ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

"MÓDULO DE ORIENTACIÓN MAGNÉTICA UTILIZADO PARA LA CAPTACIÓN DE ÁNGULOS DE ORIENTACIÓN CON CAPACIDAD DE COMUNICACIÓN SERIAL A DATALOGGER E INTERFAZ GRÁFICA"

TESINA DE SEMINARIO

Previa a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

MISAEL ENRIQUE RAMÍREZ CASTILLO

RODRIGO ALBERTO LINO BORBOR

GUAYAQUIL – ECUADOR

2010

DEDICATORIA

A Dios y a mis padres.

Misael Enrique Ramírez Castillo

A Dios y a mi madre Oty.

Rodrigo Alberto Lino Borbor

AGRADECIMIENTO

Al ING. CARLOS
VALDIVIESO y a todas
aquellas personas que
depositaron su esfuerzo y
su confianza en la
culminación del presente
informe.

TRIBUNAL DE SUSTENTACIÓN

Ing. Carlos Valdivieso A.

PROFESOR DEL SEMINARIO DE GRADUACIÓN

Ing. Hugo Villavicencio

DELEGADO DEL DECANO

DECLARACIÓN EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestas en esta tesina de seminario, nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral"

(Reglamento de exámenes y títulos profesionales de la ESPOL)

____

Misael Enrique Ramírez Castillo

____

Rodrigo Alberto Lino Borbor

RESUMEN

La brújula es un instrumento que sirve para determinar una dirección sobre la superficie terrestre por medio de una aguja imantada que siempre señala hacia el polo magnético norte. En este proyecto se utiliza un módulo compás que presenta un funcionamiento similar a una brújula, con la diferencia que puede aportar con una salida digital de la orientación con respecto al campo magnético terrestre.

Para el manejo de la brújula digital es necesario el uso de un microcontrolador, que es un circuito integrado programable que contiene todos los componentes de un computador, aunque de limitadas prestaciones. Este microcontrolador servirá de interfaz entre el módulo compás y otro circuito destinado a alguna aplicación que requiera de una entrada de orientación.

Sintetizando, este proyecto se encargará del manejo y control de las señales de una brújula digital a través de un microcontrolador, donde posteriormente se utilizará las entradas otorgadas por la brújula digital en aplicaciones de almacenamiento e interfaz gráfica.

ÍNDICE GENERAL

INTRODUCCIÓN

CAPITULO I

1.- Generalidades

1.1.- Alcance y limitaciones del proyecto……………………………………………1

1.2.- Antecedentes…………………………………………………………………..2

1.3.- Identificación del problema……………………………………………………..2

1.4.- Descripción breve de la solución…………………………………………….3

1.5.- Razón para escoger las herramientas usadas en este proyecto……….3

1.6.- Soluciones similares del sensor...……………………………………………..4

CAPITULO II

2.- Sustento Teórico

2.1.- Conceptos Generales del módulo HM55B…...……………………………….8

2.1.1.- Recomendación del sensor a ser usado………………………………9

2.1.2.- Principio de funcionamiento del HM55B……………………………..10

2.1.3.- Pines de conexión……………………………………………………...12

2.2.- El Microcontrolador……………………………………………………………15

2.2.1.- Características generales……………………………………………...16

2.3.- Librería LCD……………………………………………………………17

2.3.1.- Configuración del LCD…………………………………………………18

2.4.- Ambiente MikroBasic Pro utilizado……………………………………..19

CAPITULO III

3.- Diseño e Implementación

3.1.- Diagrama de bloques y de flujo……………………………………………23

3.2.- Descripción detallada de cada bloque……………………………………24

3.2.1.- Bloque Sensor…………………………………………………………...24

3.2.2.- Bloque base de datos……...………………………………………….25

3.2.3.- Bloque presentación de datos…………………………………………26

3.2.3.1.- Presentación por Lcd……………………………………………26

3.2.3.2.- Gráficas mostradas en el proyecto del GLCD………………... 28

3.2.4.- Bloque de control y procesamiento……………………………………29

3.3.- Programación del 18F4431….…………………………………………..31

CAPITULO IV

4.- Simulación y resultados

4.1.- Circuito simulado del proyecto con sus bloques principales……………37

4.2.- Programa que simula al módulo compás Hitachi HM55B…...…………38

4.3.- Gráficas de la simulación en Proteus…………………….……..………….39

4.4.- Simulación de la comunicación con el datalogger………….………….. 43

4.5.- Pistas del PCB construidos………………………………………….…. 45

4.6.- Resultados experimentales…………………………………………….. 47

CONCLUSIONES Y RECOMENDACIONES

ANEXO A

MANUAL DE USUARIO

1.- INTRODUCCIÓN

2.- ANTES DEL ENCENDIDO

3.- FUNCIONAMIENTO Y PUESTA EN MARCHA

ANEXO B

CÓDIGO FUENTE PRINCIPAL

CÓDIGO FUENTE SENSOR

BIBLIOGRAFÍA

ÍNDICE DE FIGURAS

CAPÍTULO 1

FIGURA 1.1 HM55B Hitachi Compas Module …………………………………. 4

FIGURA 1.2 HMC1051/HMC1052L/HMC1053…………………………………. 5

FIGURA 1.3 SPARTON SP3002D-4D KIT COMPASS DEVELOPMENT KIT.... 6

CAPÍTULO 2

FIGURA 2.1 Sensor HM55B……………………………………………………….8

FIGURA 2.2 Principio de funcionamiento del HM55B…………………………10

FIGURA 2.3 Ejemplo de uso……………………………………………………..11

FIGURA 2.4 Interface 3 hilos……………………………………………………..11

FIGURA 2.5 Pines del sensor……………………………………………………12

FIGURA 2.6 Pic 18F4431…………………………………………………………15

FIGURA 2.7 Características Pic 18F4431………………………………………16

FIGURA 2.8 XTAL para 18F4431………………………………………………..16

FIGURA 2.9 Pines del 18F4431…………………………………………………17

FIGURA 2.10 LCD TS1620A……………………………………………………..17

FIGURA 2.11 Configuración LCD………………………………………………..19

FIGURA 2.12 Creación de nuevo programa……………………………………19

FIGURA 2.13 Inicio de compilado……………………………………………….20

FIGURA 2.14 Presentación encabezado del programa……………………….21

FIGURA 2.15 Presentación Build………………………………………………..21

FIGURA 2.16 Presentación de mensajes……………………………………….21

FIGURA 2.17 Ficheros generados………………………………………………22

CAPÍTULO 3

FIGURA 3.1 Diagrama de bloques………………………………………………23

FIGURA 3.2 Entradas del sensor………………………………………………..25

FIGURA 3.3 Datos ingresados…………………………………………………...26

FIGURA 3.4 Menú 1……………………………………………………………….27

FIGURA 3.5 Menú 4……………..………………………………………………..27

FIGURA 3.6 Menú 3……………..………………………………………………..28

FIGURA 3.7 Gráfica XY ………..………………………………………………..29

FIGURA 3.8 Gráfica ángulo vs tiempo ………………………………………..29

FIGURA 3.2 Diagrama de flujo del procedimiento principal .………………..31

FIGURA 3.10 Salidas del Pic……………………………………………………32

CAPÍTULO 4

FIGURA 4.1 Esquemático en Proteus………………………………………….37

FIGURA 4.2 Simulación del sensor……………………………………………..39

FIGURA 4.3 Simulación salida del sensor…………………………………….40

FIGURA 4.4 Simulación comando measure……………………….…………..41

FIGURA 4.5 Simulación envío datos X e Y……………………….……………42

FIGURA 4.6 Interfaz MAX232 …………………………………………………. 43

FIGURA 4.7 Simulación de la comunicación serial …………………………. 44

FIGURA 4.8 Diseño PCB ……………………………………………………… 45

FIGURA 4.9 Prototipo ……..…………………………………………………… 46

FIGURA 4.10 Comparación entre brújulas ……………………….………… 47

FIGURA 4.11 Ubicación del sensor …………..……………………………… 48

INTRODUCCIÓN

En este trabajo se describe el diseño y construcción, de un sistema de orientación por medio de una brújula electrónica. Las coordenadas proporcionadas por el sensor se guardarán en una memoria USB y se mostrarán en una pantalla LCD.

En el capítulo 1 se describen, el alcance así como las limitaciones del proyecto, los antecedentes y las razones de la implementación del mismo.

En el capítulo 2 se fundamenta la teoría de los elementos involucrados como: características del sensor magnético, manejo de librerías y tipos de datos del compilador utilizado, características usadas y comunicación serial del microcontrolador así como del manejo de librerías y funciones de la pantalla LCD.

En el capítulo 3 se realiza el diseño del proyecto. El diagrama de bloques y la descripción detallada de cada bloque y la programación del microcontrolador.

Finalmente en el capítulo 4 se simula e implementa el circuito planteado para establecer conclusiones y realizar las correcciones finales para la realización del grabado del circuito en la placa.

CAPÍTULO 1

En este capítulo se aborda el planteamiento del proyecto así como sus alcances y limitaciones. Se expone además la razón de esta implementación, así como la explicación de la solución al problema planteado; finalizando con los proveedores que pueden proporcionar soluciones similares.

1.1.- Alcance y limitaciones del proyecto

Implementación con un compás digital y tecnología 3 WIRE, capaz de presentar las siguientes características:

- Las variaciones de campos magnéticos en la superficie de la tierra.
- Capacidad de almacenamiento de la información en un datalogger utilizando comunicación serial.

- Visualización inmediata de la medición tomada a través de una pantalla LCD.
- Medición periódica de la orientación en un intervalo de 30 segundos, para la posible aplicación real en una embarcación, pero en virtud del tiempo para exponer este informe se redujo a 5 segundos.

1.2.- Antecedentes

Desde tiempos inmemorables, el hombre ha hecho uso de las herramientas que hasta ese entonces ha tenido para su orientación, tal es el caso del sol, las estrellas y la luna, que aún siguen siendo iconos de ubicación geográfica, pero debido al desarrollo de la tecnología y de las múltiples necesidades de buscar métodos alternativos de orientación es que se ha evolucionado en la fabricación de instrumentos ajustados a la época actual en donde la electrónica juega un papel muy preponderante.

1.3.- Identificación del problema

La tecnología actual ha dejado muy rezagada a las brújulas ordinarias reemplazándolas por el GPS (Sistema de Posicionamiento Global); pero estos aparatos a pesar de ser muy exactos, son muy caros y de difícil acceso. En Ecuador, las aplicaciones de instrumentos electrónicos de posicionamiento son usados principalmente para el rastreo y recuperación de vehículos robados.

Los navíos y algunos otros transportes marítimos y aéreos tienen y requieren de sistemas para saber hacia dónde se dirigen ya sea para el norte, sur o algún otro punto. Aclarando que no todo es 100 % confiable, se correría el riesgo de que ocurrieran muchos accidentes si es que falla el GPS.

1.4.- Descripción breve de la solución.

En Ecuador hay una falta de servicios de sistemas de orientación y posicionamiento. De ahí que se pretende realizar un sistema de orientación (brújula electrónica) que permita tener un margen de error muy pequeño y que su precisión sea mayor comparándola con una brújula ordinaria. Es por ello que se diseñó y se construyó un prototipo capaz de obtener la dirección más precisa y exacta que una brújula ordinaria y, que el costo no sea tanto como es la tecnología GPS, pudiendo ser el sistema propuesto considerado también de emergencia donde los aviones o barcos la usen para reafirmar su dirección.

Esta brújula está montada sobre una pequeña tarjeta que permite la fácil conexión con el resto de componentes electrónicos necesarios.

1.5.- Razón para escoger las herramientas usadas en este proyecto

Mikrobasic Pro v3.2 es un software que permite diseñar y programar en un lenguaje conocido como es el Basic, el cual es una herramienta que permite trabajar con múltiples librerías que minimizan la extensión y complejidad del programa.

Los Microcontroladores, son pequeños chips que poseen características especiales, los cuales son programables, con funciones de un computador tales como procesamiento, memoria y almacenamiento. Tal es el caso del PIC18F4431 de MICROCHIP que dada sus características y gran almacenamiento lo hemos escogido para la interacción con el módulo HM55B.

El modulo Hitachi HM55B nos muestra de manera objetiva la toma de datos, en donde su fabricante PARALLAX nos brinda documentación comprensiva para la programación de sus sensores.

1.6.- Soluciones similares del Sensor

Encontrar empresas dedicadas a la fabricación de módulos de orientación magnética no es muy común en nuestro país, sin embargo podemos hacer pedidos a las siguientes empresas en el Ecuador:

Electroavil-es , Empresa dedicada a la venta de circuitos integrados y materiales electrónicos. Además, presta servicio de compras en el extranjero de equipos electrónicos, herramientas y libros.

Figura 1.1 HM55B Hitachi Compas Module

En la figura 1.1 se muestra el Compás magnético propuesto para este proyecto y proporcionado por Electroavil-es. Diseñado sobre todo para robótica, se comunica con el microcontrolador por medio del protocolo serial sincrónico.

Entre los sistemas ofertados por fabricantes extranjeros de módulos de orientación magnética tenemos por ejemplo a:

Honeywell Aerospace Plymouth, que es una empresa proveedora de circuitos electrónicos a industrias en telecomunicaciones y aéreas, es también conocida como fabricante de sensores de precisión magnético para una amplia variedad de aplicaciones comerciales.

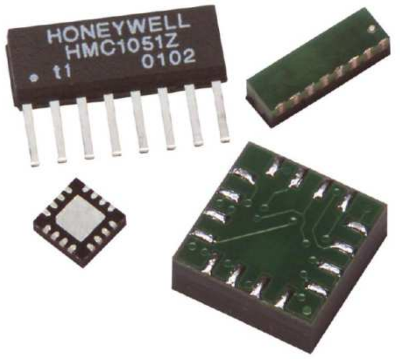


Figura 1.2 HMC1051/HMC1052L/HMC1053

En la figura 1.2 se muestran los una familia de sensores de campos magnéticos de alta sensibilidad proporcionados por esta empresa. Pueden ser usados en ambientes de bajo y fuerte campo magnético. Poseen 1, 2 y hasta 3 axis.

