



**Universidad de San Carlos de Guatemala**  
**Facultad de Ingeniería**  
**Escuela de Ingeniería Mecánica Eléctrica**

## **DISEÑO DE UNA ANEMÓGRAFO ELECTRÓNICO**

**PEDRO JAVIER OBREGÓN LECHE**

**ASESORADO POR: ING. ENRIQUE EDMUNDO RUIZ CARBALLO**

**GUATEMALA, JULIO DE 2,005**

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE UN ANEMÓGRAFO ELECTRÓNICO**

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**PEDRO JAVIER OBREGÓN LECHE**  
ASESORADO POR: ING. ENRIQUE EDMUNDO RUIZ CARBALLO

AL CONFERÍRSELE EL TÍTULO DE  
**INGENIERO ELECTRÓNICO**

GUATEMALA, JULIO DE 2,005

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA



### NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	
VOCAL II	Lic. Amahán Sánchez Álvarez
VOCAL III	Ing. Julio David Galicia Celada
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
VOCAL V	Br. Elisa Yazminda Vides Leiva
SECRETARIO	Ing. Carlos Humberto Pérez

### TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Sydney Alexander Samuels Milson
EXAMINADOR	Ing. Fernando Waldemar de León Contreras
EXAMINADOR	Ing. Julio César Solares Peñate
EXAMINADOR	Ing. Manuel Fernando Barrera Pérez
SECRETARIO	Ing. Pedro Antonio Aguilar Polanco

**HONORABLE TRIBUNAL EXAMINADOR**

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**DISEÑO DE UN ANEMOGRAFO ELECTRONICO**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 16 de septiembre de 2004.

Pedro Javier Obregón Leche

## **AGRADECIMIENTOS:**

A mis padres, Carlos Humberto Obregón y Vilma Elisa Leche.

A mis hermanos, Carlos Humberto, Juan Pablo y Nora Briseida.

A mis amigos, Jose Luís Chavarría y Luis Estuardo Velásquez.

A mis compañeros de grupo, Alejandro Vettorazzi, Steeve Toledo, Juan Fernando López, Justo Orlando Cap, Luis Carlos Morales y Victor Morales.

A las familias Vettorazzi González y López Campang.

A la Universidad de San Carlos, en especial a Zaida de Toledo y Anselma Jáuregui.

A mis tías, Alaide González y Stefanía González, por apoyarme durante mi carrera, en mis prácticas y en mi tesis, y brindarme su apoyo en todo momento.

Y a todas aquellas personas que me brindaron su amistad y confianza.



# ÍNDICE GENERAL

<b>ÍNDICE DE ILUSTRACIONES.....</b>	<b>VII</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>IX</b>
<b>LISTA DE SÍMBOLOS.....</b>	<b>XI</b>
<b>GLOSARIO.....</b>	<b>XIII</b>
<b>RESUMEN.....</b>	<b>XV</b>
<b>OBJETIVOS.....</b>	<b>XVII</b>
<b>INTRODUCCIÓN.....</b>	<b>XIX</b>
<b>1. ELEMENTOS EÓLICOS.....</b>	<b>1</b>
1.1. El Viento.....	1
1.1.1. Energía del Viento.....	2
1.1.1.1. Densidad del aire.....	4
1.1.1.2. Área de barrido.....	4
1.1.1.3. Velocidad del viento.....	4
1.2. Mediciones eólicas.....	5
1.2.1. Equipos eólicos de medición.....	6
1.2.1.1. Anemómetro.....	8
1.2.1.2. Anemógrafo.....	11
1.2.1.3. Bota de viento.....	11
1.2.1.4. Veleta.....	12
1.2.1.5. Radar.....	13
1.3. Mediciones eólicas en Guatemala.....	13
1.3.1. Estudios de potencial eólico en Guatemala.....	14
1.3.2. Sitios de mediciones eólicas en Guatemala.....	15

<b>2. FUNDAMENTOS DE INSTRUMENTACIÓN.....</b>	<b>19</b>
2.1 Mediciones.....	19
2.1.1. Transductor.....	22
2.1.2. Dispositivo de procesamiento.....	23
2.1.3. Despliegue de mediciones.....	25
2.2. Análisis de señales.....	26
2.3. Mediciones eólicas.....	29
2.3.1. Parámetros eólicos de medición.....	30
2.3.1.1. Velocidad indicada.....	31
2.3.1.2. Velocidad calibrada.....	31
2.3.1.3. Velocidad verdadera.....	31
<b>3. MICROCONTROLADORES.....</b>	<b>33</b>
3.1 Definición.....	33
3.2 Fabricantes de microcontroladores.....	34
3.2.1 Tipos de microcontroladores.....	36
3.2.1.1 Gama baja.....	36
3.2.1.2 Gama media.....	36
3.2.1.3 Gama alta.....	37
3.3 Arquitectura.....	37
3.3.1 Núcleo.....	39
3.3.1.1 Procesador.....	40
3.3.1.2 Unidad aritmética lógica.....	40
3.3.1.3 Memoria.....	41
3.3.1.3.1 Memoria de programa.....	43
3.3.1.3.2 RAM de datos.....	43
3.3.1.3.3 Memoria EEPROM.....	44

3.3.1.4 Oscilador.....	44
3.3.1.5 Operación de interrupción.....	47
3.3.1.6 Instrucciones.....	47
3.3.2 Periféricos.....	49
3.3.3 Características especiales.....	50
3.4 Herramientas de programación.....	50
<b>4. DISEÑO DE UN ANEMÓGRAFO ELECTRÓNICO.....</b>	<b>55</b>
4.1 Diagrama de Bloques.....	55
4.2 Transductor.....	55
4.2.1 Diseño mecánico.....	56
4.2.2 Diseño eléctrico.....	61
4.3 Acondicionador de señal.....	64
4.3.1 Limitador de sobrevoltajes.....	64
4.3.2 Etapa atenuadora / amplificadora.....	66
4.4 Dispositivo de almacenamiento de datos.....	68
4.4.1 Funcionamiento de la memoria EEPROM.....	70
4.4.2 Diseño electrónico.....	74
4.5 Interfaz serial computadora anemógrafo.....	78
4.6 Despliegue de datos.....	80
4.6.1 Funcionamiento de Display de cristal líquido (LCD).....	81
4.6.2 Diseño electrónico.....	85
4.7 Teclado.....	87
4.7.1 Diseño electrónico.....	87
4.8 Microcontrolador.....	88
4.8.1 Diseño electrónico.....	89
4.8.2 Programación del microcontrolador.....	92
4.8.2.1 Programación de puertos.....	92

