario presione la pantalla y así avance a la siguiente imagen hasta que llegue al menú principal. La primera imagen muestra el logo de la ESPOL, una vez presionada la pantalla muestra otra imagen en la cual indica el nombre del seminario y del profesor quien lo dicta

“MICROCONTROLADORES AVANZADOS”, presionado de nuevo mostramos el diagrama de bloques del proyecto, con esto queremos indicar que para ir avanzado de ventana hay que presionar la pantalla.

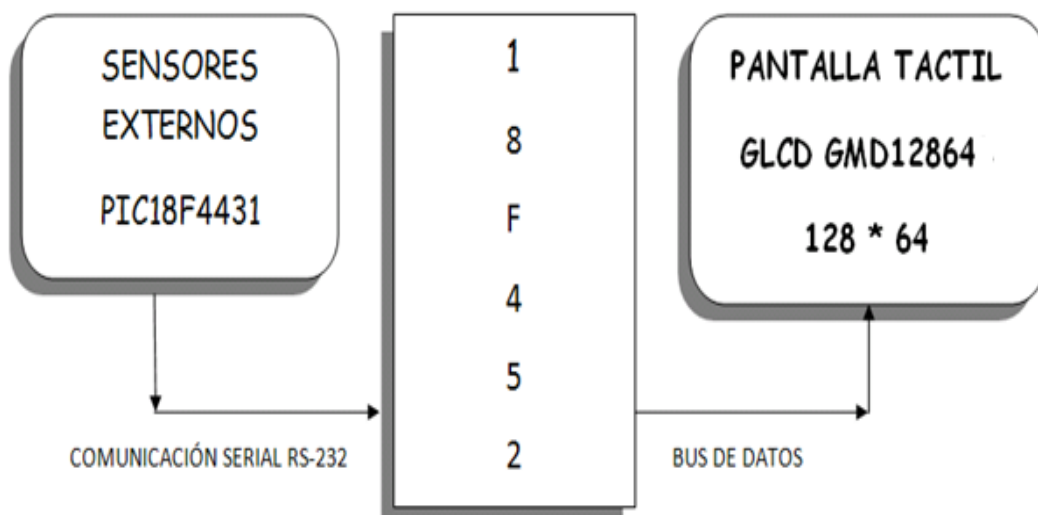
Por ultimo llegamos al menú de los sensores externos donde podremos seleccionar los datos de la muestras del sensor que queramos presentar, dependiendo de la elección escogida se mostrará diferentes ejes y tipos de graficas con sus respectivas unidades, como por ejemplo para el medidor de

distancia se mostrará un eje Kms vs Tiempo, donde se apreciará como varia la distancia con el pasar del tiempo, en el caso de la brújula se muestra las coordenadas norte, sur, este y oeste, mientras que para medidor de inclinación se muestran los ángulos.

### 3.2.1. Diagrama de bloques

Para la elaboración del proyecto utilizamos el PIC18F452 debido a que posee mayor cantidad de memoria, ya que dentro de la programación manejamos la graficación de 5 proyectos lo cual demanda una gran cantidad de líneas de código, además que utilizamos una diversa gama de gráficos y menús que consumen espacio en memoria.

Este proyecto no se encarga de almacenar los datos en memoria sino solo graficarlos para su posterior análisis, así que éstos son enviados por medio de la comunicación serial RS-232 a nuestro microcontrolador, para luego graficarlos en una pantalla GLCD de dimensiones 128\*64, además que hemos implementado un panel táctil que será de gran ayuda para el fácil manejo de los diversos menús del proyecto con solo presionar en la región deseada.



**Fig. 3.2 Diagrama de bloques del proyecto**

### 3.3 Programación en Mikrobasic Pro 3.2

A continuación se explica cada parte del programa realizado para el funcionamiento del microcontrolador.

#### Inicialización de las variables

```

dim GLCD_DataPort as byte at PORTD

dim GLCD_CS1 as sbit at LATB0_bit ' for writing to output pin always use latch (PIC18 family)
    GLCD_CS2 as sbit at LATB1_bit ' for writing to output pin always use latch (PIC18 family)
    GLCD_RS as sbit at LATB2_bit ' for writing to output pin always use latch (PIC18 family)
    GLCD_RW as sbit at LATB3_bit ' for writing to output pin always use latch (PIC18 family)
    GLCD_EN as sbit at LATB4_bit ' for writing to output pin always use latch (PIC18 family)
    GLCD_RST as sbit at LATB5_bit ' for writing to output pin always use latch (PIC18 family)

dim GLCD_CS1_Direction as sbit at TRISB0_bit
    GLCD_CS2_Direction as sbit at TRISB1_bit
    GLCD_RS_Direction as sbit at TRISB2_bit
    GLCD_RW_Direction as sbit at TRISB3_bit
    GLCD_EN_Direction as sbit at TRISB4_bit
    GLCD_RST_Direction as sbit at TRISB5_bit

dim DRIVE_A as bit ' sirve de salida digital para la pantalla tactil
dim DRIVE_B as bit ' sirve de salida digital para la pantalla tactil
dim i as byte ' contador

dim adc_rd1 as word ' variable donde se adquiere la coordenada X del panel
dim adc_rd2 as word ' variable donde se adquiere la coordenada Y del panel
dim muestras as byte ' variable usada para el conteo de las muestras a tomar