Spartons Electronics, es una empresa dedicada al diseño y fabricación de equipos médicos y equipos para submarinos de guerra, además provee a múltiples industrias con circuitos electrónicos, así como de sensores magnéticos.

Figura 1.3 SPARTON SP3002D-4D KIT COMPASS DEVELOPMENT KIT

Es un kit que provee un conector RS232 para la interacción directamente con la Pc a través de un software. El kit provee una brújula que funciona en base a campos magnéticos.

NOTA:

Muchas de las soluciones ofrecidas por los fabricantes extranjeros están disponibles, solo hay que considerar además el costo del envío.

CAPÍTULO 2

En este capítulo se presenta una descripción completa del sensor HM55B para una brújula electrónica, que es uno de los sensores que más se ocupan para este tipo de aplicaciones de orientación. También cuenta con una explicación breve del microcontrolador 18F4431 y de la pantalla LCD. Finalmente, se tratará brevemente acerca del software MikroBasic Pro v3.2 que es el software utilizado para programar nuestro Pic.

2.1.- CONCEPTOS GENERALES DEL MÓDULO HM55B

Figura 2.1 Sensor HM55B

El módulo Hitachi HM55B es un sensor de campo magnético de dos coordenadas (X e Y) ideal para aplicaciones en donde se necesite visualizar o controlar la dirección, como por ejemplo un robot que siga el punto cardinal que se le ha designado como destino. Es capaz de detectar variaciones bajo la exposición de un campo magnético. Este producto fue asignado por PARALLAX Inc. en un cómodo empaquetado DIP con regulador de voltaje compatible con 5Vdc.

2.1.1.- Recomendación del sensor magnético para ser usado en este proyecto.

Este tipo de sensor magnético (HM55B), se ajusta a nuestros requerimientos ya que podemos lograr obtener un determinado grado con solo girar la brújula magnética. Con relación al costo, el sensor magnético se adapta a la perfección a nuestras limitaciones del informe, ya que para el tamaño del proyecto, es ideal.

> **NOTA:** La calibración del sensor, con las especificaciones dadas por el fabricante, solo se realiza en software; es decir que el sensor en si no presenta cambio alguno, sino que se ajustan los ejes X e Y dentro de la ejecución del programa del microcontrolador.

2.1.2.- PRINCIPIO DE FUNCIONAMIENTO DEL HM55B

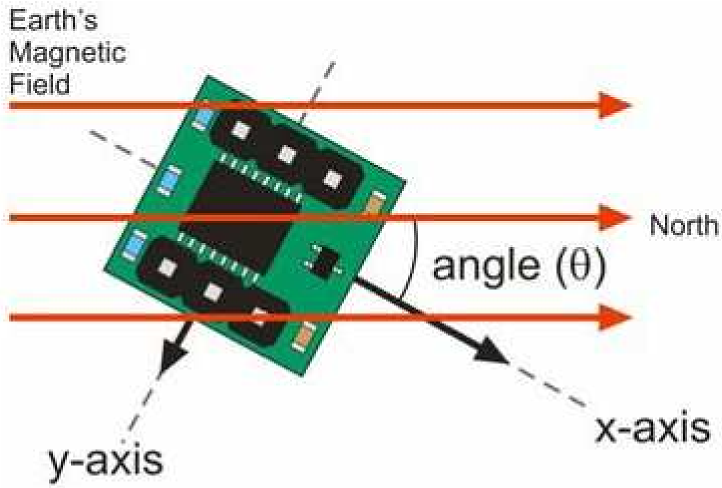


Figura 2.2 Principio de funcionamiento

La brújula digital Hitachi HM55B tiene dos coordenadas X e Y. Cada coordenada X reporta (X.Cos (θ)), y la coordenada Y reporta (Y. Sen (θ)). Para hallar "θ" (que es el ángulo de inclinación con la parte frontal del aparato y el Norte en sentido horario) se usa la función ATAN (-Y/X) con lo cual obtenemos fácilmente el punto cardinal hacia el cual nos dirigimos, a través de una escala de 0 a 360° donde el cero representa al Norte Geográfico.

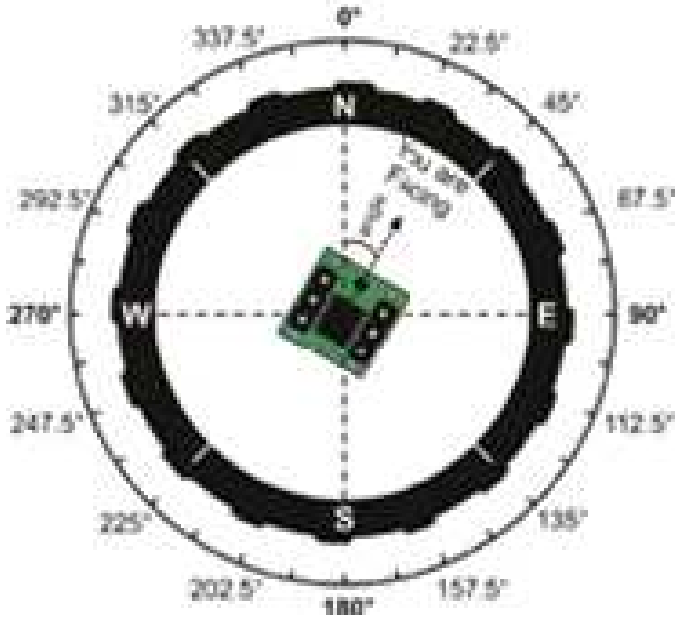


Figura 2.3 Ejemplo de uso

El módulo compás usado, utiliza comunicación serial síncrona y tiene la posibilidad de ser conectado como 3 WIRE y como 4 WIRE; este informe se basó en el empleo de la primera aplicación.

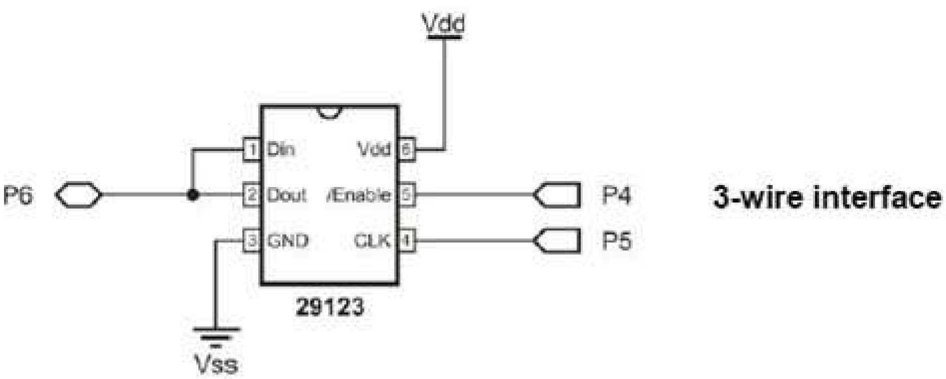


Figura 2.4 Interface 3 hilos

2.1.3.- Pines de Conexión:

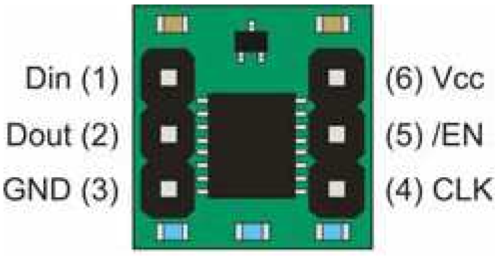


Figura 2.5 Pines del sensor

Pin (1): Entrada de datos seriales síncronos

Pin (2): Salida de datos seriales síncronos

Pin (3): Tierra

Pin (4): Reloj de sincronía

Pin (5): Entrada de habilitación

Pin (6): Voltaje de alimentación Vcc

Para el programa desarrollado se utilizaron las siguientes conexiones entre el PIC y el sensor:

PIC 18F4431	Hitachi HM55B
Pin 6	Pin 5
Pin 5	Pin 4
Pin 4	Pin 1
Pin 4	Pin 2

Set de Comandos

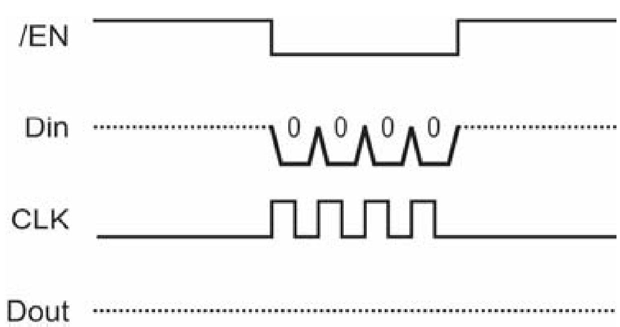
Comando	Significado
0000	Reset
0001	Start
0011	Pedir reporte de estado

Banderas de Estado:

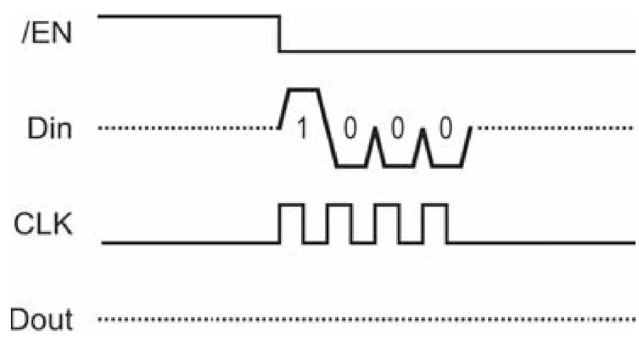
Código	Significado
1100	Medición exitosa
00XX	Medición en proceso o reseteado
XX11	Pedir reporte de estado

Protocolo de comunicación:

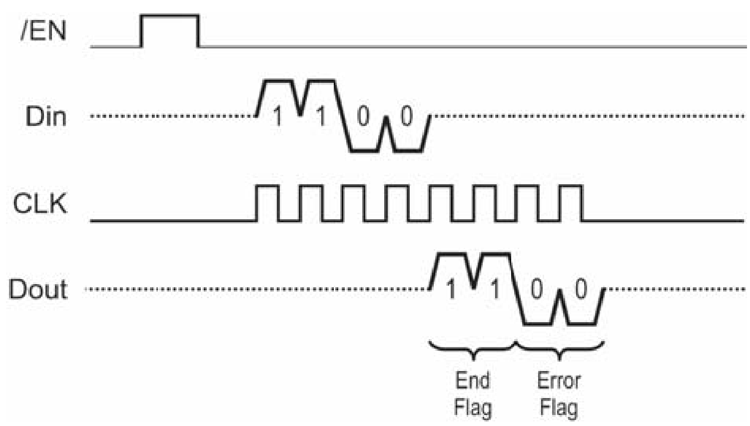
Para Resetear el sensor:



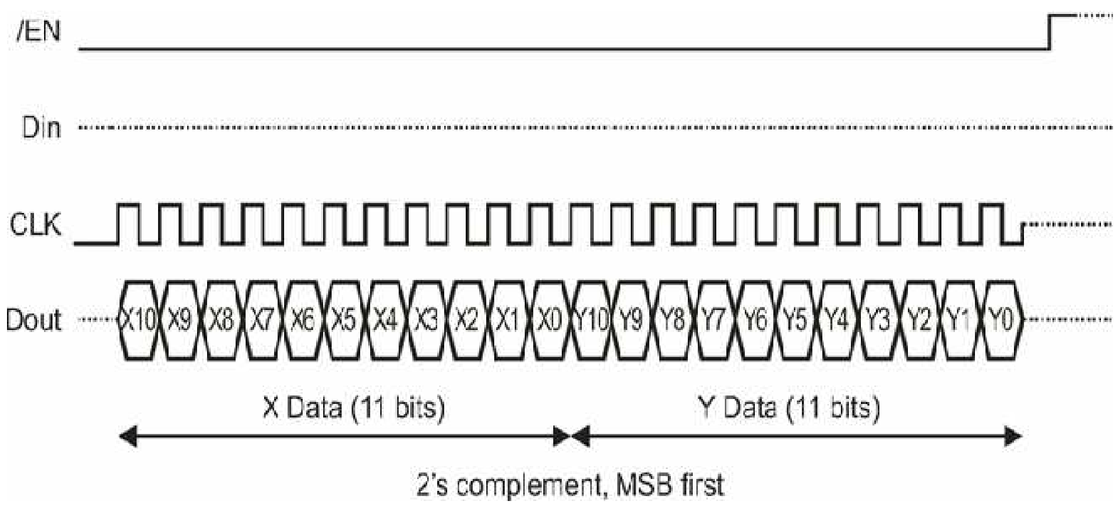
Para Empezar la Medición:



Para la Solicitud y Validación del Estado de la Medición:



Para la Recepción de los Valores Asociados a las Coordenadas X e Y:



2.2.- MICROCONTROLADOR 18F4431

Figura 2.6 Pic 18F4431

El 18F4431 es un circuito integrado fabricado por la empresa Microchip, que funciona como un controlador. Está diseñado para servir como una interface para una comunicación entre cualquier dispositivo electrónico que cuente con comunicación serial.

Algunas características: 40 pines (36 entradas o salidas), 8 canales PWM, conversor A/D de 9 canales, oscilador interno 8 MHz, comunicación USART.

2.2.1.- Características generales:

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP	SSP		EUSART	Quadrature Encoder	14-Bit PWM (ch)	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Slave I^2C^{TM}				
PIC18F2331	8192	4096	768	256	24	5	2	Y	Y	Y	Y	6	1/3
PIC18F2431	16384	8192	768	256	24	5	2	Y	Y	Y	Y	6	1/3
PIC18F4331	8192	4096	768	256	36	9	2	Y	Y	Y	Y	8	1/3
PIC18F4431	16384	8192	768	256	36	9	2	Y	Y	Y	Y	8	1/3

Figura 2.7 Características 18F4431

Es importante acotar que el microcontrolador es polarizado con 5V en Vdd y con 0V en Vss. Aparte, se necesita conectarlo a Vdd en el pin1 /MCLR. . Por último, el cristal oscilador de 8 MHz para este proyecto cuenta con la siguiente configuración:

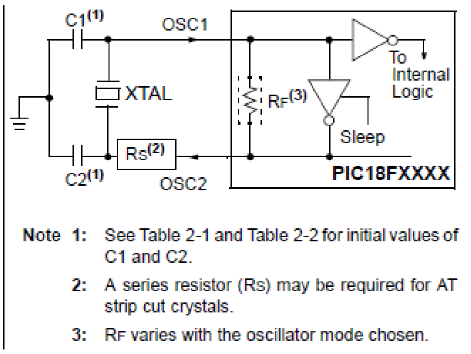


Figura 2.8 XTAL para 18F4431

En donde los capacitores de acoplamiento son de entre 18 y 33 pF.