4.8.2.1.1 Configuración del puerto analógico y el convertidor analógico digital.....	93
4.8.2.1.2 Configuración de los puertos digitales.....	96
4.8.2.1.3 Adquisición de señales del transductor....	98
4.8.2.2 Programación de Comunicación serial con memoria EEPROM.....	99
4.8.2.2.1 Configuración de registros para comunicación IIC.....	100
4.8.2.2.2 Programas de comunicación.....	104
4.8.2.3 Programación de pantalla de cristal líquido.....	108
4.8.2.4 Programación de la comunicación serial con la computadora.....	110
4.8.2.4.1 Configuración de registros para la comunicación serial.....	111
4.8.2.4.2 Programa para la comunicación serial...	118
4.8.2.5 Programación de teclado.....	121
4.8.2.6 Programación de temporización.....	122
4.8.2.6.1 Configuración de registros del Temporizador TMR0.....	123
4.8.2.6.2 Programa para el temporizador.....	125
4.9 Alimentación.....	128
4.10 Programa de adquisición de datos.....	129
4.10.1 Adquisición de datos.....	130
4.10.2 Configuración de constantes del anemógrafo.....	133
4.10.3 Análisis de registros.....	135
4.11 Funcionamiento del dispositivo.....	137

CONCLUSIONES.....	143
RECOMENDACIONES.....	145
REFERENCIA.....	147
BIBLIOGRAFÍA.....	149
APÉNDICE A CÓDIGO FUENTE DEL PROGRAMA DEL ANEMÓGRAFO....	151



# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Factores biológicos para determinar sitios de medición eólica.....	7
2.	Anemómetro de copas y anemómetro de hélice.....	10
3.	Bota de viento y veleta.....	12
4.	Diagrama de bloques de un instrumento de medición básico.....	20
5.	Despliegue digital.....	26
6.	Microcontrolador PIC18F458.....	37
7.	Memoria de programa de un PIC18F452.....	45
8.	Memoria de datos del PIC18F452.....	46
9.	Circuito del programador de PIC EPIC de Micro Engineering Labs.....	53
10.	Diagrama de bloques que conforman el anemógrafo.....	55
11.	Torques que existen en un anemómetro de copas.....	58
12.	Diagrama físico del transductor del anemógrafo.....	63
13.	Diagrama esquemático del limitador de sobrevoltaje.....	65
14.	Diagrama esquemático del circuito de la etapa atenuadora / amplificadora.....	66
15.	Memoria EEPROM serial, encapsulado y diagrama interno.....	69
16.	Secuencia de transferencia de datos en el bus serial.....	72
17.	Formato del byte de control.....	72
18.	Formato del ciclo de escritura.....	73
19.	Formato del ciclo de lectura.....	73
20.	Diagrama esquemático del banco de memorias seriales EEPROM.....	75
21.	Configuración de terminales de MAX 232.....	79

22.	Diagrama esquemático de conexión serial del microcontrolador con la computadora.....	80
23.	Distribución de la información a mostrar en LCD.....	81
24.	Diagrama de pantalla PVC – 160204A.....	83
25.	Diseño esquemático de conexión de pantalla LCD.....	86
26.	Diseño esquemático del teclado del anemógrafo.....	87
27.	Diagrama esquemático de conexión de microcontrolador PIC18F458....	91
28.	Registro TRISA y su configuración.....	94
29.	Registro ADCON0 y su configuración.....	95
30.	Registro ADCON1 y su configuración.....	96
31.	Configuración del registro TRISB.....	97
32.	. Configuración del registro TRISC.....	98
33.	Configuración del registro TRISD.....	98
34.	Algoritmo de rutina de adquisición de datos.....	99
35.	Registro SSPCON1 y su configuración.....	101
36.	Registro SSPCON2 y su configuración.....	102
37.	Registro SSPSTAT y su configuración.....	104
38.	Configuración de registro SSPADD.....	104
39.	Algoritmo del ciclo de escritura del microcontrolador en la memoria serial EEPROM.....	106
40.	Algoritmo de lectura de memoria EEPROM por el microcontrolador.....	107
41.	Algoritmo de configuración de pantalla de cristal líquido.....	109
42.	Registro INTCON y su configuración.....	112
43.	Registro PIR1.....	112
44.	Registro PIE1.....	113
45.	Registro IPR1 y su configuración.....	114
46.	Registro TXSTA y su configuración.....	115
47.	Registro RCSTA y su configuración.....	117
48.	Configuración de registro SPBRG.....	117

49.	Algoritmo de inicio de transferencia de información.....	119
50.	Algoritmo de transferencia de información.....	120
51.	Algoritmo de funcionamiento del teclado.....	121
52.	Registro T0CON y su configuración.....	124
53.	Algoritmo de control de hora y fecha.....	126
54.	Algoritmo para determinar el número de días por mes.....	127
55.	Diagrama esquemático del circuito de alimentación.....	128
56.	Configuración de puerto serial en AsyncFree.....	130
57.	Algoritmo de recepción de datos del microcontrolador.....	132
58.	Pantalla de inicio del programa de registro eólico.....	133
59.	Menú del programa de Registro Eólico.....	134
60.	Configuración de constantes del anemógrafo.....	134
61.	Menú de configuración de análisis de la información.....	135
62.	Gráfica mensual de velocidades promedio del viento.....	136
63.	Algoritmo de programa principal del microcontrolador.....	138
64.	Algoritmo de funcionamiento del teclado.....	139
65.	Algoritmo de muestreo de datos y almacenamiento.....	141

## **TABLAS**

I.	Formato de muestra almacenada en memoria.....	76
II.	Resumen de comandos de LCD.....	85



## LISTA DE SÍMBOLOS

Kbps      kilo bytes por segundo

MIPS      millones de instrucciones por segundo



## GLOSARIO

<b>Anemógrafo</b>	Instrumento de medición que registra la velocidad del viento.
<b>EEPROM</b>	Memoria electrónica eléctricamente programable y borrrable.
<b>Microcontrolador</b>	Circuito integrado conformado por una unidad central de proceso, puertos de entrada y salida, módulos periféricos y memoria integrada programable.
<b>Potencial eólico</b>	Capacidad de flujo de aire de una región geográfica.
<b>Tacómetro</b>	Dispositivo electromecánico cuya respuesta eléctrica es proporcional a la velocidad de su eje.
<b>Transductor</b>	Dispositivo que convierte de una forma de energía a otra. En los instrumentos de medición electrónicos, de energía mecánica a energía eléctrica.



## RESUMEN

En Guatemala, las estaciones meteorológicas, donde se registran la velocidad del viento, utilizan anemómetros y anemógrafos analógicos, lo cual requiere de un monitoreo diario de las estaciones para obtener datos eólicos. Un anemógrafo electrónico elimina la necesidad de monitorear a diario cada estación, además, se almacenan datos eólicos de grandes períodos de tiempo. Estos datos son generadores de información para determinar el potencial eólico del sitio de medición.

Un anemógrafo tiene aplicación en el ámbito nacional en meteorología, en aeronáutica y en investigaciones de generación de energía eólica.

El diseño de un anemógrafo electrónico, fue dividido en las siguientes etapas:

- Documentación de recursos y mediciones eólicas.
- Diseño de instrumentos.
- Diseño con microcontroladores.
- Diseño de un transductor eólico-eléctrico.
- Diseño de adquisición, procesamiento, almacenamiento, despliegue y transferencia de datos.