```

```

dim trayectoria as byte      ' variables usadas para ingresar a los diferentes menus
dim opcion_menu as byte
dim conversion as byte
dim tipo_grafico as byte

dim x as word                ' variables donde se almacenan las coordenadas en X y Y para
dim y as word                ' verificar donde esta tocando el usuario en la pantalla tactil
dim x_coord128 as word
dim y_coord64 as word

dim y1 as byte              ' variables usadas para graficar en los diferentes ejes
dim y2 as byte
dim x1 as byte
dim x2 as byte

dim coordenadaX as byte
dim coordenadaY as byte
dim grados as byte

dim bandera as byte         ' banderas que me mantienen dentro de un menu especifico
dim bandera1 as byte       ' sirven para la presentacion
dim bandera2 as byte
dim bandera3 as byte

```

## Subrutinas

```

'FUNCION QUE GENERA LA COORDENADA EN X DEL TOUCH PANEL
sub function GetX() as word
    portb.6 = 1          ' DRIVEA = 1 (LEFT drive on, RIGHT drive on, TOP drive off )
    portb.7 = 0          ' DRIVEB = 0 (BOTTOM drive off )
    Delay_ms(10)
    adc_rd1 = adc_read(0) ' get ADC value from 2nd channel
    x = adc_rd1

    tmp4 = (x *128) / 1023

    result = tmp4
end sub

'FUNCION QUE GENERA LA COORDENADA EN Y DEL TOUCH PANEL
sub function GetY() as word
    portb.6 = 0          ' DRIVEA = 0 (LEFT drive off , RIGHT drive off , TOP drive on)
    portb.7 = 1          ' DRIVEB = 1 (BOTTOM drive on)
    Delay_ms(10)
    adc_rd2 = adc_read(1)
    y = adc_rd2
    tmp5 = 63 - ((y *63) / 1023)

    result = tmp5
end sub

```

```

' FUNCION ENCARGADA DE ELEGIR LOS PROYECTOS
sub function ELEGIR_PROYECTO(dim x_coord as word, dim y_coord as word) as byte
|
|   if (((x_coord >=28) and (x_coord <= 98))and ((y_coord >=5) and (y_coord <=21))) then
|       result = 1      ' si el usuario desea elegir acelerometro
|   else
|       if (((x_coord >=28) and (x_coord <= 98))and ((y_coord >=24) and (y_coord <= 40))) then
|           result = 2      ' si el usuario desea elegir brujula
|       else
|           if (((x_coord >=28) and (x_coord <= 98))and ((y_coord >=43) and (y_coord <= 59))) then
|               result = 3      ' si el usuario desea elegir GPS
|           else
|               if (((x_coord >=5) and (x_coord <= 22))and ((y_coord >=44) and (y_coord <= 58))) then
|                   result = 4      ' si el usuario desea salir
|               else
|                   if (((x_coord >=103) and (x_coord <= 120))and ((y_coord >=44) and (y_coord <= 58))) then
|                       result = 5      ' si el usuario desea elegir otro programa
|                   else
|                       result =6      ' si el usuario desea escoge cualquier otra opcion fuera de los rangos ;
|                   end if
|               end if
|           end if
|       end if
|   end if
| end if
end sub

```

## Configuración de puertos de E/S

```

ADCON1 = ADCON1 or 0x05      ' Configure AN pins as digital
'CMCON = CMCON or 7         ' Turn off comparators
TRISA = 0xFF
TRISB = 0
TRISD = 0
Glcd_Init()                  ' Initialize Glcd
Glcd_Fill(0x00)              ' Clear Glcd
inicializar()
Glcd_Image(@tortuga_bmp)
|
while (banderaT = 0)
|   x_coord128 = GetX()      ' obtiene la coordenada en X
|   y_coord64 = GetY()      ' obtiene la coordenada en Y
|   if (((x_coord128 >=0) and (x_coord128 <= 128))and ((y_coord64 >=0) and (y_coord64 <=63))) then
|       LIMPIAR()
|       banderaT = 1
|   else
|       banderaT = 0
|   end if
| end while
|
Glcd_Image(@PRESENTACION_bmp)
while (banderaP = 0)

```

```

while (bandera1 = 0)
  Glcd_Image(@proyectos_bmp)
  Move_Espera()
  x_coord128 = GetX()      ' obtiene la coordenada en X
  y_coord64 = GetY()      ' obtiene la coordenada en Y
  opcion_menu = ELEGIR_PROYECTO(x_coord128,y_coord64) ' obtengo el proyecto
  ' OPCION_MENU = 1   ACELEROMETRO
  ' OPCION_MENU = 2   DISTANCIA
  ' OPCION_MENU = 3   BRUJULA
  ' OPCION_MENU = 4   SALIR
  ' OPCION_MENU = 5   OTRO MENU DE PROYECTOS
  ' OPCION_MENU = 6   MANTENGO EL LAZO
  if (opcion_menu = 1) then ' ACELEROMETRO
    while (bandera2 = 0)
      Glcd_Image(@menu_bmp)      ' trayectorias
      Move_Espera()
      x_coord128 = GetX()      ' obtiene la coordenada en X
      y_coord64 = GetY()      ' obtiene la coordenada en Y
      trayectoria = ELEGIR_TRAECTORIA(x_coord128,y_coord64)
      ' si trayectoria es 1 voy al menu de las primeras muestras
      ' si trayectoria es 2 voy al menu de las segundas muestras
      ' si trayectoria es 3 regreso al menu anterior pa elegir otro proyecto
      ' si trayectoria es 4 me mantengo en el lazo hasta que presione correctamente
      if (trayectoria = 1) then 'trayectoria 1
        while (bandera3 = 0)
          Glcd_Image(@graficos_bmp) |
          Move_Espera()
          x_coord128 = GetX()      ' obtiene la coordenada en X
          y_coord64 = GetY()      ' obtiene la coordenada en Y