Usaremos para el cristal de 8 MHz, dos capacitores de 22 pF.

La posición de sus pines de interface es:

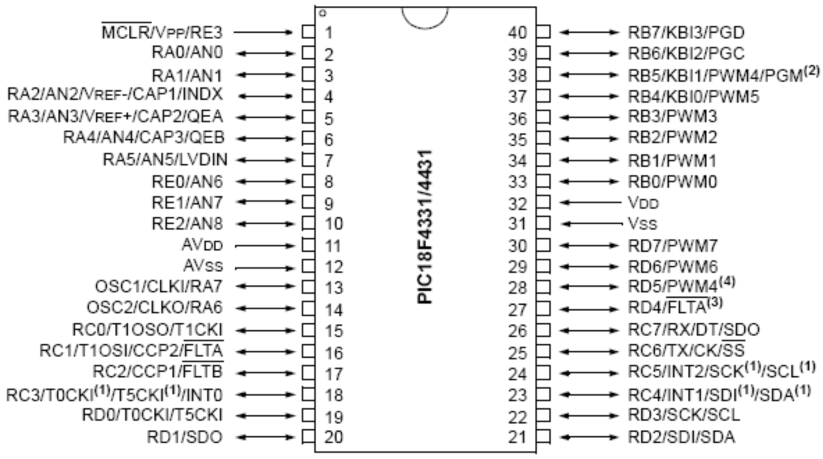


Figura 2.9 Pines 18F4431

Este poderoso microcontrolador tiene muchas características importantes, pero solo nombraremos en el presente escrito las usadas.

2.3.- LCD

Figura 2.10 LCD TS1620A

Muchas aplicaciones que usan microcontroladores requieren mostrar datos de diversas formas. Para ello se puede emplear fácilmente un display LCD. Estos módulos son la solución ideal en los casos donde se desea mostrar menús al usuario, respuestas a determinadas secuencias de comandos, para lo cual el hardware de control se resume en un par de teclas. También son muy útiles en sistemas de mediciones múltiples y simultáneas, donde de otra forma habría que emplear cantidades de decodificadores BCD y transistores para comandar displays de 7 segmentos convencionales.

2.3.1.- Configuración del LCD:

La conexión del módulo LCD y el circuito puede realizarse por medio de un cable plano de 14 hilos, similar al que se emplea en los discos duros. Admitiéndose que el display está conformado por un controlador del tipo HD44780, de la empresa japonesa Hitachi. Si bien los terminales no son normalizados, los tipos de señal manejados por ellos son casi estándar, por lo que no hay casi diferencia entre cada uno de ellos. Puede variar uno que otro comando, pero no el cableado del módulo en lo que a señales se refiere.

Pin	Símbolo	E/S	Función
1	Vss	-	0V (Tierra)
2	Vdd	-	+5V ± 0.25V (Tensión positiva de alimentación)
3	Vo(*)	-	Tensión negativa para el contraste de la pantalla
4	RS	E	Selector de Dato/Instrucción*
5	R/W*	E	Selector de Lectura/Escritura*
6	E	E	Habilitación del módulo
7	DB0	E/S	BUS
8	DB1	E/S	

9	DB2	E/S	DE DATOS
10	DB3	E/S	
11	DB4	E/S	
12	DB5	E/S	
13	DB6	E/S	
14	DB7	E/S	

Figura 2.11 Configuración LCD

2.4.- AMBIENTE DE MIKROBASIC PRO UTILIZADO EN EL PROYECTO.

De una manera muy corta y sencilla posible estableceremos los pasos para crear y compilar un nuevo proyecto en MikroBasic:

1.- Seleccionamos nuevo programa:

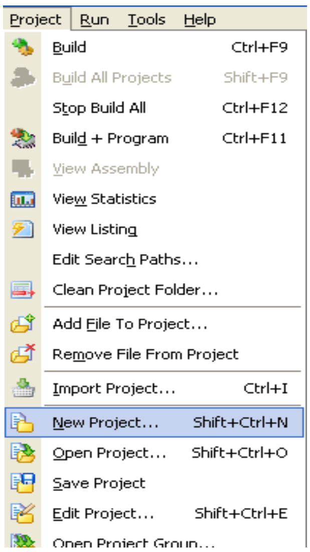


Figura 2.12 Creación de nuevo programa

2.- Aparecerá una ventana como la que sigue que le guiará en el proceso de compilado:

Figura 2.13 Inicio de compilado

El proceso de creación de un nuevo proyecto consiste en cinco pasos en total:

a) Seleccionar el PIC a trabajar. En nuestro caso es el 18F4431.

b) Seleccionar la frecuencia de reloj del microcontrolador. La frecuencia nuestra es de 8 MHz.

c) Seleccionar el nombre y la ruta del proyecto, al cual se le asignará automáticamente la extensión ***.mbppi.***

d) Si el programa consiste en varios ficheros habrá que incluirlos. En este informe no hay ficheros adicionales.

e) Por último se necesita confirmar las opciones seleccionadas haciendo clic en ***Finish***.

Luego aparecerá la ventana donde escribiremos el programa:

```
' * Project name:
'     Módulo de Orientación Magnética utilizado para la captación de Ángulos
'     de Orientación con capacidad de comunicación serial a Datalogger e
'     Interfaz Gráfica.
' * Copyright:
'     (c) Mikroelektronika, 2010.
' * Revision Historica:
'
'       - inicio       : 03-04-2010
'       - modificación: 19-04-2010
' * Descripción:
'     Es un sensor del ángulo de orientación con respecto al campo magnético:
'
' * Prueba de configuración:
'     MCU:              PIC18F4431
'                       http://ww1.microchip.com/downloads/en/DeviceDoc/41291F.pdf
'     Dev.Board:        EasyPIC6
'                       http://www.mikroe.com/en/tools/easypic6/
'     Oscillator:       HS, 08.0000 MHz
```

Figura 2.14 Presentación encabezado programa

Una vez terminado el programa le ponemos un nombre, se lo guarda y compila:

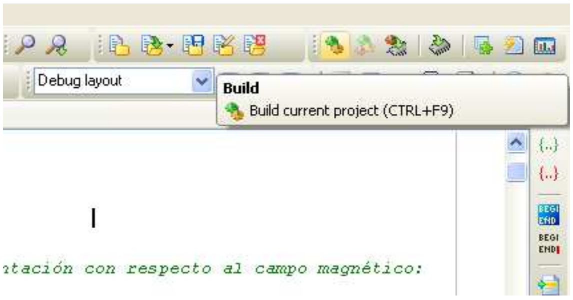


Figura 2.15 Presentación Build

Todos los errores detectados durante la compilación aparecerán en la ventana ***Messages.***

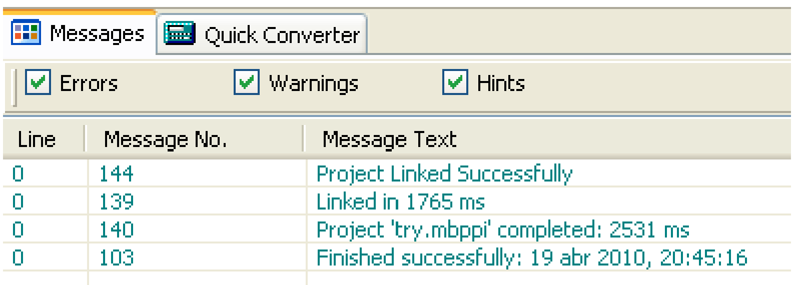


Figura 2.15 Presentación de Messages

Si no hay errores, el MikroBasic Pro generará ficheros de salida, de los cuales cargaremos en el PIC el ***.hex***:

Formato	Descripción	Tipo de fichero
Intel HEX	Registros del formato Intel hex. Este fichero se utiliza para la programación de los microcontroladores PIC.	.hex
Fichero Binario	Librería compilada que se puede incluir en otros proyectos.	.mcl
Fichero Ensamblador	Fichero ensamblador con los nombres simbólicos	.asm
Fichero List	Visión general de uso de la memoria de los microcontroladores PIC. El fichero List representa una versión extendida del código ensamblador, es decir, contiene las direcciones de instrucciones, registros, rutinas y etiquetas.	.lst

Figura 2.16 Ficheros generados

CAPÍTULO 3

3.- Diseño e Implementación.

En este capítulo de procederá al diseño e implementación del proyecto. Para la explicación se ha elaborado un diagrama de bloques así como también su diagrama de flujo. Explicaciones detalladas se darán a continuación.

3.1.- Diagrama de bloques

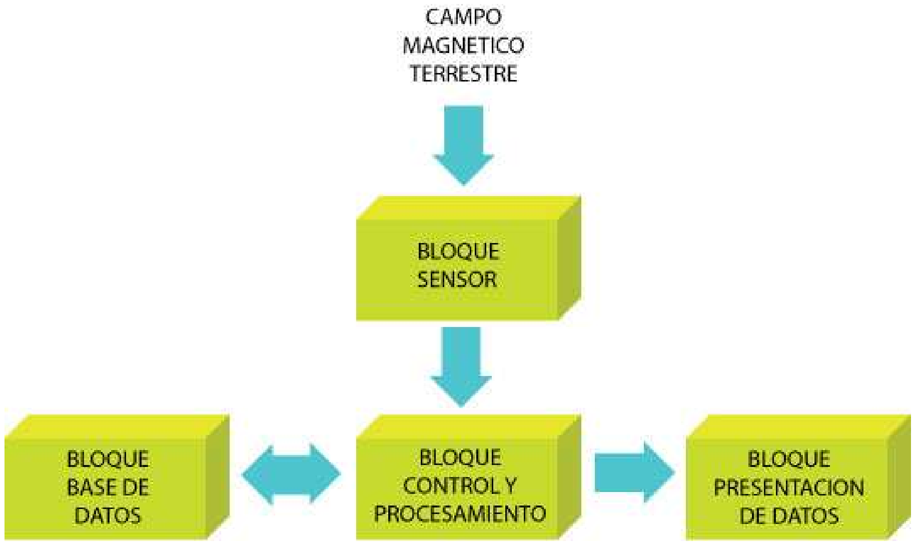


Figura 3.1 Diagrama de bloques

3.1.1.- Explicación breve del funcionamiento del circuito general.

El presente proyecto presenta como finalidad tomar datos de orientación magnética a través del módulo compas Hitachi. Estos datos serán tomados preferiblemente en un espacio libre de emanaciones magnéticas, evitando así errores en las mediciones tomadas por el sensor. A medida que se toman los datos, estos podrán ser apreciados inmediatamente en un Lcd conectado al microcontrolador.

Posteriormente, estos datos serán almacenados en la memoria volátil del microcontrolador, donde se presentará la opción de poder conectarse por medio de una conexión serial UART a uno de los dispositivos siguientes:

- Datalogger, que nos servirá como un banco de datos para enviar y recibir las muestras.
- Lcd, donde se presentarán los datos de forma legible.

3.2.- Descripción Detallada de cada bloque.

3.2.1.- Bloque Sensor.

El sensor dispuesto para obtener las mediciones es el Módulo Compás Hitachi HM55B. Este dispositivo tiene como objetivo medir las variaciones magnéticas, reflejadas en dos valores X e Y. Aparte de su alimentación (Vcc, Gnd) necesita tres entradas para su normal funcionamiento: enable, din, clock y presenta una salida: dout.

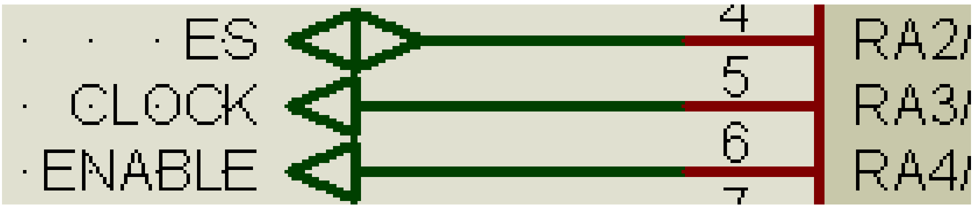


Figura 3.2 Entradas del sensor

En el caso de nuestro proyecto estamos utilizando el sistema 3 wire, que consiste en enviar tres cables para la comunicación entre el microcontrolador y el sensor. Esto solo es posible cuando un pin del microcontrolador (pin cuatro denominado ES), trabaja en algunos momentos como entrada (para recibir datos del pin dout); y otros momentos como salida (enviando datos al pin din). El cambio de entrada a salida del pin ES se logra vía software desde las líneas de código del MikroBasic Pro.

3.2.2.- Bloque base de datos

El proyecto presenta dos tipos de almacenamiento. El primero guarda en la memoria volátil del microcontrolador los valores que han sido tomados.

Se guardarán ocho datos de X, ocho de Y , y ocho del ángulo Θ, lo que equivale a 24 datos tipo palabra (tipo de dato de 16 bits).

Al ser la memoria provisional de tipo volátil esta no se guardará en el Pic una vez que sea apagado el sistema, produciendo la perdida de los datos.

El segundo tipo de almacenamiento (con el fin de evitar perder datos ya tomados) se realiza mediante comunicación serial UART a un circuito Datalogger. Este circuito tiene como objetivo recolectar datos desde sensores por medio de comunicación serial y almacenarlos en una memoria de almacenamiento USB. Igualmente puede realizar el proceso inverso, es decir que puede enviar datos previamente almacenados desde la memoria de almacenamiento USB.