El anemógrafo registra la velocidad promedio del viento, fecha y hora de la medición, en un período programado por el usuario. Posee una interfaz para transferir la información a una computadora. En la computadora, se analizan los registros para mostrar de forma gráfica el comportamiento del viento.

## **OBJETIVOS**

- **GENERAL**

Diseñar un dispositivo electrónico capaz de registrar la velocidad del viento, para realizar mediciones meteorológicas y además determinar el potencial eólico de una región en donde se desee implementar una central eólica.

- **ESPECÍFICOS**

1. Definir criterios de selección para el diseño y construcción de un transductor eólico – eléctrico.
2. Describir el funcionamiento de un microcontrolador como procesador de señales analógicas.
3. Diseñar un software de aplicación para poder analizar las mediciones realizadas por el dispositivo diseñado y determinar si una región posee el potencial eólico necesario para instalar una central eólica.
4. Determinar los parámetros necesarios a considerar para calibrar un instrumento de medición eólica.



## INTRODUCCIÓN

Las mediciones eólicas en el país, son realizadas en estaciones meteorológicas del país, donde se cuenta con anemómetros y anemógrafos analógicos. Estos instrumentos son fabricados en el extranjero y eso motiva a que existan pocas estaciones donde se registren datos eólicos. Los datos eólicos tienen importancia para la generación de energía eólica, como alternativa a la generación de energía con hidrocarburos y con recursos hidráulicos.

Es necesario diseñar y fabricar en el país, instrumentos de medición y dispositivos electrónicos en general, para reducir costos y además cubrir con la demanda en diferentes áreas. Basándose en esta necesidad, el siguiente estudio detalla el diseño de un anemógrafo electrónico para registrar la velocidad del viento y el procesamiento de estos datos para determinar el potencial eólico de una zona.

El presente trabajo de graduación tiene como objetivo diseñar un anemógrafo electrónico, el cual permitirá registrar datos de la velocidad promedio del viento, desplegar esta información y transferir los datos a una computadora para su análisis.

En el primer capítulo, se dan los parámetros básicos de la energía eólica, mediciones eólicas, tipos de instrumentos de medición eólica y las estaciones de medición eólica en Guatemala, así como las instituciones que se dedican a ello.

En el segundo capítulo, se hace una reseña de los fundamentos de instrumentación, así como de los elementos que conforman un instrumento (transductor, dispositivo de procesamiento y despliegue de datos). Además, se definen las características del análisis de señales en instrumento. Por último, se definen parámetros básicos y de interés en las mediciones de viento.

En el tercer capítulo, se describe las características físicas y lógicas de un microcontrolador, para determinar dentro de los existentes, el indicado para la construcción del anemógrafo.

Por último, se describe la etapa de diseño del anemógrafo digital. Donde iniciamos con el reconocimiento de cada uno de los elementos que deben de conformar el dispositivo: transductor eólico-eléctrico, acondicionador de señal, microcontrolador, teclado, despliegue de datos, interfaz con computadora y el dispositivo de almacenamiento externo de datos. Luego, procedemos al diseño electrónico de cada uno de los componentes y por último describimos el programa de control de cada uno de éstos.

El diseño del anemógrafo electrónico, es útil para determinar características básicas del diseño de un transductor eólico-eléctrico, además de mostrar algunas aplicaciones de microcontroladores en control de instrumentos.

El estudio del anemógrafo electrónico, contribuye en la investigación del diseño de dispositivos en el medio nacional, además de establecer bases, para la investigación del potencial eólico en Guatemala en búsqueda de generación de energía con recursos renovables.

# 1. ELEMENTOS EÓLICOS

## 1.1. El Viento

La atmósfera es un enorme intercambiador de calor, alimentado por el sol, que transfiere calor de una parte de la tierra a otra. La transferencia se da por medio del movimiento del aire, el cual, mediante convección, transfiere el calor de las partes calientes a las más frías. El aire en movimiento es conocido como viento.

Cuando el sol calienta la superficie de la tierra, el aire cercano a ésta se calienta, lo cual hace disminuir su densidad. Como el aire de las partes superiores de la atmósfera es más frío que en la superficie de la tierra, por ende más denso, este desplaza el aire de la superficie, y el de la superficie se mueve hacia arriba. El ciclo se completa, debido a que el aire frío que esta ahora en contacto con la superficie se calienta y el aire caliente que se elevó en la atmósfera perdió energía y se enfrió, y nuevamente se desplazan entre sí. Este ciclo es el que propicia el movimiento del aire.

El viento se define como aire en movimiento debido a un gradiente de temperatura de la atmósfera que varía respecto a la altura a evaluar. Cuando existe un diferencial de temperatura en dos puntos de la atmósfera, el aire siempre fluirá del punto de mayor temperatura hacia el punto de menor temperatura. Este fenómeno es debido al cambio de la densidad que sufre el aire con la temperatura, el aire frío es más denso que el aire caliente, por lo

que siempre será desplazado por este. El sentido del viento, está en función de la hora del día, ya que depende exclusivamente del calor del sol proporcionado a la tierra.

### **1.1.1 Energía del Viento**

La energía del viento es la característica más importante para cualquier persona que desea instalar una turbina eólica o diseñar instrumentos de medición eólica. Debido a que el viento no es nada más que un fluido gaseoso (aire), para analizar la energía debida a este debemos de analizarlo de acuerdo a la dinámica de fluidos. Para esto utilizaremos la ecuación de continuidad de Bernoulli. En donde la energía del fluido es debido a la presión a la que es sometido, la altura a donde es elevado y la velocidad a la que este se mueve. El viento es aire en movimiento, entonces su principal componente de energía si asumimos un mismo nivel y por lo tanto una presión constante, es la componente de energía cinética.

La energía cinética se expresa de la siguiente forma:

$$K = \frac{1}{2} * M * V^2 \quad (\text{Ec. 1})$$

Donde

- K es la energía cinética del viento en joules.
- M es la masa del viento en Kg.
- V es la velocidad del viento en m/s.

Para un fluido, la masa esta definida en función de su densidad, para lo cual, la masa del viento es:

$$M = \rho * A * V * t \quad (\text{Ec. 2})$$

Donde

- M es la masa del viento en Kg.
- $\rho$  es densidad del viento en Kg/m<sup>3</sup>.
- A es el área donde circula el viento en m<sup>2</sup>.
- V es la velocidad del viento en m/s.
- t es el período de tiempo de muestreo en segundos.

Entonces sustituimos la ecuación 2 en la ecuación 1 y obtenemos la energía del viento (Ec. 3) en función de la densidad, el área de circulación del viento, el período de medición y la velocidad de éste.

$$E = \frac{1}{2} * \rho * A * t * V^3 \quad (\text{Ec. 3})$$

Ahora deseamos saber la cantidad de energía por unidad de tiempo, disponible en el viento, para lo cual derivamos la ecuación 3 respecto al tiempo y obtenemos la potencia del viento.

$$P = \frac{1}{2} * \rho * A * V^3 \quad (\text{Ec. 4})$$

De la anterior ecuación, podemos deducir que a medida que aumenta cualquier de las variables anteriores, también aumentará la potencia disponible en el viento.