```

```

if (trayectoria = 1) then      'trayectoria 1
  while (bandera7 = 0)
    Glcd_Image(@graficos_bmp)
    x_coord128 = GetX()      ' obtiene la coordenada en X
    y_coord64 = GetY()      ' obtiene la coordenada en Y
    tipo_grafico = ELEGIR_GRAFICO(x_coord128,y_coord64)
    if (tipo_grafico = 1) then  ' eligio grafico X vs Y
      Glcd_Image(@brujula_bmp)
      UART1_INIT(9600)
      Delay_ms(1000)
      muestras=0
      bit_inicio = 0x15
      while (muestras < 8)
        while(bit_confirmacion<>0x20)
          UART1_Write(bit_inicio)
          bit_confirmacion = uart1_read()
          Move_Delay()
        wend
        while (UART1_Data_Ready() = 0)
          wend
          coordenadaX = UART1_Read()
          while (UART1_Data_Ready() = 0)
            wend
            coordenadaY = UART1_Read()

          Glcd_Dot(coordenadaX, coordenadaY, 1)
      wend
  wend

if (tipo_grafico = 2) then  ' eligio grafico ANGULO vs TIEMPO
  LIMPIAR()
  Glcd_Image(@angulo_bmp)
  UART1_INIT(9600)
  Delay_ms(1000)
  muestras=0
  x1 = 16
  y1 = 54
  bit_inicio = 0x40
  while (muestras < 8)
    while(bit_confirmacion<>0x30)
      UART1_Write(bit_inicio)
      bit_confirmacion = uart1_read()
      Move_Delay()
    wend

    while (UART1_Data_Ready() = 0)
      wend
      angulo = UART1_Read()
      grados = (angulo * 48)/255
      x2 = x1 + 12
      y2 = 54 - grados
      Glcd_Line(x1 ,y1 ,x2, y2, 1)
      x1 = x2
      y1 = y2
      muestras= muestras +1
  wend

```

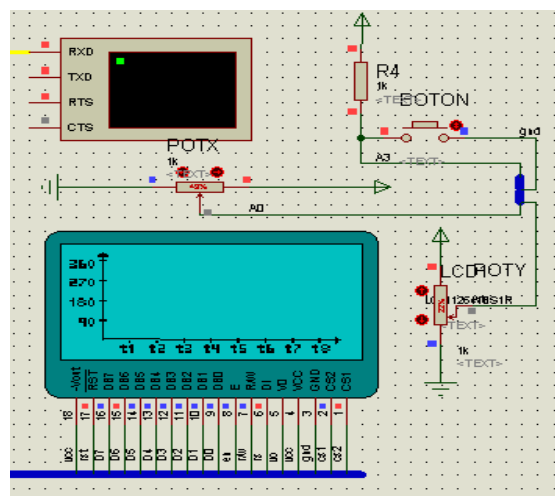
# CAPÍTULO 4

## 4. SIMULACIÓN Y PRUEBAS

### 4.1. Simulación en Proteus

A través del software de simulación Proteus y todas sus herramientas a utilizar pudimos realizar la simulación de nuestro proyecto.

En este caso la imagen a mostrar fue el eje grados vs tiempo, ya que Proteus no contiene una pantalla táctil entre sus elementos se implemento el panel táctil a través de dos potenciómetros y un pulsador.

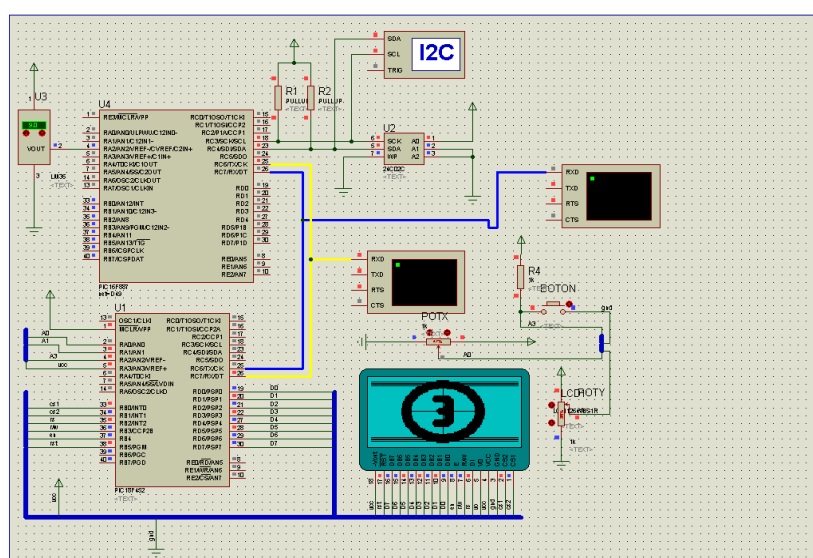


**Fig. 4.1 Potenciómetros para poder simular la parte del panel táctil**

Inicialmente realizamos la simulación de la interfaz gráfica para luego su posterior implementación. Como nuestro proyecto no captura muestras sino que solo las grafica, tuvimos que simular dos proyectos en conjunto para probar y alcanzar nuestros objetivos iniciales. Tomamos como referencia el proyecto realizado en las clases del seminario, dicho proyecto se encargaba de capturar