3.2.3.- Bloque presentación de datos

Conocemos varias formas de apreciación del proyecto. Aquí hemos considerado la visualización de los datos tomados en la Lcd, y además se señalaran las gráficas que se presentarán en el proyecto del GLCD.

3.2.3.1.- Presentación por Lcd.

En este proyecto es utilizado para mostrar inmediatamente los valores tomados por el sensor. A medida que el sensor toma un dato en el tiempo ya acordado en alcance del proyecto (5 segundos), este será presentado en el Lcd y permanecerá ahí hasta la siguiente obtención de una muestra.

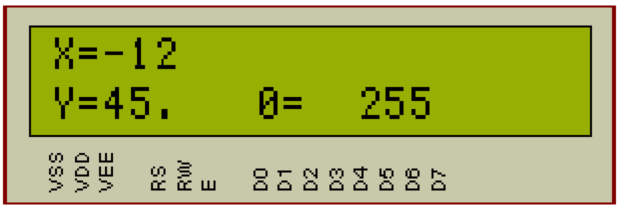


Figura 3.3 Datos ingresados

Además, sirve para la presentación de menús, que junto a dos botones (BOTÓN 1, BOTÓN 2 y BOTÓN 3) conectados al microcontrolador (pines 2, 3 y 7), nos servirán para la elección de algunas opciones dispuestas en el proyecto y analizadas más adelante.

Los menús que se tienen, son los siguientes:

Figura 3.4 Menú 1

Este primer menú permite seleccionar entre tres opciones. Si es que se eligió sensar, se guardarán en un arreglo 8 muestras (cada muestra tiene 3 datos tipo palabra) en intervalos de 5 segundos.

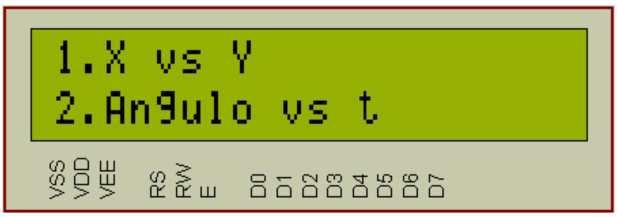


Figura 3.5 Menú 4

Si es que se eligió ingresar al GLCD, se puede escoger entre observar una gráfica XY, o de apreciar una gráfica Angulo vs tiempo. Estos se explicarán mas adelante.

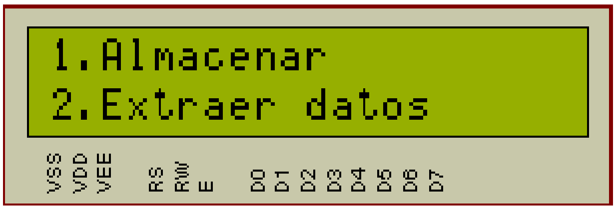


Figura 3.6 Menú 3

Si es que se eligió ingresar al datalogger, se puede escoger entre almacenar los datos en una base de datos (memoria USB) o el de extraer los datos de un archivo previamente ingresado.

3.2.3.2.- Gráficas mostradas en el proyecto del GLCD

En el proyecto del GLCD podremos observar las 8 muestras previamente tomadas en 2 tipos de gráfico.

El primer gráfico denominado gráfico XY, presenta una especie de rosa de los vientos en donde se graficará un punto que refleje la dirección sensada. El ángulo de dirección del punto esta referenciado en sentido horario desde el norte magnético (que se encuentra en la parte superior).

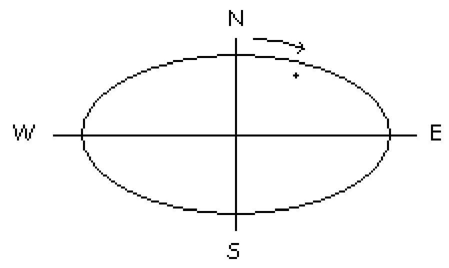


Figura 3.7 Gráfica XY

Apreciaremos puntos que irán apareciendo cada 5 segundos, y que reflejan las 8 muestras que se han tomado previamente.

El segundo gráfico denominado ángulo vs tiempo presentará la variación del ángulo (referenciado al norte magnético) a través del tiempo.

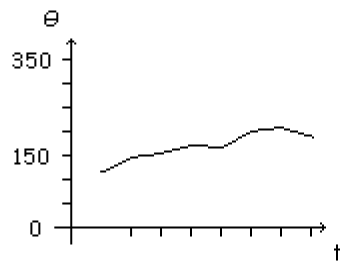


Figura 3.8 Gráfica ángulo vs tiempo

3.2.4.- Bloque de control y procesamiento

Considerado como el bloque modular del proyecto. Está integrado únicamente por el microcontrolador 18F4431, que debido a sus características ya antes mencionadas es ideal para cubrir las demandas que exige este proyecto.

Como principales funciones tiene:

- Controlar los mensajes de menú y muestra de datos que aparecen en la pantalla LCD.
- Control de la comunicación con el módulo compás Hitachi.
- Procesamiento de los datos recolectados por el sensor.
- Almacenamiento temporal de ocho lecturas (24 datos tipo palabras) del sensor.
- Control de la comunicación serial UART con el circuito Datalogger.

3.3.- Programación del PIC 18F4431

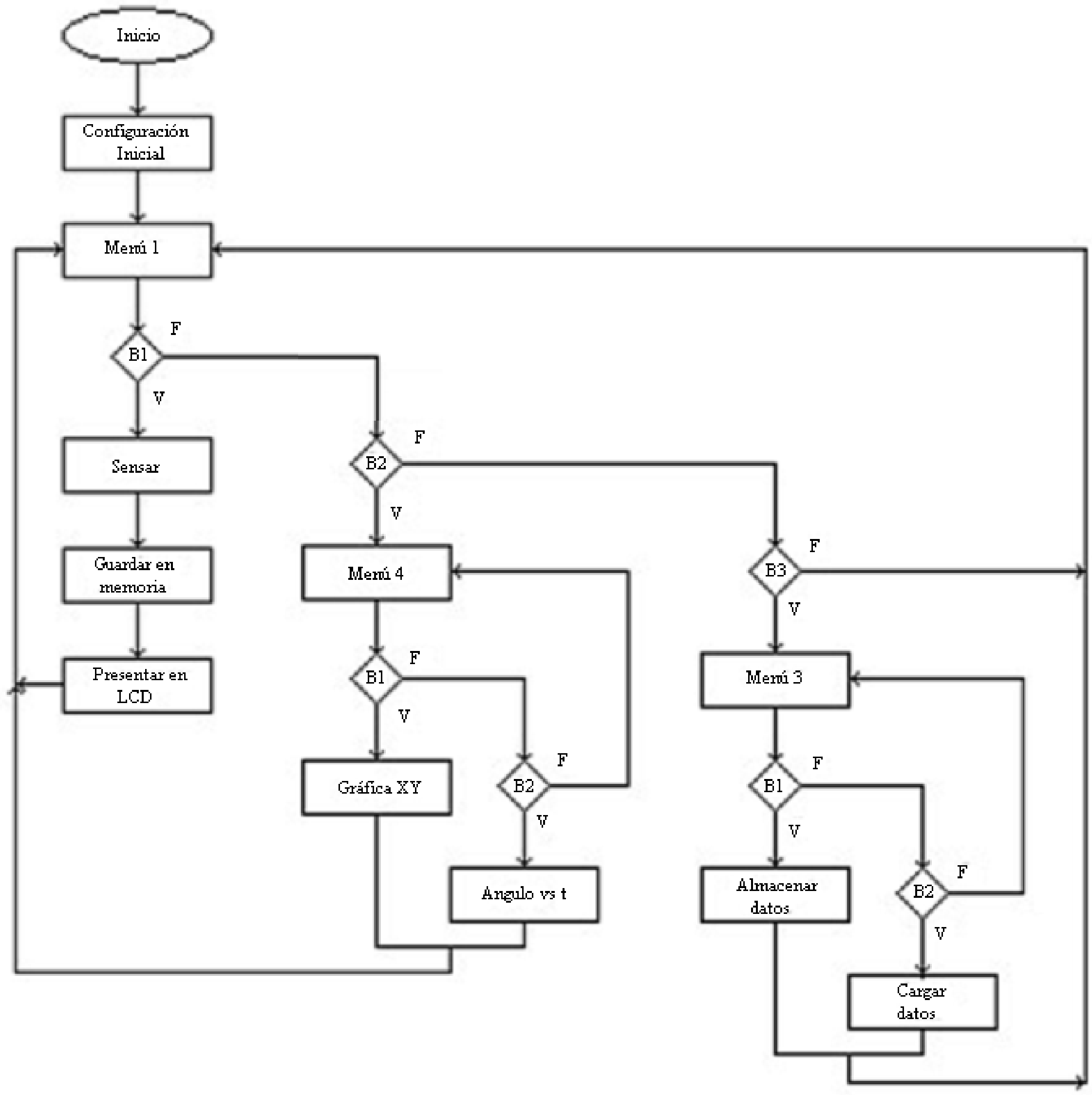


Figura 3.9 Diagrama de flujo del procedimiento principal

Para una mejor comprensión se le ha dado nombres a los pines que permiten al microcontrolador comunicarse con el módulo Hitachi HM55B, como lo son:

E/S – es el pin 4 del microcontrolador que trabaja como salida y luego sirve de entrada, y su función es la de enviar o recibir datos del sensor.

CLOCK – es el pin 5 del microcontrolador que se encarga de enviarle pulsos al sensor.

ENABLE – es el pin 6 del microcontrolador que tiene como propósito habilitar la actividad del sensor.

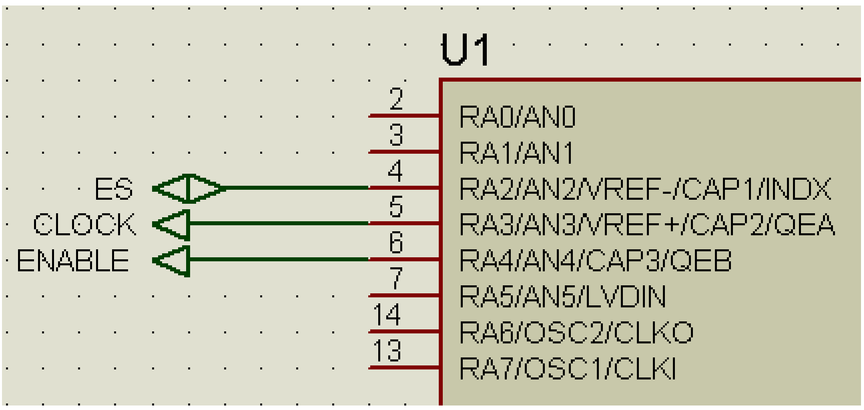


Figura 3.10 Salidas del Pic

Entre las funciones implementadas tenemos a las siguientes:

Habilitar

El procedimiento Habilitar es una función que deja al pin6 ENABLE del microcontrolador en estado bajo habilitando al sensor para recibir o emitir datos.

In

El procedimiento In establece al pin 4 ES del microcontrolador como una entrada y así poder recibir datos del sensor.

Out

El procedimiento Out establece al pin 4 ES del microcontrolador como una salida y así poder enviar datos al sensor.

Pulse

El procedimiento Pulse envía un pulso de reloj por el pin 5 CLOCK del microcontrolador al sensor.

Reset_sensor

Es un procedimiento que tiene como objetivo enviar el comando reset (0000) por el pin 4 ES del microcontrolador al sensor.

Medir

Es un procedimiento que envía el comando medir (1000) por el pin 4 ES del microcontrolador al sensor.

Reporte

Es un procedimiento que envía el comando reporte (1100) al sensor por el pin 4 ES del microcontrolador, con lo que posteriormente el sensor enviara el estado en que se encuentra.

Extractor

Es una función que recibe un dato por el pin 4 ES del microcontrolador enviado por el sensor, mientras se está enviando un pulso por el pin 5 CLOCK del microcontrolador.

Get_axes

Es un procedimiento que utilizando los procedimientos y funciones antes citados, tiene como finalidad hacer la lectura de los valores X y Y del sensor, y además calcula el ángulo.

Mostrar LCD

Función que imprime en el LCD los valores tomados de la medición X, Y, y del ángulo.

Almacenar

Es un proceso que controla la comunicación serial UART para la transmisión de datos (mediciones tomadas por el sensor) desde el presente proyecto hacia el circuito datalogger, logrando así respaldar estos valores.

Cargar

Es un proceso que controla la comunicación serial UART para la recepción de datos (previamente guardados en la base de datos) desde el circuito datalogger hacia el presente proyecto, logrando así restaurar los valores de una sesión anterior.

Gráficos

Es un proceso que envía los datos a presentarse en la Gráfica XY en el proyecto del GLCD. Estos datos han sido procesados antes de ser enviados vía comunicación serial UART.

Gráficos 2

Es un proceso que envía los datos a presentarse en la Ángulo vs tiempo en el proyecto del GLCD. Estos datos han sido procesados antes de ser enviados vía comunicación serial UART.

Menú 1

Es un proceso que presenta por el LCD las opciones a elegirse en el menú 1 (Botón 1 - Sensar, Botón 2 – GLCD y Botón 3 – Datalogger).

Menú 4

Es un proceso que presenta por el LCD las opciones a elegirse en el menú 4 (Botón 1 – Gráfico XY y Botón 2 – Ángulo vs t).

Menú 3

Es un proceso que presenta por el LCD las opciones a elegirse en el menú 3 (Botón 1 – Almacenar datos y Botón 2 – Cargar datos).

Nota:

Para mayor detalle con respecto a la programación, en la sección de ANEXOS B se detalla el código fuente grabado en el microcontrolador 18F4431.

CAPÍTULO 4

En este capítulo se muestra las pruebas y mediciones realizadas, incluyendo la simulación en el software Proteus v7.5, así como las gráficas de los pulsos para la operación del sensor.