#### **1.1.1.1. Densidad del aire**

La densidad del viento es inversamente proporcional a la temperatura. Lo cual quiere decir que a medida que aumenta la temperatura del viento, así disminuye la densidad de éste. Pero la densidad del viento es más afectada por los cambios de altura que por los cambios de temperatura. La densidad del viento también es inversamente proporcionalmente a la altura, por lo que a mayor altura tendremos un viento con menor densidad.

#### **1.1.1.2. Área de barrido**

El área de barrido o de circulación, es uno de los parámetros más importantes al momento de diseñar dispositivos eólicos, ya que como observamos, la potencia del viento está en función del área de circulación de viento evaluada. El área de barrido de una turbina esta en función del radio de sus aspas. Por lo tanto, mientras más potencia del viento, queramos obtener, debemos de aumentar el área de dispositivo que ha de estar en contacto con el viento.

#### **1.1.1.3. Velocidad del viento**

De acuerdo a la ecuación 4, observamos que la potencia del viento depende principalmente de la velocidad del viento elevada al cubo. La velocidad del viento es a su vez dependiente de la altura de donde se coloque el

dispositivo para medir el viento (anemómetro) o para obtener potencia de él (turbina eólica), esta aumenta conforme aumentamos la altura.

De las características vistas anteriormente, hemos observado que la única que podemos variar para obtener una mayor potencia del viento es únicamente el área de barrido, mientras que la densidad y la velocidad del viento son factores dependientes del clima, la altura y la topografía del terreno.

## **1.2. Mediciones eólicas**

La potencia del viento depende en gran parte de la velocidad de éste, por lo cual es importante realizar mediciones para determinar lugares con altas velocidades. Las mediciones eólicas comprenden todas aquellas mediciones destinadas a determinar la velocidad del viento y su dirección. Algunas de las principales aplicaciones de las mediciones eólicas son:

- **Mediciones meteorológicas:** están orientadas a registrar velocidades del viento, con el objeto de determinar la dirección de tormentas y el tiempo en el que éstas afectarán a una región.
- **Mediciones aeronáuticas:** se registra la velocidad del viento y su dirección para proporcionar estos datos a los pilotos. Cuando un avión aterriza lo hace con el viento en su contra y para despegar lo hace a favor del viento. Además sirve para determinar regiones con vientos altos, que pueden hacer que los pilotos pierdan el control de la aeronave.
- **Potencial eólico:** estas mediciones tienen como objetivo determinar regiones con velocidades de viento elevadas, que permitirán la

instalación de turbinas eólicas. Como la potencia del viento depende primordialmente de la velocidad del viento, se buscan regiones donde hayan viento elevados, pero más que eso, se busca regiones donde la velocidad del viento en promedio sea elevada o por lo menos constante.

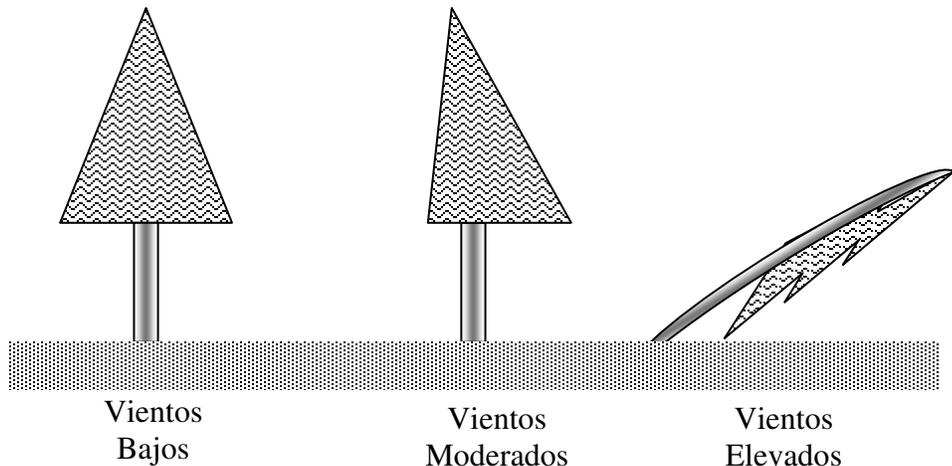
### **1.2.1. Equipos eólicos de medición**

En la actualidad existe una amplia variedad de dispositivos para determinar la velocidad del viento y la dirección de este. Estos equipos pueden ir desde sencillos como una veleta hasta complejos como los dispositivos electrónicos. Antes de instalar un equipo de medición eólico debemos de atender a observar otros factores, como lo son la topografía del terreno y factores biológicos.

La topografía del terreno, nos puede ayudar a determinar cañones de viento, donde éste recorre largos tramos entre montañas con salida al mar. Este es el caso del Cañón de Palín, que es uno de los sitios en Guatemala con mayor potencial eólico, no tanto por su altura, sino por su topografía, al encontrarse entre montañas que conducen el viento desde el océano.

Los factores de tipo biológico, nos pueden ayudar a decidir donde instalar un medidor. El factor biológico más utilizado es la forma de los árboles, que de acuerdo a la velocidad del viento, así variará la forma de éstos. Un árbol ubicado en una región con vientos moderados, mostrará una leve inclinación permanente en dirección del viento. Si los vientos son elevados, estos pueden variar la forma de un árbol, como dejar un lado totalmente descubierto de ramas o en algunos casos extremos, llegar a derribarlo.

**Figura 1.** Factores biológicos para determinar sitios de medición eólica.



Al momento de instalar un instrumento de medición eólica se debe de seleccionar un lugar que no posea construcciones cercanas e incluso árboles. Además se debe de colocar el instrumento de medición a una altura no menor de 10 metros. Además se debe de cumplir con normas internacionales para realizar mediciones, aparte de la altura mínima de medición, se debe de calibrar el instrumento cada seis meses. Debido a que en Guatemala, no existen normas sobre instrumentos de medición eólica, utilizaremos los criterios establecidos por la Oficina de Normalización Nacional de los Estados Unidos para instrumentos eólicos.

Algunos de los instrumentos que son más utilizados para realizar mediciones de velocidad y dirección del viento son:

- Anemómetro.
- Anemógrafo.
- Bota de viento.
- Veleta.
- Radar.

#### **1.2.1.1. Anemómetro**

El anemómetro es un instrumento que nos da lecturas inmediatas de la velocidad de viento y en algunos casos posee dispositivos para registrar la dirección de éste. Dentro de los anemómetros existen dos variedades, los dispositivos analógicos y los digitales. Los analógicos poseían un galvanómetro para desplegar la velocidad del viento. En el caso de los digitales, utilizan convertidores de analógico a digital para desplegar el resultado en un display o en una pantalla LCD.

Los anemómetros miden la velocidad instantánea del viento, pero las ráfagas (fluctuaciones habituales del viento) se producen con tal frecuencia que restan interés a dicha medición, por lo que se toma siempre un valor medio en intervalos de 10 minutos.