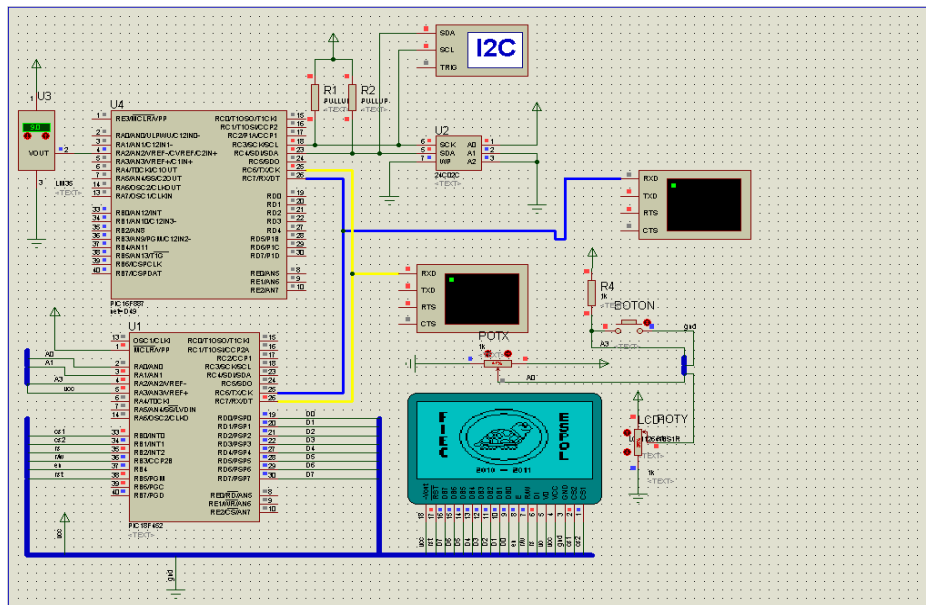
muestras de temperaturas en un intervalo de 1 hora por muestra. Después de un riguroso análisis, tuvimos la iniciativa de juntar estos dos proyectos para ir comprobando la comunicación serial entre los dos PICs y con esto llegamos a alcanzar las metas planteadas. La siguiente figura corresponde a una animación para interpretar como que si fuera el comienzo de alguna película dándole una mejor presentación al proyecto.



**Fig. 4.1.1** Secuencia de inicialización del proyecto

Las siguientes figuras corresponden a las diversas imágenes que hemos diseñado para una mejor presentación del proyecto.

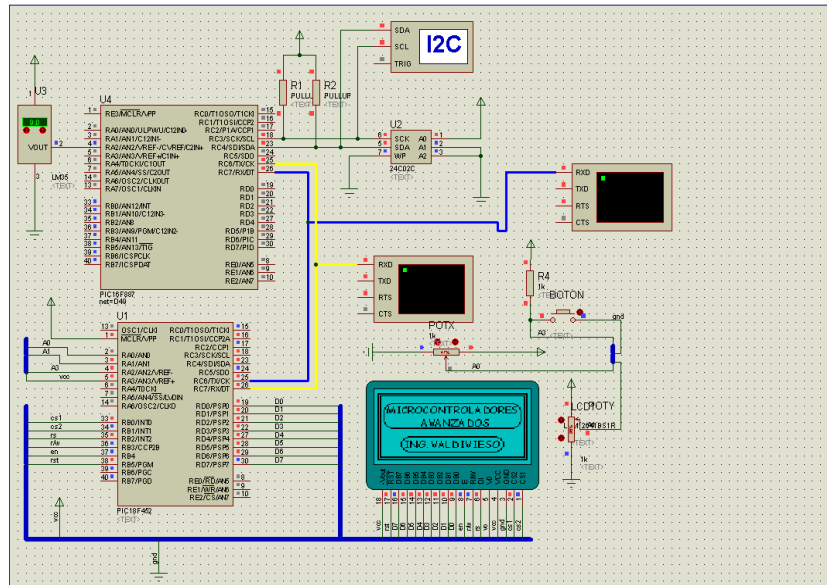
La primera imagen es un logo característico de la ESPOL que representan a la institución en donde hemos realizado el seminario de graduación.



**Fig. 4.1.2 Simulación de la imagen representativa de la institución**

Cabe indicar que para pasar a la siguiente imagen se debe presionar cualquier región de la pantalla táctil, ya que dentro de nuestro código fuente se programó para que se mantenga la imagen congelada hasta que reciba la pulsación del usuario.

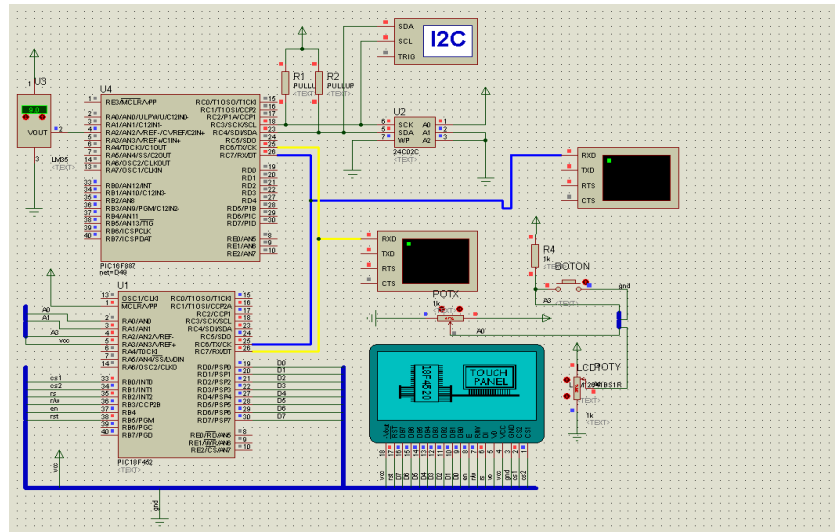
La segunda imagen es una presentación del tema de la materia de graduación (“MICROCONTROLADORES AVANZADOS”) y del profesor que imparte la cátedra (“ING. CARLOS VALDIVIESO”).