4.1 Circuito simulado del proyecto en sus bloques principales.

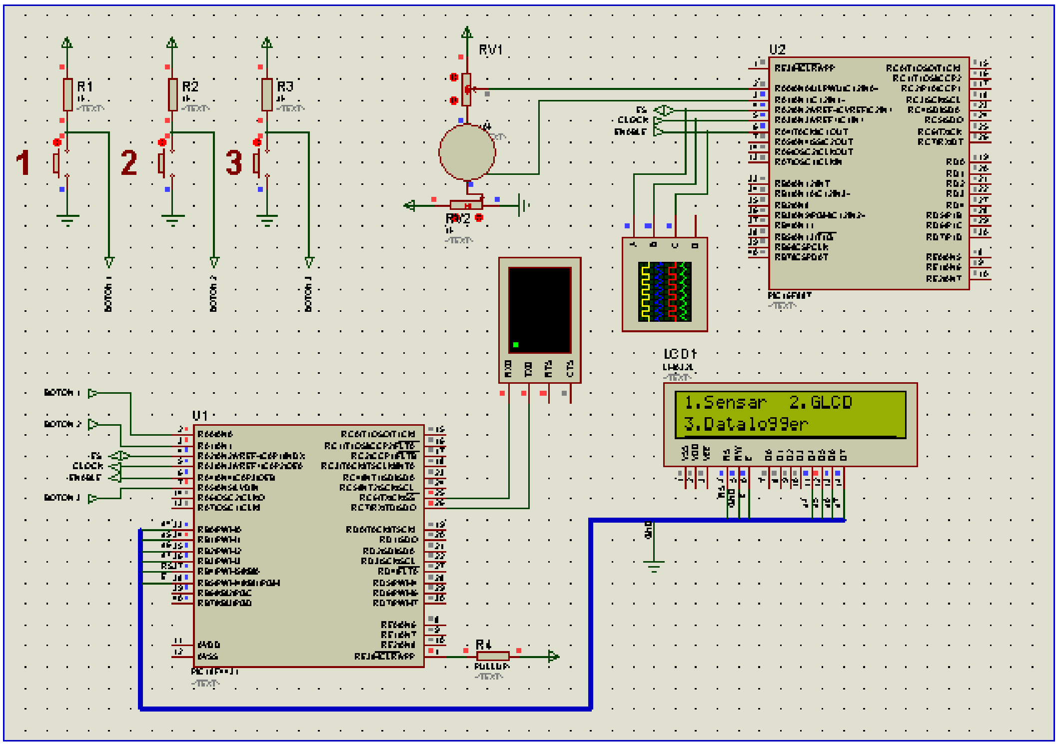


Figura 4.1 Esquemático en proteus

Para la simulación del sensor y debido a que este no consta en el listado de los componentes que tiene el programa Proteus v7.5, se ha tenido que simularlo utilizando otro microcontrolador, el 16F887. El código utilizado para el control de este Pic, también fue realizado en el programa MikroBasic Pro.

El primer microcontrolador 16F887 que hace la función del sensor, simula la obtención de los valores X e Y adquiridos, utilizando una conversión analógica a digital de las variaciones de voltaje de dos potenciómetros, ajustando estos valores por medio de conversiones a las muestras obtenidas por el sensor real y finalmente maneja el envío de datos al otro microcontrolador 18F4431. Cabe resaltar que esta simulación no comprende todas las características del sensor, más bien sirve para entender cómo funciona el intercambio de datos entre el módulo compás y el microcontrolador 18F4431.

4.2.-Programa que simula al módulo compás Hitachi.

El programa tiene como objetivo obtener los valores de X desde los potenciómetros y después de algunas conversiones enviarlos a guardar en las variables X1 e Y1. Después, recibe los comandos enviados desde el microcontrolador 18F4431: reset, measure, status.

Al recibir el comando status este envía al microcontrolador 18F4431 el mensaje de que esta listo a través del comando 1100.

Finalmente, envía uno a uno los bits de la variable X1 e Y1 antes guardados.

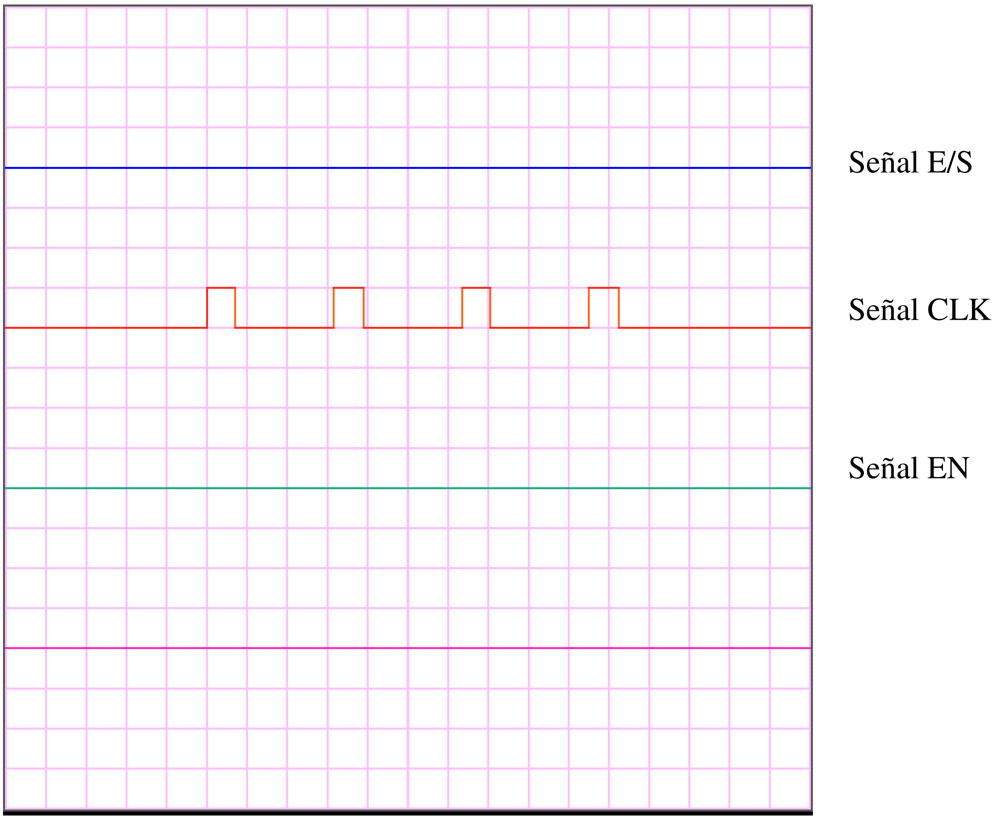


Figura 4.2 Simulación del sensor

4.3.- Gráficas obtenidas en el osciloscopio de la simulación en Proteus

Las siguientes gráficas son los resultados de las mediciones hechas en las entradas al microcontrolador 18F4431, mediante el uso de un osciloscopio.

En el siguiente gráfico, se podrán apreciar 4 líneas que representan las señales del osciloscopio. La señal en color amarillo representa a la señal ES, ésta es la que lleva los datos entre el microcontrolador 18F4431 y el sensor. Le sigue en color azul, la señal CLOCK que muestra los pulsos que se envía del microcontrolador al módulo compás. Finalmente la línea color rojo, presenta el estado del pin EN que es el que se encarga de la habilitación del sensor.

En este gráfico se observa el envío del comando reset 0000 (línea amarilla) hacia el módulo compás.

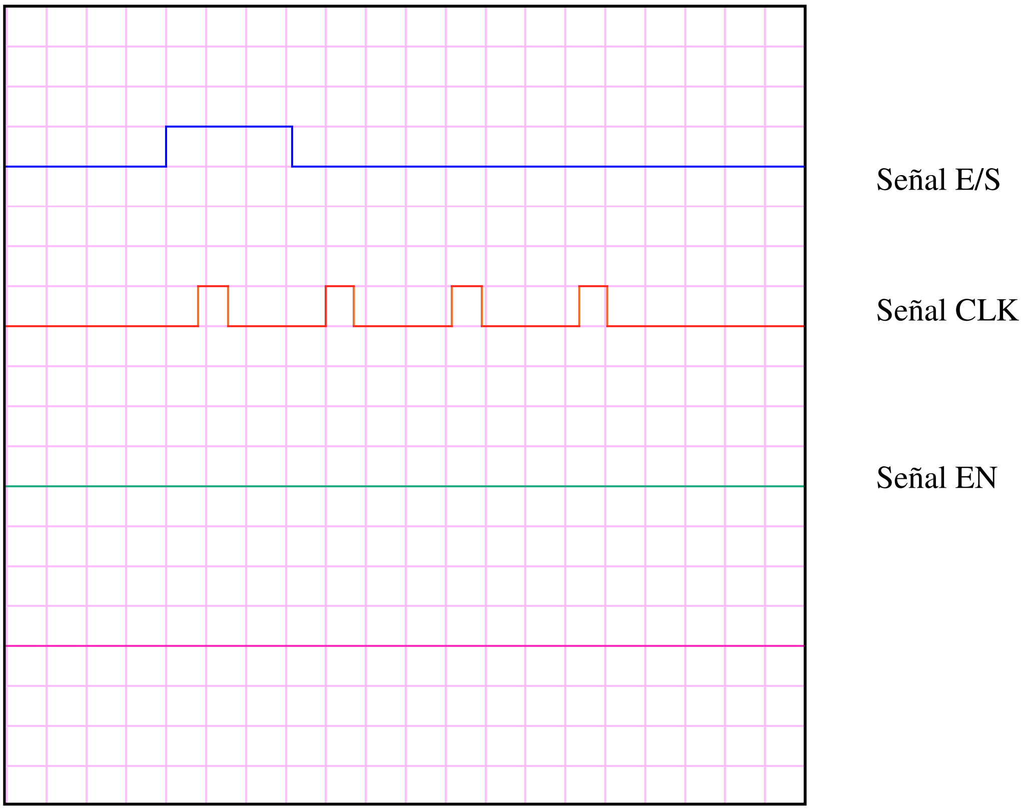


Figura 4.3 Simulación salidas del Pic

En esta segunda imagen se puede ver el comando measure 1000, que le indica al sensor que se prepare para hacer una medición.

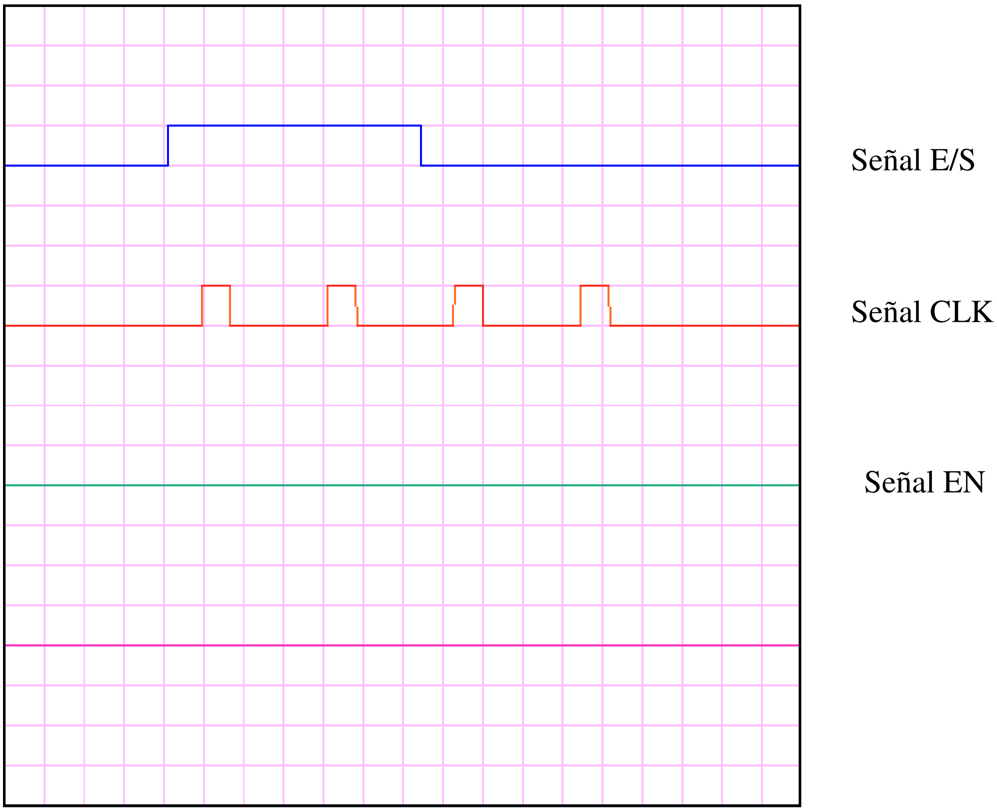


Figura 4.4 Simulación comando measure

Podemos apreciar en este gráfico que se está enviando el dato estado 1100. El sensor responderá con el dato 1100 al microcontrolador cuando no presente errores en su medición.

Finalmente podemos observar el envío de los valores X e Y desde el sensor al microcontrolador. En este caso se está enviando el dato de X =

11111110100 (en decimal -12, porque trabaja con complemento a 2) y Y = 00000110000 (en decimal 48).