Los anemómetros se pueden ser clasificados de acuerdo al transductor utilizado:

- Los de empuje están formados por una esfera hueca y ligera (Daloz) o una pala (Wild), cuya posición respecto a un punto de suspensión varía con la fuerza del viento, lo cual se mide en un cuadrante.

- El anemómetro de rotación está dotado de copas (Robinson) o hélices unidas a un eje central cuyo giro, proporcional a la velocidad del viento, es registrado convenientemente; en los anemómetros magnéticos, dicho giro activa un diminuto generador eléctrico que facilita una medida precisa.
- El anemómetro de compresión se basa en el tubo de Pitot y está formado por dos pequeños tubos, uno de ellos con orificio frontal (que mide la presión dinámica) y lateral (que mide la presión estática), y el otro sólo con un orificio lateral. La diferencia entre las presiones medidas permite determinar la velocidad del viento.

Por su construcción, los anemómetros de rotación pueden ser de dos tipos:

- De copas
- De hélice.

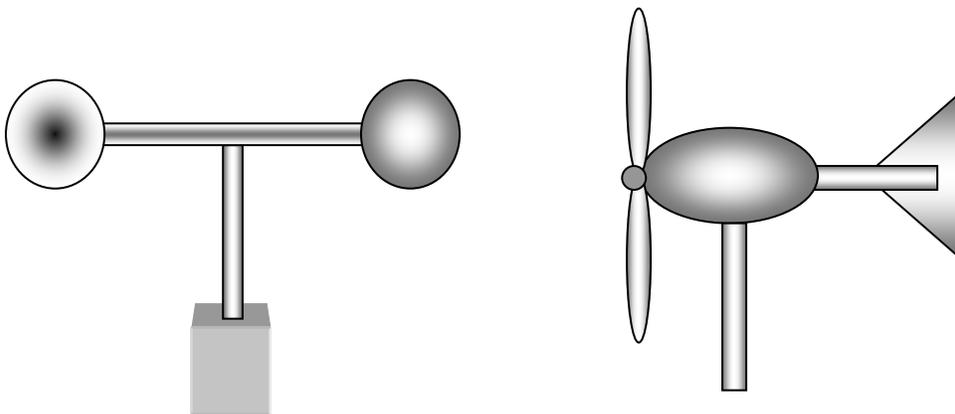
La única diferencia entre los anemómetros de copas y los de hélice son su construcción, ya que ambos utilizan el principio de utilizar una pieza móvil (las copas o la hélice) que gira proporcionalmente a la velocidad del viento. Esta pieza móvil está unida por un eje a un dispositivo que genera una señal eléctrica proporcional a la señal mecánica. Esta señal eléctrica es proporcional a la velocidad del viento.

Por el tipo de transductor, los anemómetros pueden ser tacómetros, que generan un voltaje que varía de amplitud así como varíe la velocidad del viento. También son utilizados los decoders, que generan una serie de pulsos que

poseen una frecuencia que es igual a la frecuencia mecánica con que se mueve la hélice o las copas.

Los anemómetros de copas son turbinas eólicas de eje vertical, son omnidireccionales y pueden girar con vientos que provengan de cualquier dirección. En cambio, los anemómetros de hélice, son pequeñas turbinas eólicas de eje horizontal, por lo cual necesitan de un dispositivo para orientarse en dirección del viento, en la mayoría de casos es utilizada una cola para lograrlo. La ventaja de los anemómetros de copas sobre los de hélice es que pueden girar con vientos que provengan de cualquier dirección, mientras que los anemómetros de hélice, necesitan de un sistema de orientación. La ventaja de los anemómetros de hélice sobre los anemómetros de copas, es que nos pueden proporcionar la dirección del viento.

Figura 2. Anemómetro de copas y anemómetro de hélice.



### **1.2.1.2. Anemógrafo**

El anemógrafo, comparado en construcción con el anemómetro, no tiene diferencia alguna. El anemógrafo es un anemómetro con capacidad de almacenar los datos registrados durante un período de tiempo determinado. Existen de dos tipos, los anemógrafos analógicos y los anemógrafos digitales.

Los anemógrafos analógicos almacenan las mediciones marcando con una aguja, un rollo de papel que está constantemente girando, su velocidad se puede variar de acuerdo a la resolución de las mediciones que deseemos realizar. El tiempo que puede estar realizando mediciones depende exclusivamente del tamaño del rollo de papel.

Los anemógrafos digitales almacenan los datos en memorias no volátiles (EPROM y EEPROM) la resolución de los datos es inversamente proporcional con la cantidad de tiempo que queremos analizar. Debido a que la memoria consta de una cantidad de registros limitados, podemos variar la resolución de acuerdo con el tiempo de mediciones que deseemos almacenar,

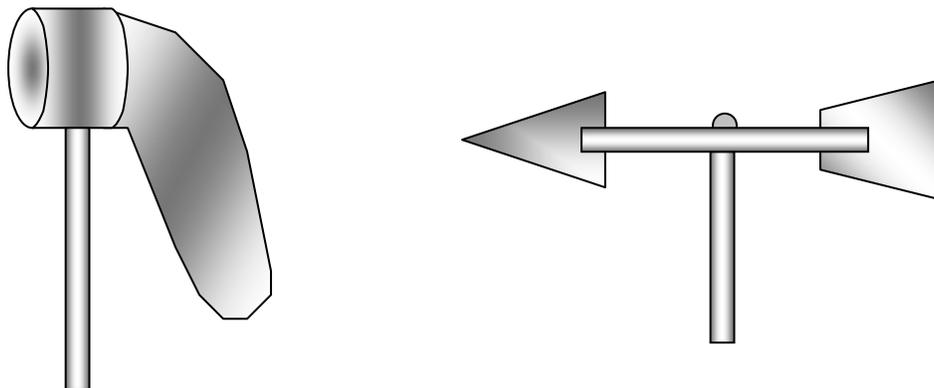
### **1.2.1.3. Bota de viento**

La bota de viento nos permite realizar mediciones sobre la dirección del viento y nos da una aproximación de la velocidad del viento. La bota de viento es una bolsa de nylon, material impermeable, que de acuerdo a la velocidad de velocidad del viento y dirección se orienta hacia este, y se infla de forma proporcional al caudal de aire que pasa a través de ella. Las botas de viento son muy utilizadas en los aeropuertos para tener datos inmediatos de la dirección del viento y su valor estimado de velocidad.

#### 1.2.1.4 Veleta

La veleta es uno de los instrumentos más antiguos para determinar la dirección del viento. Es una pequeña lámina de metal o de algún polímero bastante ligero para poder moverse con el más ligero viento. Esta lámina de metal es más delgada en uno de sus extremos, lo que hace que el extremo más grueso siempre apunte en dirección a favor del viento, mientras que la lámina delgada en posición contraria. Este puede ser un instrumento visual, aunque en la actualidad se han diseñado veletas electrónicas, que constan de una serie de pares emisores – receptores de infrarrojo, que nos permiten interpretar los diferentes estados para determinar la dirección del viento.