**Fig. 4.1.3 Simulación que muestra el nombre del Seminario de Graduación y del Profesor**

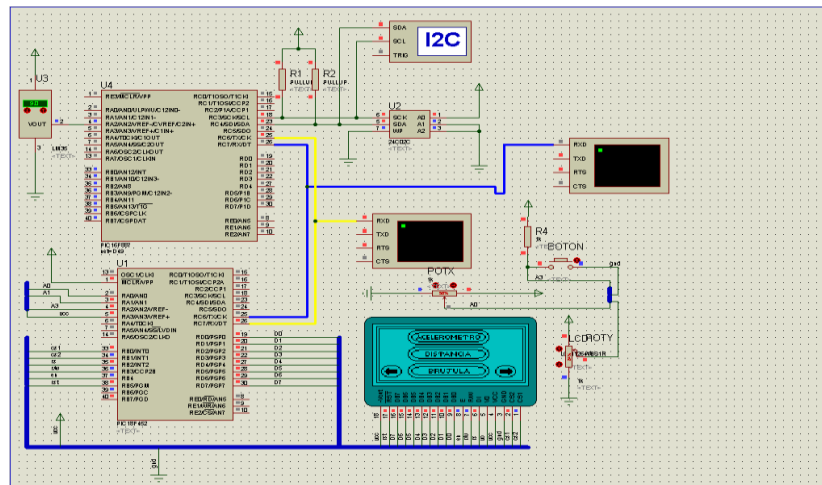
La tercera imagen es un dibujo esquemático de los diferentes componentes más significativos de nuestra interfaz gráfica.

Todos los gráficos fueron diseñados en el programa PAINT y son imágenes de formato bitmap, además que son de color blanco y negro debido a que la pantalla GLCD solo grafica imágenes de este tipo, sin embargo en la implementación de la pantalla GLCD se puede manipular los colores del fondo de la pantalla a través de los potenciómetros RGB y del contraste.



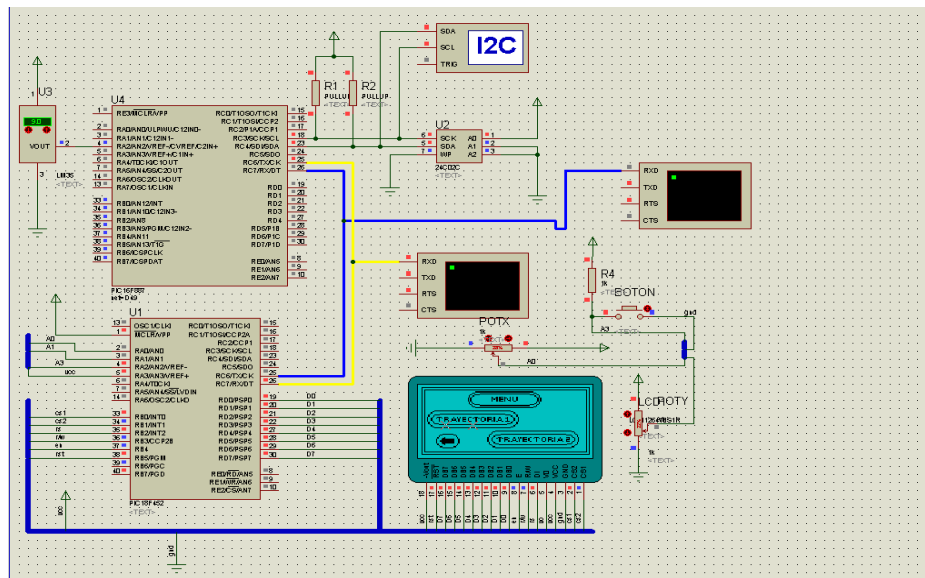
**Fig. 4.1.4 Simulación para mostrar en pantalla el diagrama de bloques del proyecto**

En la siguiente figura se muestra el menú con las opciones a escoger de los cinco proyectos a interconectarse con la interfaz gráfica, para mostrar los datos obtenidos de las diversas muestras.



**Fig. 4.1.5 Simulación que muestra sensores que el usuario desee escoger**

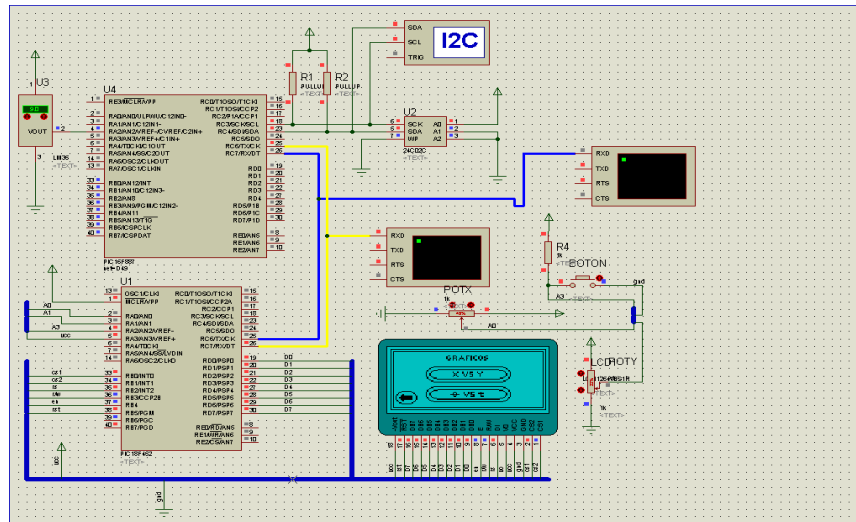
Al escoger la opción ACELEROMETRO se despliega un submenú que presenta dos trayectorias, donde puedo presentar los datos censados en dos ejes diferentes y así ir comparando muestras.