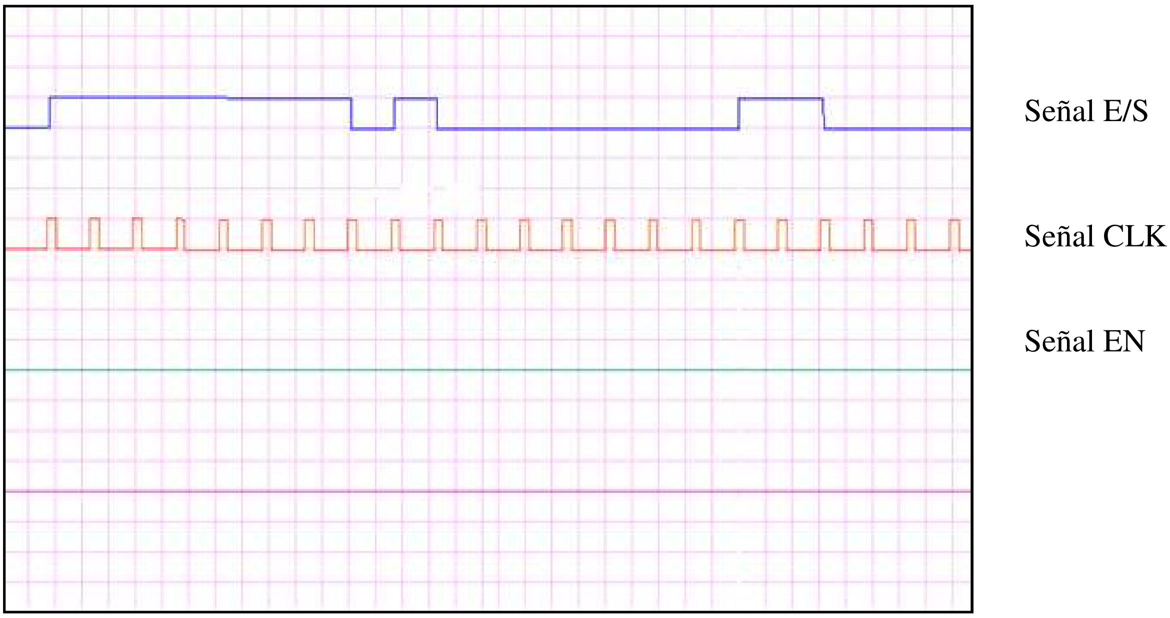


Figura 4.5 Simulación envío datos X e Y

Nota:

Como simulación de nuestro sensor se detalla en ANEXO B la programación con el Pic 16F887.

4.4.- Simulación de la comunicación con el datalogger.

Una prueba para comprobar la comunicación serial entre la brújula digital y el proyecto del datalogger, es simular este envío de datos con la PC.

Para esta simulación se usó un pequeño circuito, el mismo que está basado en el MAX232 que permite la interfaz entre el conector RS232 y circuitos TTL.

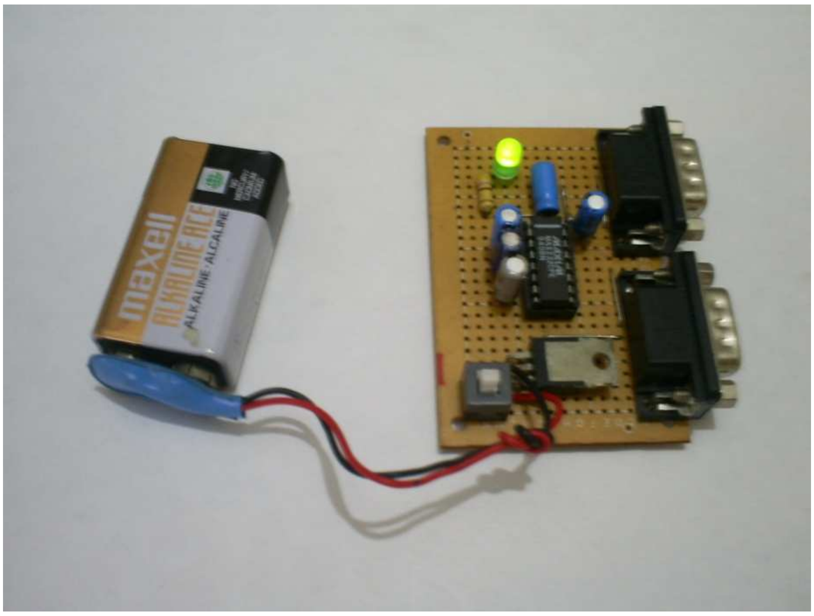


Figura 4.6 Interfaz MAX232

En la figura 4.6 se muestra el circuito utilizado para la interface. Este no consta como parte del presente proyecto.

Además se utilizó la aplicación USART TERMINAL (que se encuentra en el menú de Herramientas) del programa Microbasic Pro, que nos permite enviar caracteres desde la computadora hacia el microcontrolador del prototipo.

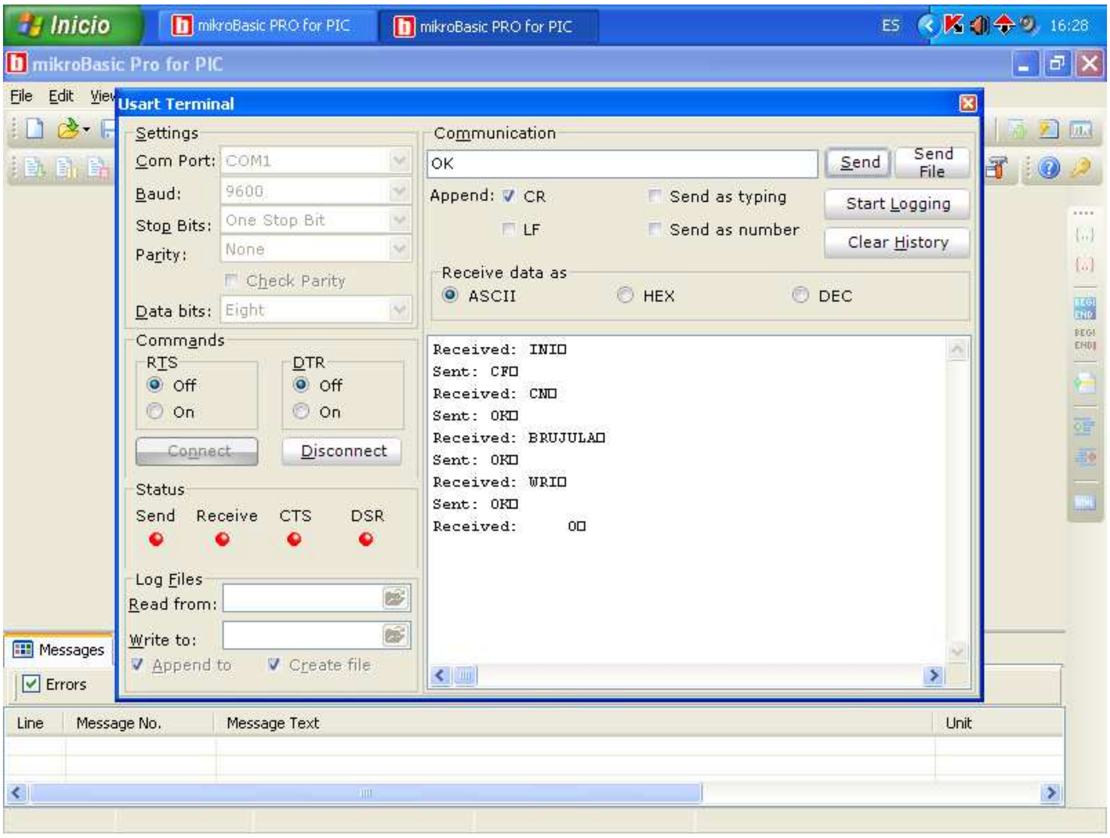


Figura 4.7 Simulación de la comunicación serial

En la figura 4.7 se puede observar el envío de comandos desde la computadora hacia la brújula, simulando la comunicación serial con el datalogger.

De igual manera se realizaron pruebas mediante software para que se conecte con el proyecto de interfaz gráfica (GLCD).

4.5.- Pistas del PCB construidos.

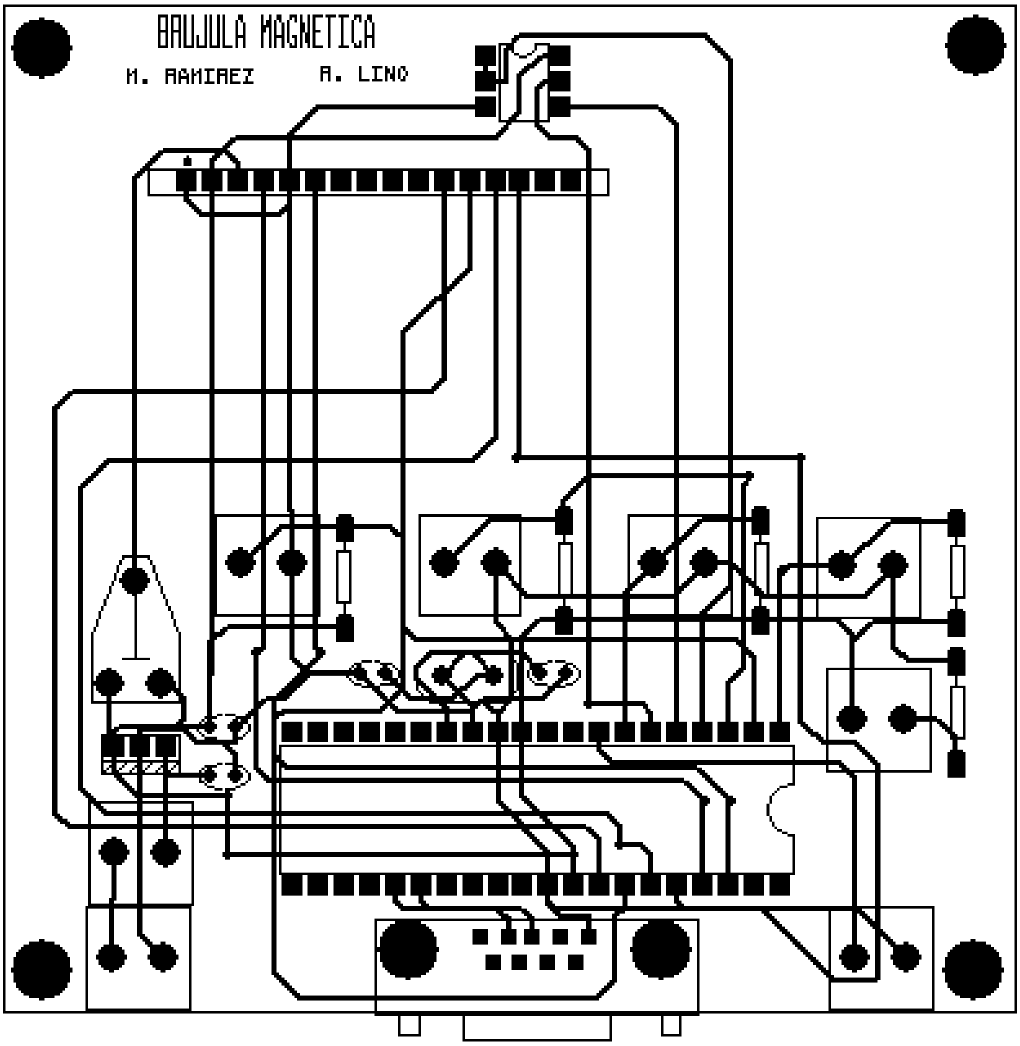


Figura 4.8 Diseño PCB

Figura 4.9 Prototipo

Para el desarrollo del proyecto propuesto se hizo uso de otra aplicación para el diseño o prototipo de la placa donde van a ser soldados los componentes electrónicos que van a participar en la implementación de nuestra brújula digital. Tal aplicación la encontramos en el mismo programa proteus con el nombre de Ares, el mismo que también cuenta con una serie de herramientas para la elaboración completa de un trabajo profesional como el presentado.

El PCB mostrado fue diseñado con la finalidad de que sea portable, de ahí su tamaño pequeño de 10 cm^2. Está hecha a doble cara a fin de no presentar demasiados puentes entre las diferentes pistas en lo referente a la parte superior para su presentación.

4.6.- Resultados experimentales.

Para el análisis de los resultados experimentales, se hizo mediciones entre la brújula propuesta en este proyecto y una brújula analógica. De las distintas pruebas que se tomaron entre las dos, se encuentra que la brújula digital tiene un error de 4° en comparación a la brújula analógica. Es decir, que los valores registrados por la brújula digital en una misma dirección variarán ligeramente, marcando a veces valores más bajos y otras veces más altos. Esto se debe principalmente a diferentes tipos de interferencia que colectivamente son llamadas ruido.

Figura 4.10 Comparación entre brújulas

En la figura 4.10 se muestra una comparación entre la orientación de una brújula común y la brújula digital.

Dada la gran sensibilidad del sensor HM55B, y recordando que cuando una corriente pasa a través de un conductor genera un campo magnético alrededor de él; se tuvo la necesidad de ubicar este sensor en una parte en donde las corrientes propias del circuito no interfieran demasiado en las lecturas tomadas.

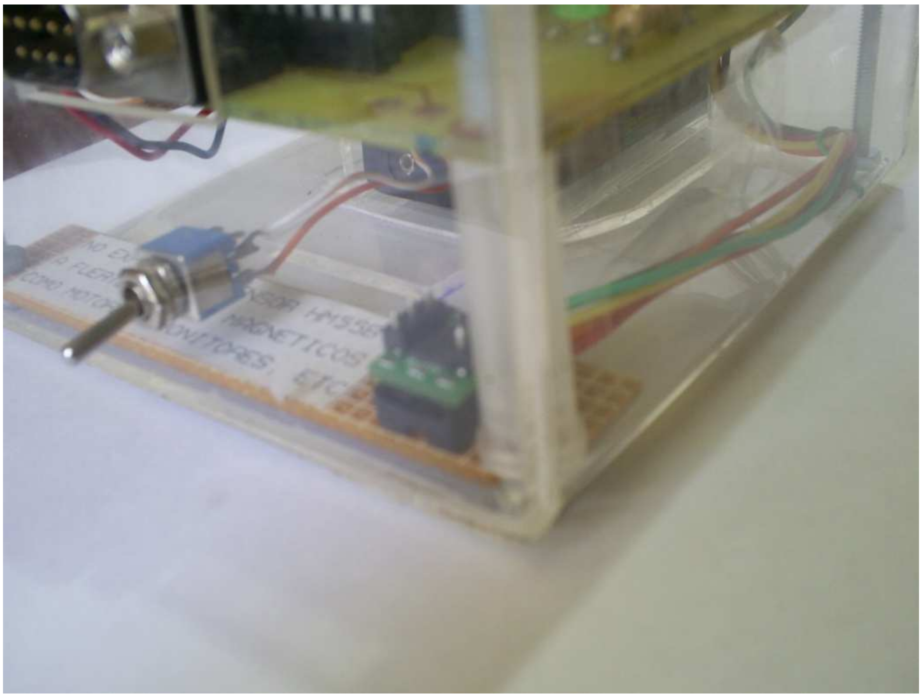


Figura 4.11 Ubicación del sensor

En la figura 4.11 se muestra la ubicación del sensor dentro de la maqueta, se puede apreciar un aislamiento del resto de la circuitería, así como de los contactos metálicos.