**Figura 3.** Bota de viento y veleta.



#### **1.2.1.4. Radar**

Este dispositivo funciona con una antena emisora y con otra receptora, que se encargan de emitir pulsos electromagnéticos para luego detectar la onda reflejada, los cambios en la onda reflejada nos pueden servir para determinar perturbaciones, que pueden ser lluvias y para nuestro interés corrientes de aire. También los hay del tipo ultrasónico. Estos dispositivos giran sobre un eje para así cubrir un perímetro circular, cuyo radio depende de la resolución de los datos a procesar y de la potencia de emisión y recepción de las antenas. Son de los dispositivos más utilizados en el mundo para realizar mediciones meteorológicas, pero de igual forma, son de los dispositivos más caros para efectuar tales mediciones.

### **1.3 Mediciones eólicas en Guatemala**

En Guatemala existen varias entidades que se han dedicado a la medición de la velocidad del viento, ya sea para usos meteorológicos, datos aeronáuticos y para estudios de potencial eólico en el país. El Instituto de Sismología, Vulcanología, Meteorología e Hidrología (INSIVUMEH) es la institución que posee un amplio registro de velocidades del viento en diferentes sitios en Guatemala. La mayoría de estos datos son recolectados con anemómetros.

El INSIVUMEH realiza mediciones diarias de la velocidad del viento en diferentes estaciones meteorológicas ubicadas en diferentes sitios del país y en la mayoría de los aeropuertos nacionales. Las estaciones meteorológicas del país registran lecturas diarias de la velocidad del viento, para esto utilizan anemómetros digitales en su mayoría y toman lecturas del viento cada seis horas. Poseen algunas estaciones donde aún se utilizan aún anemógrafos del

tipo analógico pero debido a la necesidad de cambiar el rollo de papel que registra las velocidades se ha discontinuado su uso en algunas estaciones.

En el caso de las estaciones ubicadas en los aeropuertos del país, éstas registran el viento, únicamente en las horas de tráfico aéreo. Se utilizan anemómetros, debido a la necesidad de obtener una lectura inmediata de la velocidad y dirección del viento, para proporcionarla a los pilotos que pretenden despegar o aterrizar en los aeropuertos.

### **1.3.1 Estudios de potencial eólico en Guatemala**

El aprovechamiento de la energía del viento, se ha dado desde hace ya muchos siglos en varios lugares del mundo. Una de las principales aplicaciones que ha tenido la energía eólica ha sido la de mover los molinos para cereales y la extracción de agua de pozos y ríos.

En Guatemala, su principal aplicación es la de bombeo de agua de pozos. Pero en la actualidad en países como Dinamarca, Holanda, España y en especial Estados Unidos, se emplean turbinas eólicas, que generan energía eléctrica a partir de la energía del viento. En estos países, ya se cuenta con sistemas eléctricos híbridos eólicos. Se emplea la cogeneración de energía eléctrica por parte de turbinas eólicas, para luego conectarse con las líneas de distribución locales y de esta forma cubrir una porción de la demanda de energía eléctrica de cualquier región cercana.

Antes de invertir en la instalación de una planta de generación de energía eólica, o por lo menos, en una pequeña turbina eólica, es necesario realizar un estudio del potencial eólico de la zona en cuestión. Instalar una turbina eólica

en una zona con poco viento, es como instalar una bomba de agua en un pozo seco. Si deseamos obtener el mayor rendimiento de nuestra turbina, debemos de realizar mediciones para determinar el punto con mayores velocidades de viento en el transcurso del año.

Ante la necesidad de investigar sobre fuentes de energía renovable, varias instituciones se han dedicado a recolectar información sobre el potencial eólico de diferentes regiones del país, para determinar los puntos donde sería factible instalar turbinas eólicas.

Algunas de las instituciones que han realizado estudios de potencial eólico en diferentes regiones en Guatemala son:

- Instituto Nacional de Sismología, Vulcanología, Meteorología e Hidrología (INSIVUMEH)
- Instituto Nacional de Electrificación (INDE)
- Ministerio de Energía y Minas (MEM)
- National Rural Electric Cooperative Association (NRECA)

### **1.3.2 Sitios de mediciones eólicas en Guatemala**

Para determinar los sitios de mediciones eólica, se han tomado en cuenta factores topográficos, biológicos e incluso sociales. Los factores topográficos, consideran los cañones de viento, los pasos entre montañas, las cuencas de los ríos y las zonas costeras. Los factores biológicos comprenden la forma de la vegetación, ya que ésta es afectada por la velocidad del viento que incide sobre ella. Y en los factores sociales, se realiza una investigación con las

personas de diferentes regiones para obtener datos cualitativos de la velocidad del viento y de las regiones consideradas de grandes vientos.

Entre las estaciones instaladas por el INSIVUMEH se encuentran:

- Observatorio Nacional
- Aeropuerto La Aurora
- Aeropuerto de Flores
- Panzós, Izabal
- Esquipulas, Chiquimula
- San Luis, Jutiapa
- Cobán, Alta Verapaz
- Los Esclavos, Santa Rosa
- Olintepeque, Quetzaltenango
- Huehuetenango, Huehuetenango
- Puerto de San José, Escuintla
- Retalhuleu, Retalhuleu

Algunas de las estaciones de medición instaladas por NRECA son:

- La Capellanía, Huehuetenango
- Comapa, Jutiapa
- El Boquerón, San Marcos
- Palín, Escuintla
- San Luis, Jutiapa

Los sitios de medición utilizados por el INDE son:

- Xayá, Petén
- Bobos, Izabal
- Carbonera, Chiquimula
- Ciudad de Guatemala
- Pueblo Viejo, Alta Verapaz
- San Cristóbal, Alta Verapaz
- Cerro La Laguna, Alta Verapaz
- La Capellanía, Huehuetenango

Los sitios de medición eólica utilizados por el Ministerio de Energía y Minas son:

- Santa Elena, Petén
- Poptún, Petén
- Puerto Barrios, Izabal
- San José, Escuintla
- Ciudad de Guatemala
- Retalhuleu, Retalhuleu
- Huehuetenango, Huehuetenango

De acuerdo a los estudios realizados por las anteriores instituciones, los lugares que poseen mayor potencial eólica en Guatemala son Palín, Escuintla y La Capellanía, Huehuetenango, en donde las velocidades promedio de viento son mayores a 5 m/s en el transcurso del año.



## 2. FUNDAMENTOS DE INSTRUMENTACIÓN

### 2.1 Mediciones

Una medición es una comparación de una magnitud con otra, tomada como patrón, hecha a través de un instrumento en forma indirecta, ya que el patrón generalmente no es accesible. Para determinar el valor físico de la magnitud de la variable deseada en un proceso, es necesario hacer uso de un instrumento para su medición. Los instrumentos nos permiten registrar los valores de variables físicas que serían muy difíciles o imposibles de cuantificar sensorialmente.

Todos los instrumentos de medición poseen características independientes del tipo de variable a determinar su magnitud, estas son:

- **Precisión:** Es la capacidad de un instrumento de reproducir su indicación al medir repetidamente una misma magnitud.
- **Exactitud:** Es la capacidad de un instrumento de indicar el valor real.
- **Sensibilidad:** Es la relación entre la magnitud de la respuesta de un instrumento y la magnitud del parámetro que se está midiendo.
- **Resolución:** Es el cambio más pequeño que se puede detectar con un instrumento.