**Fig. 4.1.6 Simulación de las dos trayectorias del acelerómetro**

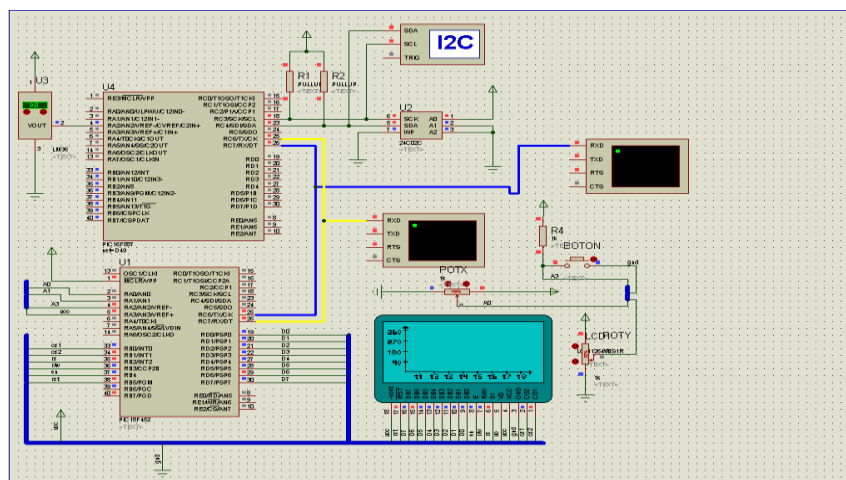
Después de escoger alguna de las dos trayectorias, continua con el siguiente submenú que trata de los dos diferentes tipos de gráficos que se van a presentar, el primero presenta las coordenadas X e Y en un plano cartesiano donde procedemos a graficar un par ordenado a la vez (X1, Y1).

El segundo presenta un eje donde se grafica el ángulo obtenido por cada par ordenado en función del tiempo, para luego analizar como varían las inclinaciones o muestras a medida que pasa el tiempo y así poder crear un modelo de cómo están distribuidas las muestras en un entorno.



**Fig. 4.1.7 Simulación de las mediciones X vs. Y, ángulo vs. Tiempo**

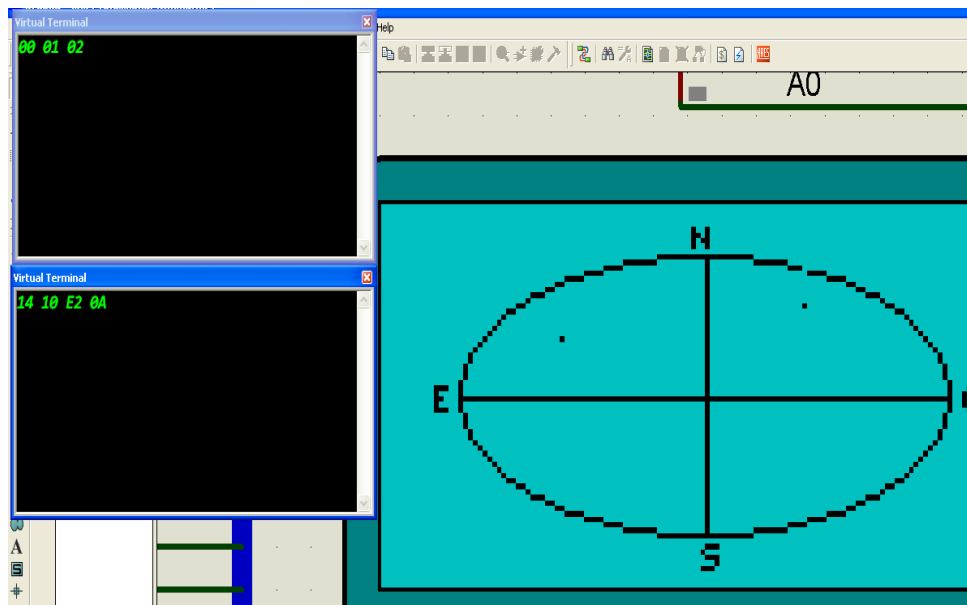
La siguiente figura corresponde a una de las dos opciones de gráficas que tiene la opción ACELEROMETRO del menú principal, en este caso se trata de mostrar como va variando el ángulo de inclinación a medida que pasa el tiempo.



**Fig. 4.1.8 Simulación del eje de coordenadas ángulo vs tiempo**

En la siguiente imagen se presenta la simulación de la gráfica de dos pares ordenados que representan las muestras capturadas por el proyecto BRUJULA, cabe recalcar que éstos datos fueron enviados a la interfaz gráfica por medio de comunicación serial, para sincronizar el flujo de datos decidimos enviar un bit