CONCLUSIONES

1) Los valores X e Y obtenidos en las mediciones tomadas en el laboratorio fueron mucho mayores en comparación a las muestras obtenidas en un ambiente en donde la intensidad del campo magnético es mucho menor, lo que confirma que en un espacio cerrado con fuentes de interferencia magnética se tendrá un mayor rango de error.
2) Para la simulación del proyecto en Proteus tuvo que implementarse un sensor a base de potenciómetros y de otro microcontrolador que convierta las señales analógicas en digitales, porque el módulo compás no se encuentra listado en las librerías del simulador Proteus, y era necesario comprobar el funcionamiento correcto de la comunicación serial síncrona entre el microcontrolador y el sensor.
3) De la programación utilizada en MikroBasic Pro, tuvimos problemas con funciones que no trabajan de la forma especificada en el manual, posiblemente a causa de ruidos; por lo que tuvimos que implementar nuevas funciones para lograr que el microcontrolador trabaje de la forma deseada.

4) El dispositivo de medición estudiado en este proyecto puede ser utilizado en el campo de la robótica o cualquier otra aplicación en donde sea necesaria la visualización o control de la orientación de algún aparato, teniendo la ventaja de una interfaz fácil de manejar e integrar mediante un microcontrolador.

5) Además de todas las ventajas que brinda el proyecto, se ha propuesto que al mismo se le pueda incorporar un pequeño parlante con la finalidad de que a medida que se acerca al norte magnético, un sonido puede aumentar o disminuir su frecuencia para mostrarnos la proximidad o alejamiento del norte.

RECOMENDACIONES

1) Para desarrollar aplicaciones con el sensor empleado, es necesario conocer como se solicitan y en que formato se reciben los datos de la medición que realiza el sensor, para evitar esfuerzos adicionales por parte del programador.
2) Para asegurar el buen funcionamiento de la brújula, es conveniente hacer pruebas y ajustes. En caso de que se compruebe un gran porcentaje de error comparado con una brújula mecánica, se tendrá que proceder a la calibración por software en caso de ser necesario.
3) Evitar el manejo del sensor cerca de campos magnéticos fuertes, por ejemplo, en sitios donde hayan motores eléctricos y monitores. La exposición a altos campos magnéticos puede dañar al sensor permanentemente.

ANEXO A

MANUAL DE USUARIO

1.- INTRODUCCIÓN

La operación y puesta en marcha del proyecto serán descritos en los siguientes literales a fin de no causar algún daño a los semiconductores y que el sistema pueda funcionar de manera correcta.

2.- ANTES DEL ENCENDIDO DEL SISTEMA

2.1.- Verificar la alimentación requerida por los datasheet de los circuitos involucrados, como son Vcc y Gnd. Para este informe se tomó como base 5Vdc para Vcc y 0v para Gnd.

2.2.- Proteger o alejar al sensor de fuertes emanaciones o fuentes de campo magnético a fin de no obtener lecturas erróneas.

3.- FUNCIONAMIENTO Y PUESTA EN MARCHA

3.1.- El circuito inicia a operar una vez que ha sido energizado. Se procede a la elección de una de las opciones del Menú 1. Se podrá escoger entre:

a) Sensar (Botón 1)
b) Acceder a una gráfica GLCD (Botón 2)
c) Acceder al Datalogger (Botón 3)

3.2.- Si se seleccionó Sensar, se procederá a tomar 8 muestras, y posteriormente se regresará al Menú 1.

3.3.- Si se seleccionó Acceder a una gráfica GLCD se alcanzará el Menú 2. En este menú se deberá escoger entre 2 tipos de gráficas de apreciación:

a) Gráfica X vs Y (Botón 1).- Donde se observará una serie de puntos que señala las posiciones que ha sensado la brújula.
b) Ángulo vs t (Botón 2).- Se mostrará el cambio del ángulo (con respecto al norte magnético) en el transcurso del tiempo.

Después de haber seleccionado una gráfica y haberla apreciado se regresará al Menú 1.

3.4.- Si se seleccionó Acceder al Datalogger, tendremos la opción de escoger:

a) Almacenar datos (Botón 1).- Donde se podrá guardar los datos sensados (de haberlos) en una memoria USB.
b) Extraer datos (Botón 2).- Donde se podrá extraer los datos guardados con anticipación.

Finalmente, se regresará al Menú 1.

ANEXO B

CÓDIGO PRINCIPAL PARA 18F4431

```
program a1

'Variables de control para el módulo LCD
' Lcd module connections
dim LCD_RS as sbit at RB4_bit
    LCD_EN as sbit at RB5_bit
    LCD_D4 as sbit at RB0_bit
    LCD_D5 as sbit at RB1_bit
    LCD_D6 as sbit at RB2_bit
    LCD_D7 as sbit at RB3_bit

    LCD_RS_Direction as sbit at TRISB4_bit
    LCD_EN_Direction as sbit at TRISB5_bit
    LCD_D4_Direction as sbit at TRISB0_bit
    LCD_D5_Direction as sbit at TRISB1_bit
```

```
    LCD_D6_Direction as sbit at TRISB2_bit
    LCD_D7_Direction as sbit at TRISB3_bit
' End Lcd module connections

'Configuración de los pines de entrada y salida
dim BOTON1 as sbit at RA0_bit
    BOTON2 as sbit at RA1_bit
    BOTON3 as sbit at RA5_bit
    ES as sbit at RA2_bit
    RELOJ as sbit at RA3_bit
    EN as sbit at RA4_bit

    BOTON1_DIR as sbit at TRISA0_bit
    BOTON2_DIR as sbit at TRISA1_bit
    BOTON3_DIR as sbit at TRISA5_bit
    ES_DIR as sbit at TRISA2_bit
    RELOJ_DIR as sbit at TRISA3_bit
    EN_DIR as sbit at TRISA4_bit

'Variables a usar en el programa
dim estado as byte[4]
dim x, y as byte[11]
```

```
dim i, j, salir_menu2, uart_rd as byte

dim x1, y1 as integer

dim angle, como as float '-->quitar angulo

dim angle2 as word

dim txt, txt2, enter as string[20]

const n2 as word[11] = (1,2,4,8,16,32,64,128,256,512,1024)

const negmask as word = $f800

dim memoria as word[24] 'guarda 8 muestras



sub procedure habilitar()

   EN = 1

   delay_us(400)

   EN = 0

end sub


sub procedure in()

     ES_DIR = 1

end sub


sub procedure out()

     ES_DIR = 0
```

```
end sub

sub procedure pulse()
  delay_us(15)
  RELOJ = 1
  delay_us(14)
  RELOJ = 0
  delay_us(31)
end sub

'procedimiento que resetea el sensor
sub procedure reset_sensor()
   out ()
   ES = 0
   pulse ()
   ES = 0
   pulse ()
   ES = 0
   pulse ()
   ES = 0
   pulse ()
end sub
```

```
'mando el comando para iniciar a medir
sub procedure medir()
    out ()
    ES = 1
    pulse ()
    ES = 0
    pulse ()
    ES = 0
    pulse ()
    ES = 0
    pulse ()
end sub

'solicito el reporte del sensor
sub procedure reporte()
      out ()
      ES = 1
      pulse ()
      ES = 1
      pulse ()
      ES = 0
```

```
      pulse ()

      ES = 0

      pulse ()

end sub


'obtengo un dato del sensor - trabajo con postmode

sub function extractor() as byte

   in ()

   delay_us(15)

   RELOJ = 1


   delay_us(14)

   result = ES

   RELOJ = 0


   delay_us(31)

end sub


sub procedure get_axes()

    habilitar()


    DELAY_US(800)
```

```
'envio el comando para reset
reset_sensor()
DELAY_US(400)

habilitar()
DELAY_US(800)

medir()
DELAY_US(800)

do
  habilitar()
  DELAY_US(800)

  reporte()
  DELAY_US(800)

  'obtengo estado del sensor
  estado[0] = extractor()
  estado[1] = extractor()
  estado[2] = extractor()
```

```
      estado[3] = extractor()

      DELAY_US(1200)
    loop until ((estado[0] = 1) and (estado[1] = 1) and (estado[2] = 0) and
(estado[3] = 0))

    'Extraigo valor X y Y del sensor
    for j = 0 to 10
        x[10-j] = extractor()
    next j
    for j = 0 to 10
        y[10-j] = extractor()
    next j

    EN = 1 'high en

    'paso las variables de binario a decimal
    x1 = 0
    for j = 0 to 10
       x1 = x1 + (n2[j]*x[j])
    next j
```

```
y1 = 0

for j = 0 to 10

    y1 = y1 + (n2[j]*y[j])

next j


'para el caso de cantidades negativas

if x[10] = 1 then

   x1 = x1 or negmask

end if


if y[10] = 1 then

   y1 = y1 or negmask

end if


'obtengo el angulo

if x1 = 0 then

   if y[10] = 1 then

      angle = 90

   else

       angle = -90

   end if

else
```

```
        angle = atan(-y1/x1)

        angle = (angle*180)/3.1416

    end if


    if (x[10] = 1) and (y[10] = 0) then

       angle = angle + 180

    end if


    if (x[10] = 0) and (y[10] = 0) then

       angle = angle + 360

    end if


    if (x[10] = 1) and (y[10] = 1) then

       angle = angle + 180

    end if

end sub


'procedimiento de lcd

'presentacion de los datos a traves del lcd (temporal)

sub procedure mostrar_lcd()

    Lcd_Cmd(_LCD_RETURN_HOME)
```

```
IntToStr(x1, txt)
Lcd_Out(1,1,txt)

txt = "X="
Lcd_Out(1,1,txt)

txt = "          "
Lcd_Out(1,7,txt)

IntToStr(y1, txt)
Lcd_Out(2,1,txt)
txt = "Y="
Lcd_Out(2,1,txt)

txt = "   0="
Lcd_Out(2,7,txt)
angle2 = angle
WordToStr(angle2, txt)

'Palabra norte
if angle2 = 360 then
   txt = "NORTE"
```

```
   Lcd_Out(2,10,txt)
   txt2 = "       "
   Lcd_Out(2,10,txt2)

else
   Lcd_Out(2,12,txt)
end if



'Palabra sur
if angle2 = 180 then
   txt = "  SUR  "
   Lcd_Out(2,10,txt)
else
   Lcd_Out(2,12,txt)
end if

'Palabra este
if angle2 = 90 then
   txt = "ESTE   "
   Lcd_Out(2,10,txt)
```

```
      txt2 = "        "
      Lcd_Out(2,10,txt2)

   else
      Lcd_Out(2,12,txt)
   end if

    'Palabra oeste
   if angle2 = 270 then
      txt = "  OESTE"
      Lcd_Out(2,10,txt)
   else
      Lcd_Out(2,12,txt)
   end if

   delay_ms(150)
end sub

sub procedure menu1()
   Lcd_Cmd(_LCD_RETURN_HOME)
   txt = "1.Sensar  2.GLCD"
```

```
    Lcd_Out(1,1,txt)
    txt = "3.Datalogger     "
    Lcd_Out(2,1,txt)
end sub

sub procedure menu3()
    Lcd_Cmd(_LCD_RETURN_HOME)
    txt = "1.Almacenar     "
    Lcd_Out(1,1,txt)
    txt = "2.Extraer datos "
    Lcd_Out(2,1,txt)
end sub
sub procedure guardando()
    Lcd_Cmd(_LCD_RETURN_HOME)
    txt = "                "
    Lcd_Out(1,1,txt)
    txt = "Guardando....   "
    Lcd_Out(2,1,txt)
end sub

sub procedure cargando()
    Lcd_Cmd(_LCD_RETURN_HOME)
```

```
    txt = "                "
    Lcd_Out(1,1,txt)
    txt = "Cargando....    "
    Lcd_Out(2,1,txt)
end sub

sub procedure graficando()
    Lcd_Cmd(_LCD_RETURN_HOME)
    txt = "                "
    Lcd_Out(1,1,txt)
    txt = "Graficando....  "
    Lcd_Out(2,1,txt)
end sub

sub procedure menu4()
    Lcd_Cmd(_LCD_RETURN_HOME)
    txt = "1.X vs Y        "
    Lcd_Out(1,1,txt)
    txt = "2.Angulo vs t   "
    Lcd_Out(2,1,txt)
end sub
```

```
sub procedure almacenar()

    UART1_Write_Text("INI")            ' sends back text
    UART1_Write(13)
    UART1_Read_Text(txt, "CF", 10)
    UART1_Read_Text(txt, enter, 10)

    UART1_Write_Text("CN")             ' sends back text
    UART1_Write(13)
    UART1_Read_Text(txt, "OK", 10)
    UART1_Read_Text(txt, enter, 10)



    UART1_Write_Text("BRUJULA")
    UART1_Write(13)
    UART1_Read_Text(txt, "OK", 10)
    UART1_Read_Text(txt, enter, 10)

    UART1_Write_Text("WRI")
    UART1_Write(13)
    UART1_Read_Text(txt, "OK", 10)
    UART1_Read_Text(txt, enter, 10)
```

```
    delay_ms(100)

    for j = 0 to 23
        WordToStr(memoria[j], txt)
        UART1_Write_Text(txt)
        UART1_Write(13)
        UART1_Read_Text(txt, "OK", 10)
        UART1_Read_Text(txt, enter, 10)
        next j

    UART1_Write_Text("END")
    UART1_Write(13)
    UART1_Read_Text(txt, "EF", 10)
    UART1_Read_Text(txt, enter, 10)

    salir_menu2 = 1
end sub

sub procedure cargar()
    'enviar datos
```

```
UART1_Write_Text("INI")                ' sends back text

UART1_Write(13)

UART1_Read_Text(txt, "CF", 10)

UART1_Read_Text(txt, enter, 10)


UART1_Write_Text("CN")                 ' sends back text

UART1_Write(13)

UART1_Read_Text(txt, "OK", 10)

UART1_Read_Text(txt, enter, 10)


UART1_Write_Text("BRUJULA")

UART1_Write(13)

UART1_Read_Text(txt, "OK", 10)

UART1_Read_Text(txt, enter, 10)


UART1_Write_Text("RD")

UART1_Write(13)


j = 0


do

   UART1_Read_Text(txt, enter, 10)
```

```
        if strcmp(txt,"EF") = 0 then
          salir_menu2 = 1
        else
          memoria[j] = StrToword(txt)
          UART1_Write_Text("R")
          UART1_Write(13)
        end if
        j = j + 1
    loop until (salir_menu2 = 1)
end sub


'grafica comandos por el glcd
sub procedure graficos()
    j = 0
    do
      do
        if (UART1_Data_Ready() = 1) then
          uart_rd = UART1_Read()
        end if
      loop until (uart_rd = 0x15)
```

```
        UART1_Write(0x20)
        delay_ms (100)

        como = memoria[(3*j)+2] - 90
        angle = ((3.1416*como)/180)

        como = 25*(cos(angle))
        X1 = como
        UART1_Write(X1)
        delay_ms (100)

        como = 25*(sin(angle))
        Y1 = como
        UART1_Write(Y1)

        delay_ms(5000)

        j=j+1
    loop until j=8

    salir_menu2 = 1
end sub
```

```
'grafica comandos por el glcd
sub procedure graficos2()
    j = 0
    do
      do
        if (UART1_Data_Ready() = 1) then
           uart_rd = UART1_Read()
        end if
      loop until (uart_rd = 0x30)


      UART1_Write(0x40)
      delay_ms (100)


      como = memoria[(3*j)+2]* 255
      como = como / 360


      X1 = como
      UART1_Write(X1)


      delay_ms(5000)
```

```
      j=j+1
    loop until j=8

    salir_menu2 = 1
end sub

'***ARRANCA EL PROGRAMA PRINCIPAL***
main:
'Configuro estado de los pines y los registros
ANSEL0 = 0
ANSEL1 = 0
OSCCON = 2
RELOJ_DIR = 0
EN_DIR = 0
BOTON1_DIR = 1
BOTON2_DIR = 1
BOTON3_DIR = 1

'inicializo variables
RELOJ = 0
EN = 0
ES = 0
```

```
trisb = 0 'configura portb como salida

portb = 0 'el lcd arranca sin ningun mensaje

enter[0] = 13

enter[1] = 0


'inicializo lcd

Lcd_Init()                    ' Initialize Lcd

Lcd_Cmd(_LCD_CLEAR)           ' Clear display

Lcd_Cmd(_LCD_CURSOR_OFF)


'inicializo uart

UART1_Init(9600)              ' Initialize UART module at 9600 bps

Delay_ms(100)

while(true)

  salir_menu2 = 0

  menu1()


  if button (PORTA, 0, 1, 0) then

    for i = 0 to 7

       get_axes()

       mostrar_lcd()
```

```
      memoria[3*i] = X1
      memoria[(3*i)+1] = Y1
      memoria[(3*i)+2] = angle
      delay_ms(5000) 'tiempo EN que sensa los datos
   next i
else
    if button (PORTA, 1, 1, 0) then 'aqui viene la parte de graficos
      delay_ms (500)
      do
      menu4()
      if button (PORTA, 0, 1, 0) then 'aca viene la funcion guardar
         graficando()
         graficos ()
         delay_ms(500)
      else
          if button (PORTA, 1, 1, 0) then 'aca viene la funcion cargar
             graficando()
             graficos2 ()
             delay_ms(500)
          end if
      end if
       loop until (salir_menu2 = 1)
```

```
        else
            if button (PORTA, 5, 1, 0) then
               delay_ms (500)
               do
               menu3()
               if button (PORTA, 0, 1, 0) then 'aca viene la funcion guardar
                  guardando()
                  almacenar()
                  delay_ms(500)
               else
                   if button (PORTA, 1, 1, 0) then 'aca viene la funcion cargar
                      cargando()
                      cargar()
                      delay_ms(500)
                   end if
               end if
               loop until (salir_menu2 = 1)
            end if
        end if
    end if
wend
end.
```