- **Error:** es la diferencia que existe entre el valor real y el valor medido de la variable.

De acuerdo a estas características descritas, podremos realizar comparaciones entre distintos instrumentos de medición. Cuando trabajamos, muchas veces es deseable tener un instrumento preciso a uno exacto, ya que si tenemos un instrumento que siempre nos da la misma lectura, podremos determinar un factor de corrección respecto al valor real de la variable. Mientras mayor sea la exactitud del instrumento, así aumentará su precio. De la misma forma, es mucho más complejo elaborar instrumentos para registrar resoluciones muy pequeñas de una variable medida, por los niveles de señales muy bajos, a pesar que éstos pueden ser amplificados.

Los instrumentos se pueden clasificar de dos formas de acuerdo a su función:

- **Instrumento indicador:** estos instrumentos únicamente muestran un valor instantáneo de la variable medida. En algunos casos, los instrumentos despliegan valores promedio de un intervalo de medición.
- **Instrumento registrador:** este tipo de instrumento nos permite una lectura promedio de la variable, pero a la vez, almacena estos valores promedio a lo largo de un período de tiempo, que depende de su capacidad de almacenar datos.

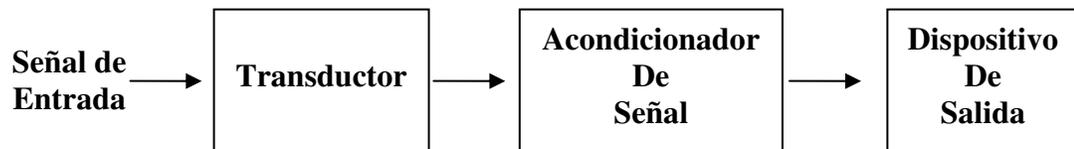
Los instrumentos a su vez, también se pueden clasificar por el tipo de variables a medir:

- Mecánicas
- Eléctricas
- Térmicas
- Químicas
- Ópticas

Un instrumento de medición está constituido de tres elementos principales:

- Un dispositivo de entrada (transductor).
- Un acondicionador de señal o dispositivo de procesamiento
- Un dispositivo de salida (despliegue de mediciones)

**Figura 4.** Diagrama de bloques de un instrumento de medición básico.



### 2.1.1 Transductor

La principal función del dispositivo transductor es la de enviar al sistema acondicionador de señal, una señal eléctrica que es proporcional a la señal de entrada.

Los transductores son dispositivos que nos transforman energía de un tipo a otro tipo de energía. Los dispositivos a tratar en este trabajo de investigación serán los transductores eléctricos o comúnmente llamados sensores, estos pueden recibir señales físicas de naturaleza diversa (señales mecánicas, térmicas, químicas, etc.) y darnos en la salida señales de tipo eléctrico. Un transductor es un dispositivo que al ser afectado por la energía de un sistema de transmisión del proceso medido, nos proporciona energía en la misma forma o en otra a un segundo sistema de transmisión. La transmisión de energía puede ser eléctrica, mecánica, química, óptica o térmica.

Los transductores eléctricos pueden ser clasificados por el origen de la energía de entrada del sistema, por el tipo de aplicación (presión, nivel, flujo, temperatura, etc.), por la naturaleza de la señal de salida, etc.

Los transductores eléctricos se pueden clasificar también de acuerdo al principio eléctrico en el cual basan su funcionamiento. Los transductores se pueden clasificar así de dos formas:

- **Transductores pasivos:** estos producen una variación en algún parámetro eléctrico (resistencia, capacitancia, inductancia, etc.) de acuerdo a los cambios de la señal de entrada. Es necesario armar un circuito para poder registrar la forma en que los cambios de los

parámetros del transductor afectan otras variables eléctricas del circuito como el voltaje, la corriente y la frecuencia de éste entre otras variables eléctricas.

- **Transductores activos:** estos son llamados los transductores del tipo de autogeneración, ya que generan un voltaje o corriente analógica cuando son estimulados por medio de alguna forma física de energía. Estos nos requieren de alimentación externa para funcionar.

### 2.1.2 Dispositivo de procesamiento

Los dispositivos de procesamiento de señales o de acondicionamiento, tienen como función principal la de amplificar la señal de entrada, filtrarla y modificar su formato para poder ser desplegada en el dispositivo de salida. Algunas de las funciones básicas del acondicionador de señales son:

- **Acoplar la señal de entrada y la salida:** el acondicionador debe de acoplar eléctricamente la señal de entrada del transductor para evitar un efecto de carga en la señal de éste, lo que nos resultaría en valores reducidos de señal además de distorsionados. El caso de acople de la salida, es igual, se evita que exista un efecto de carga por parte del dispositivo que desplegará la medición, ya sea de forma analógica o de forma digital.
- **Amplificar la señal:** muchas veces los transductores tienen una buena resolución de la señal de entrada, pero las señales eléctricas que llegan hacia el acondicionador pueden ser de muy pequeña amplitud, por lo que

es necesario amplificar las señales a niveles adecuados para poder utilizar éstas señales.

- **Filtrar la señal:** la señal proveniente del transductor puede ser afectada por varios tipos de ruidos (señales indeseadas de naturaleza eléctrica o térmica), que ocasionan distorsiones de ésta, para eso es necesario filtrarla para eliminar cualquier otra señal que ocasiona errores en el procesamiento. Los filtros también pueden ocasionar ruido e incluso el mismo dispositivo acondicionador, pero esto debe ser tomado en cuenta por el diseñador para realizar las correcciones necesarias para afectar el despliegue de la medición.
- **Digitalización de la señal:** la digitalización de la señal tiene como objeto primario, cuantizar valores analógicos de señales, para poder desplegarlos de forma digital o como etapa anterior a dispositivos de procesamiento digital de señales.
- **Procesamiento de la señal:** esta etapa es la encargada de modificar la señal de entrada de tal manera que el desplegar muestre el valor medido. Esta etapa puede incluir, secciones de filtrado, amplificación, corrección de nivel, entre otras. Si la señal es digitalizada, este proceso se puede realizar con un procesador digital de señales (DSP), el cual evita la necesidad de elaborar circuitos para el tratamiento analógico de las señales. El DSP es un tratamiento para señales discretas, ya que en la mayoría de los casos, estamos realizando una medición discreta de señales continuas.

### **2.1.3 Despliegue de mediciones**

La principal función de este dispositivo es la mostrar en forma gráfica el valor de la variable medida, de acuerdo a lo registrado por el instrumento. El despliegue de señales puede ser analógico o digital.