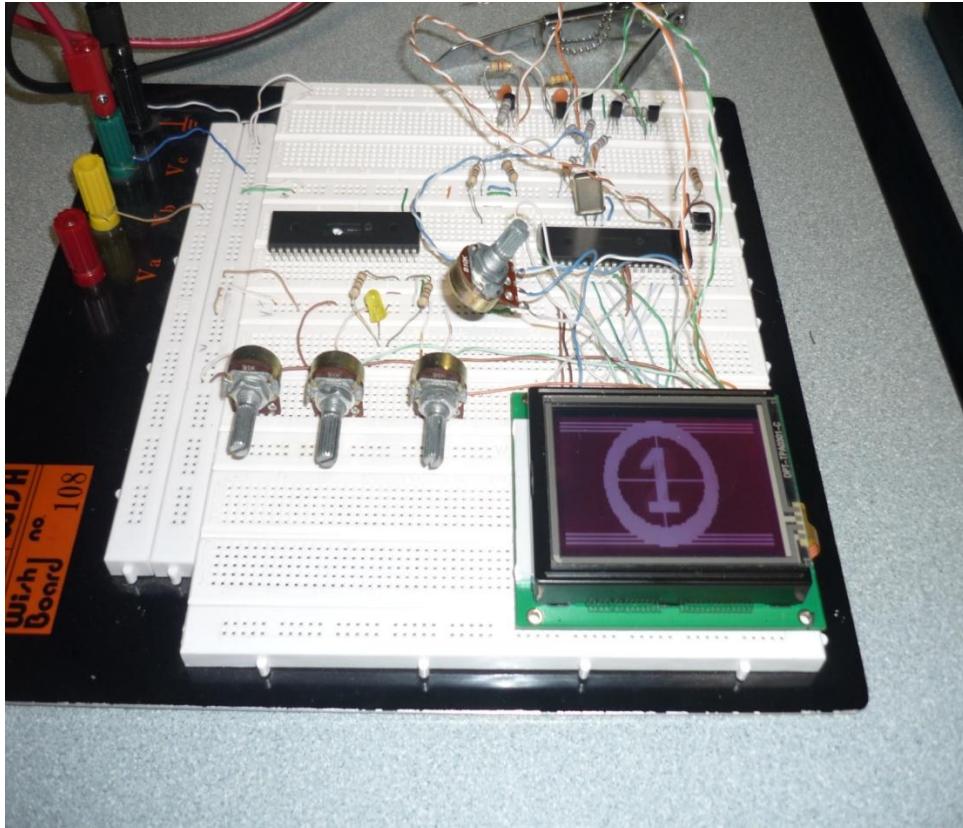
de inicio para empezar la transmisión y a su vez inicialmente recibir un bit de confirmación para verificar la sincronización de los dos proyectos y con esto logramos que no se pierda información. Decidimos representar los valores con puntos ya que hicimos diversas pruebas con líneas y se distorsionaba la presentación del gráfico, en cambio con puntos la presentación mejoró mucho.



**Fig. 4.1.9 Simulación de dos ángulos censados por la brújula magnética**

#### **4.2. Implementación en protoboard**

En la siguiente imagen se presenta la implementación del circuito controlador de la interfaz gráfica en protoboard, podemos apreciar en la pantalla GLCD una imagen de inicialización que proporciona una mejor presentación a nuestro proyecto.



**Fig. 4.2 Interfaz gráfica con pantalla GLCD táctil en protoboard**

#### **4.3. Implementación de la Tarjeta Electrónica PCB**

El diseño se realizó en PROTEUS y ARES de LAB CENTER ELECTRONICS.

Es un diseño elaborado mediante pistas a doble cara.