CÓDIGO FUENTE SENSOR

program sensor

```
' Declaraciones
dim POT1 as sbit at RA0_bit        'potenciometro 1 a entrada análogica
    POT2 as sbit at RA1_bit        'potenciometro 2 a entrada análogica
    DINDOUT as sbit at RA2_bit      'pin que funciona como entrada y salida de
datos
    RELOJ as sbit at RA3_bit       'entrada de reloj proporcionado por el pic
18F4431
    EN as sbit at RA4_bit          'entrada habilitadora por el pic 18F4431

    POT1_DIR as sbit at TRISA0_bit
    POT2_DIR as sbit at TRISA1_bit
    DINDOUT_DIR as sbit at TRISA2_bit
    RELOJ_DIR as sbit at TRISA3_bit
    EN_DIR as sbit at TRISA4_bit
```

```
dim X as float              'variable que almacena los datos obtenidos de los
pots
dim X1, Y1 as integer        'datos X e Y a salir
dim entrada as byte[4]       'registra el comando status ingresado por el
micro
dim i as byte               'contador auxiliar

main:
   ANSEL = 3                'indica que PORTA.0 Y PORTA.1 son entradas
analógicas
   ANSELH = 0
   POT1_DIR = 1             'registra a POT1 como entrada
   POT2_DIR = 1             'registra a POT2 como entrada
   DINDOUT_DIR = 1          'registra a DINDOUT como entrada
   RELOJ_DIR = 1            'registra a RELOJ como entrada
   EN_DIR = 1               'registra a EN como entrada

   while (true)
   DINDOUT_DIR = 1          'registra a DINDOUT como entrada

   'Toma de datos desde los pots
   X = ADC_READ (0)
```

```
X = X/1023                  'conversiones que pasan el rango de los valores
X = X*100                   'obtenidos en los potenciometros a el rango de
X = X - 50                  'los obtenidos en las pruebas
X1 = X
X = ADC_READ (1)            'se procede igualmente con el valor de y
X = X/1023
X = X*100
X = X - 50
Y1 = X

'Esta parte me captura 4 entradas, comparandolas con el comando status
'1100 enviado por el microcontrolador 18F4431
do
  do
  loop until RELOJ = 1      'este lazo espera por el pulso de subida
    entrada[3] = DINDOUT   'cuando se recibe este pulso se captura el
valor
  do
  loop until RELOJ = 0      'este lazo espera a que el pulso caiga

  do
  loop until RELOJ = 1
```

```
      entrada[2] = DINDOUT
    do
    loop until RELOJ = 0

    do
    loop until RELOJ = 1
      entrada[1] = DINDOUT
    do
    loop until RELOJ = 0

    do
    loop until RELOJ = 1
      entrada[0] = DINDOUT
    do
    loop until RELOJ = 0

  loop until ((entrada[3] = 1) and (entrada[2] = 1) and (entrada[1] = 0) and
(entrada[0] = 0))

  'ahora que se ha recibido el comando de status 1100
  'se proporcionara la salida 1100 correspondiente a el
  'mensaje del sensor al pic de que esta listo
```

```
DINDOUT_DIR = 0

  do
  loop until RELOJ = 1
    dindout = 1
  do
  loop until RELOJ = 0

  do
  loop until RELOJ = 1
    dindout = 1
  do
  loop until RELOJ = 0

  do
  loop until RELOJ = 1
    DINDOUT = 0
  do
  loop until RELOJ = 0

  do
  loop until RELOJ = 1
```

```
    DINDOUT = 0
  do
  loop until RELOJ = 0

'ya enviado este comando 1100 se procede a enviar los valores X e Y
'comenzando por el pin msb de los 11 bits del dato X para despues
'seguir con el valor de Y

  do
  loop until RELOJ = 1
    DINDOUT = TestBit(x1, 10)        'X10
  do
  loop until RELOJ = 0

  do
  loop until RELOJ = 1
    DINDOUT = TestBit(x1, 9)         'X9
  do
  loop until RELOJ = 0

  do
  loop until RELOJ = 1
```

```
  DINDOUT = TestBit(x1, 8)         'X8
do
loop until RELOJ = 0
do
loop until RELOJ = 1
  DINDOUT = TestBit(X1, 7)         'X7
do
loop until RELOJ = 0
do
loop until RELOJ = 1
  DINDOUT = TestBit(X1, 6)         'X6
do
loop until RELOJ = 0
do
loop until RELOJ = 1
  DINDOUT = TestBit(X1, 5)         'X5
do
loop until RELOJ = 0
do
loop until RELOJ = 1
  DINDOUT = TestBit(X1, 4)         'X4
do
```

```
loop until RELOJ = 0
do
loop until RELOJ = 1
   DINDOUT = TestBit(X1, 3)         'X3
do
loop until RELOJ = 0
do
loop until RELOJ = 1
   DINDOUT = TestBit(X1, 2)         'X2
do
loop until RELOJ = 0
do
loop until RELOJ = 1
   DINDOUT = TestBit(X1, 1)         'X1
do
loop until RELOJ = 0
do
loop until RELOJ = 1
   DINDOUT = TestBit(X1, 0)         'X0
do
loop until RELOJ = 0
```

```
'aca arranca y
do
loop until RELOJ = 1
   DINDOUT = TestBit(y1, 10)        'Y10
do
loop until RELOJ = 0
do
loop until RELOJ = 1
   DINDOUT = TestBit(y1, 9)         'Y9
do
loop until RELOJ = 0
do
loop until RELOJ = 1
   DINDOUT = TestBit(y1, 8)         'Y8
do
loop until RELOJ = 0
do
loop until RELOJ = 1
   DINDOUT = TestBit(y1, 7)         'Y7
do
loop until RELOJ = 0
do
```

```
loop until RELOJ = 1
   DINDOUT = TestBit(y1, 6)         'Y6
do
loop until RELOJ = 0
do
loop until RELOJ = 1
   DINDOUT = TestBit(y1, 5)         'Y5
do
loop until RELOJ = 0
do
loop until RELOJ = 1
   DINDOUT = TestBit(y1, 4)         'Y4
do
loop until RELOJ = 0
do
loop until RELOJ = 1
   DINDOUT = TestBit(y1, 3)         'Y3
do
loop until RELOJ = 0
do
loop until RELOJ = 1
   DINDOUT = TestBit(y1, 2)         'Y2
```

```
        do
        loop until RELOJ = 0
        do
        loop until RELOJ = 1
          DINDOUT = TestBit(y1, 1)        'Y1
        do
        loop until RELOJ = 0
        do
        loop until RELOJ = 1
          DINDOUT = TestBit(y1, 0)        'Y0
        do
        loop until RELOJ = 0

      DINDOUT = 0
      wend

   end.
```

BIBLIOGRAFÍA

1. Filecrop, Mikrobasic v3.2 download, http://www.filecrop.com/search.php?w=mikrobasic+v3.2&size_i=0&size_f=1048576&engine_r=1&engine_m=1, **Fecha de consulta:** 29 de marzo del 2010.
2. Filecrop, Proteus v7.5 download, http://www.filecrop.com/search.php?w=proteus+v7.5&size_i=0&size_f=1048576&engine_r=1&engine_m=1, **Fecha de consulta:** 29 de marzo del 2010.
3. Parallax, Página principal Parallax, http://www.parallax.com/, **Fecha de consulta:** 29 de marzo del 2010.
4. Mikroelektrónica , Compiladores, ejemplos, libros, http://www.mikroe.com/, **Fecha de consulta:** 29 de marzo del 2010.
5. Microchip , Página principal Microchip, http://www.microchip.com/, **Fecha de consulta:** 2 de abril del 2010.
6. Mikroelectrónica, Programming PIC Microcontrollers in BASIC , http://www.mikroe.com/en/books/picbasicbook/00.htm, **Fecha de consulta:** 2 de abril del 2010.
7. Parallax, Smart Sensors and Applications, http://www.parallax.com/dl/docs/prod/sic/smartsensors-v1.0.pdf, **Fecha de consulta:** 3 de abril del 2010.
8. Parallax, ¿Qué es un Microcontrolador?, http://www.parallax.com/dl/docs/books/edu/WAMv1_1Spanish.pdf, **Fecha de consulta:** 4 de abril del 2010.
9. Microchip , Datasheet PIC 18F4431, http://ww1.microchip.com/downloads/en/DeviceDoc/39616C.pdf, **Fecha de consulta:** 5 de abril del 2010.
10. Parallax , HM55B Basic Stamp, http://www.parallax.com/Portals/0/Downloads/docs/prod/compshop/HitachiPckIns.pdf, **Fecha de consulta:** 5 de abril del 2010.
11. Parallax , Módulo Hitachi características, http://www.parallax.com/Store/Sensors/CompassGPS/tabid/173/CategoryID/48/List/0/SortField/0/Level/a/ProductID/98/Default.aspx, **Fecha de consulta:** 6 de abril del 2010.
12. Parallax, Device Information HM55B, http://www.parallax.com/Portals/0/Downloads/docs/prod/compshop/HM55BModDocs.pdf, **Fecha de consulta:** 7 de abril del 2010.

13. Parallax , Preliminary DataSheet HM55B, **http://www.parallax.com/Portals/0/Downloads/docs/prod/compshop/HM55BDatasheet.pdf**, **Fecha de consulta:** 7 de abril del 2010.
14. Parallax, Foro acerca HM55B, **http://forums.parallax.com/forums/default.aspx?f=5&m=74964**, **Fecha de consulta:** 8 de abril del 2010.
15. Martín Avilés , Proveedor del sensor utilizado HM55B, **http://www.electroavil-es.com/**, **Fecha de consulta:** 9 de abril del 2010.
16. Honeywell Aerospace Plymouth , Sensores HMC1051/HMC1052L/HMC1053, **http://www.magneticsensors.com/landnav.html**, **Fecha de consulta:** 10 de abril del 2010.
17. Spartons Electronics, SP3002D-4D Kit Compass Development Kit , **http://www.thedigitalcompass.com/index.html**, **Fecha de consulta:** 10 de abril del 2010.