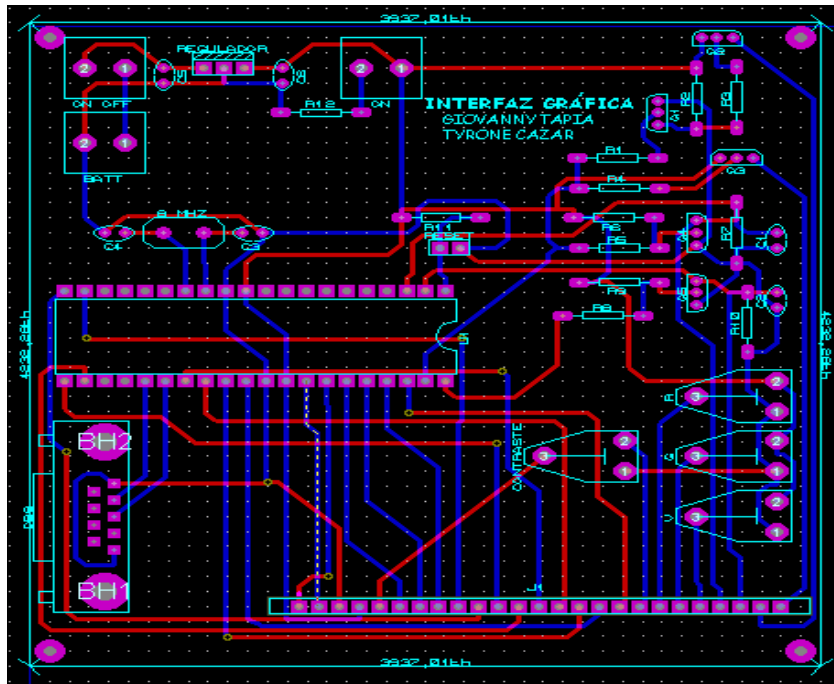


Fig. 4.3.1 Diseño de la tarjeta electrónica

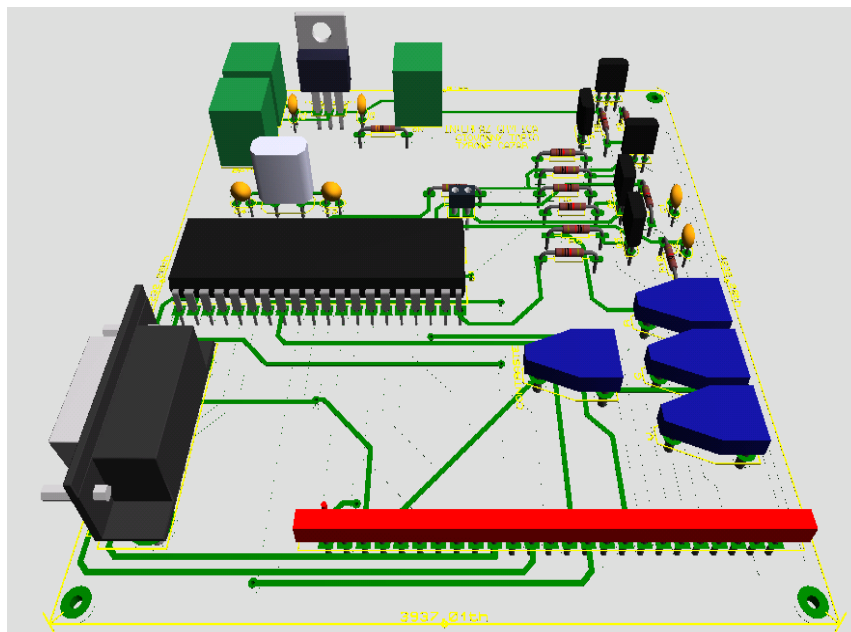


Fig. 4.3.2 Vista 3-D

# CONCLUSIONES

1.- Mediante nuestro proyecto hemos logrado implementar de una manera sencilla una interfaz gráfica que tiene una amplia gama de aplicaciones, permitiendo visualizar los datos de diferentes sensores obtenidos en cualquier lugar que queramos tomar muestras de campo, logrando una herramienta portátil de comparación y análisis de muestras.

2.- Una conclusión muy importante para la conexión del panel táctil la pudimos comprobar mediante experimentos y nos dimos cuenta que debe hacerse de una manera específica para que funcione correctamente porque sino toma coordenadas erróneas que provocan que nuestro código fuente colapse.

3.- Para la elaboración de nuestra interfaz gráfica usamos el PIC18F452 debido a que posee más cantidad de memoria que es indispensable para visualizar las imágenes y menús que son parte de la presentación de la materia de graduación de la ESPOL además que nuestro código fuente se encarga de la presentación de 5 proyectos los cuales demandan mucha memoria (2000 líneas).

4.- Para comunicarnos con los demás proyectos usamos la comunicación serial RS-232, ya que es asíncrona, tuvimos que crear una lista de comandos para

iniciar la transmisión, porque nosotros somos los encargados de controlar el flujo de datos, para lo cual definimos un bit de inicio para el envío de datos a la interfaz gráfica.

## **RECOMENDACIONES**

1.- Revisar y entender el manual de especificaciones de la pantalla táctil para su buen funcionamiento y de esta manera no cometer errores en la conexión de sus pines sobre todo en los que son para definir las coordenadas X y Y, a su vez también tener cuidado en conectar los pines de control de contraste y de colores.

2.- En la utilización de los transistores para el convertidor A/D, verificar que todos estos sean del mismo fabricante porque sino la ganancia de cada uno varía teniendo como resultado el mal funcionamiento del convertidor y a su vez del panel táctil.

3.- Una recomendación que cabe recalcar es leer detenidamente el datasheet del PIC18F452, para revisar la configuración de sus pines y así evitar cometer errores en las conexiones, logrando la prevención de quemar algún componente del proyecto.

## BIBLIOGRAFÍA

### SITIOS WEBS

1. CIRCUIT-ED, 128x64Q ALL-IN-ONE-TOUCHSCREEN-GLCD-With-RGB-LED- BACKLIGHT, <http://www.circuit-ed.com/128x64Q-ALL-IN-ONE-TOUCHSCREEN-GLCD-With-RGB-LED-BACKLIGHT-P168C8.aspx>, 30 de abril del 2010
2. DMC-CO-LTD, How Touchscreens Detect Where You Pressed, [http://www.dmccoltd.com/english/tecnical/logs/tec20041126\\_b.asp](http://www.dmccoltd.com/english/tecnical/logs/tec20041126_b.asp), 5 de mayo del 2010

3. ACTEL, Touchscreens Basics, [http://www.actel.com/documents/Fusion\\_TouchScreen\\_AB.pdf](http://www.actel.com/documents/Fusion_TouchScreen_AB.pdf), 5 de mayo del 2010
  
4. PLANET-ANALOG, Powering resistive touch screens efficiently, <http://www.planetanalog.com/article/printableArticle.jhtml?articleID=199702778>, 7 de mayo del 2010
  
5. EMBEDDED, Writing drivers For Common Touchscreen Interface Hardware, <http://www.embedded.com/columns/technicalinsights/164303539?requestid=86464>, 15 de mayo del 2010
  
6. CIRCUIT-ED, MECHANICAL LAYOUT, [http://circuit-ed.com/128x64\\_TS\\_RGB/12864STN\\_TSRGB.pdf](http://circuit-ed.com/128x64_TS_RGB/12864STN_TSRGB.pdf), 15 de Mayo del 2010
  
7. CIRCUIT-ED, GLCD/CONTROLLER SPECIFICATIONS, [http://circuit-ed.com/128x64\\_TS\\_RGB/GDM12864B.pdf](http://circuit-ed.com/128x64_TS_RGB/GDM12864B.pdf), 17 de mayo del 2010
  
8. CIRCUIT-ED, RGB BACKLIGHT SPECIFICATIONS, [http://circuit-ed.com/128x64\\_TS\\_RGB/RGB\\_BL\\_SPEC.pdf](http://circuit-ed.com/128x64_TS_RGB/RGB_BL_SPEC.pdf), 17 de mayo del 2010
  
9. MIKROE, TouchScreen Article, <http://www.mikroe.com/eng/products/view/272/touchscreen-article/>, 17 de mayo del 2010

10. **WINMATE, Compare All Resistive Touch Technologies**

[http://www.winmate.com.tw/resistive\\_touch.htm](http://www.winmate.com.tw/resistive_touch.htm), 2 de mayo del 2010