Un despliegue analógico utiliza un dispositivo electrodinámico, el más utilizado es un galvanómetro. El galvanómetro esta constituido por una bobina y un imán permanente o por dos bobinas. El funcionamiento es bastante sencillo, a medida que aumenta la señal eléctrica de la bobina, así aumenta proporcionalmente su campo magnético, y debido a la interacción de campos magnéticos (ya sea por un imán permanente o por otra bobina) existente existe un par resultante. Una de las bobinas se encuentra sobre un cojinete que permite que ésta se desplace libremente de acuerdo al torque resultante. En el eje de esta bobina se encuentra instalado un indicador (una aguja) que se desplaza sobre una superficie marcada, que nos indica por medio de una escala, el valor de la medición.

El despliegue digital, recibe un valor digital de salida del dispositivo acondicionador de señales, por medio de indicadores luminosos nos muestra una lectura numérica del valor de la medición. Los indicadores digitales pueden ser contruidos con displays (arreglo de leds) o con pantallas de cristal líquido (LDC).

**Figura 5.** Despliegue digital. ([www.jameco.com](http://www.jameco.com))



## 2.2 Análisis de señales

Una señal está definida como una cantidad física que lleva información. Las señales nos sirven para determinar el comportamiento de sistemas físicos. La información en la señales, está en función de la variación de la señal, ya que si no existiera variación en ésta, la información procesada sería mínima. Una de las ventajas de utilizar señales en cualquier sistema de medición de un proceso, es que las señales se pueden representar matemáticamente, por medio de funciones dependientes del tiempo y en algunos casos de otras variables del proceso.

Las señales se pueden clasificar de acuerdo a su distribución respecto el tiempo:

- **Señales continuas:** mejor conocidas como señales analógicas, representan mediciones del tipo analógico, donde éstas representan de forma continua los cambios en la variable de medición.

- **Señales discretas:** estas señales se toman en períodos de tiempo definidos en el instrumento. Las mediciones representan valores discretos de la variable medida.

Además las señales se pueden clasificar por su amplitud, independientemente de su distribución respecto al tiempo.

- **Señales analógicas:** si las señales poseen valores de amplitud continuos.
- **Señales digitales:** son señales con valores discretos de amplitud.

Esto nos permite clasificar a los instrumentos como analógicos y digitales. Los sistemas de medición analógicos son aquellos en que tanto las señales de entrada y salida son continuas en el tiempo y en amplitud. Los sistemas digitales de medición son aquellos en los que sus señales de entrada y salida son señales en tiempo discreto.

Las señales de tiempo discreto pueden tener como origen señales discretas, pero de igual forma, pueden ser obtenidas de muestrear señales continuas en el tiempo. Para nuestro estudio, utilizaremos el tratamiento de señales para tiempo discreto después de muestrear de señales continuas en el tiempo (velocidad del viento).

El teorema fundamental de muestreo (Teorema de Nyquist) permite representar una señal de tiempo continuo a partir de una secuencia de muestras. Para esto, únicamente debemos de observar que la frecuencia de muestreo sea superior a la máxima frecuencia de la señal (como mínimo el doble) para evitar distorsión en la señal reconstruida. Para recuperar la señal a partir de sus muestras discretas, se necesita de tan solo un filtro pasabajos (este será digital para poder procesar los datos registrados en la computadora).

Dentro de las técnicas utilizadas para el tratamiento en tiempo discreto para muestreo de señales en tiempo continuo, es la utilización de filtros digitales tanto para la señal de entrada como para la señal de salida. Estas técnicas de filtrado y el procesamiento digital, nos permitirán realizar una representación real del comportamiento del viento luego de obtener los datos registrados por el instrumento.

En el anemógrafo, se registrará un valor promedio de viento cada período de tiempo, este promedio será el resultado de la suma de todos los valores instantáneos registrados durante períodos de tiempo más cortos entre el número de mediciones realizadas en ese período. Esto es debido a que el viento presenta un comportamiento demasiado aleatorio, por lo cual, será necesario tomar un elevado número de muestras antes de poder registrar o desplegar una velocidad promedio. Además este determinará la desviación media y estándar de los datos respecto a la media aritmética. Todos estos datos serán registrados para luego ser recuperados por una computadora y en el programa de análisis de potencial eólico los utilizaremos para realizar un modelo del comportamiento real del viento.

Los métodos estadísticos utilizados por el instrumento para la obtención de una velocidad promedio serán:

- Media aritmética
- Desviación medía
- Desviación estándar

En el programa de análisis eólico, se utilizará procesamiento de señales en tiempo discreto para poder procesar toda la información almacenada por el instrumento. El objetivo de este procesamiento digital de señales, es de poder determinar valores reales de velocidades del viento, de acuerdo a los datos obtenidos en las mediciones realizadas. Para los datos existentes en el país, no se ha realizado un análisis de los datos obtenidos, y además la forma en que fueron realizadas las mediciones, no nos permiten determinar patrones reales del comportamiento de la velocidad del viento.

### **2.3 Mediciones eólicas**

En el caso de las mediciones eólicas, hay que considerar algunos parámetros para el diseño de un instrumento de medición eólico. La altura del dispositivo y el área de barrido.

Debemos de recordar que la velocidad del viento está en función de la altura en la que nos encontremos. Las normas de la Oficina de Normalización Nacional de los Estados Unidos, han determinado como estándar de medición de la velocidad del viento la altura de 10 metros sobre el suelo.

El área de barrido determinará la cantidad de energía proveniente del viento, que nuestro dispositivo puede tomar. Para el diseño de anemómetros, se debe considerar otro aspecto para el área del elemento sensor, como lo es la diferencia de presiones debido a una diferencia de altura respecto al punto más bajo del sensor y el punto más elevado de éste. El modelo para determinar la relación entre la velocidad mecánica del dispositivo y la velocidad del viento, se puede complicar demasiado. Es por eso, que los anemómetros de rotación, siempre tienen un área de barrido bastante pequeño para simplificar el análisis de construcción y funcionamiento del dispositivo.

### **2.3.1 Parámetros eólicos de medición**

Todo instrumento de medición eólica debe de cumplir con ciertos parámetros, dentro de ellos existe una nomenclatura de velocidades, que debe ser registrada por el fabricante en el manual del usuario, para la correcta utilización del dispositivo de medición. La nomenclatura de velocidades dadas por los fabricantes son:

- Velocidad indicada
- Velocidad calibrada
- Velocidad verdadera

### **2.3.1.1 Velocidad indicada**

La velocidad indicada (Indicated Airspeed, AIS), es la velocidad leída directamente del anemómetro. Las velocidades indicadas pueden variar con los cambios de altura, debido a la variación de presión del aire.

### **2.3.1.2 Velocidad calibrada**

La velocidad calibrada (Calibrated Airspeed, CAS) es la velocidad corregida al momento de la instalación. Está es conocida también como la velocidad de error cero. Cuando se calibra un instrumento en una cámara de viento con una velocidad bajo condiciones controladas, se toma que la diferencia entre la velocidad indicada y la velocidad calibrada es igual a cero.

### **2.3.1.3 Velocidad verdadera**

La velocidad verdadera (True Airspeed, TAS) es la velocidad indicada corregida por un factor dependiente de la densidad del viento y por otro factor dependiente de la altura de instalación del dispositivo.



## 3. MICROCONTROLADORES

### 3.1 Definición

Un microcontrolador es un circuito integrado utilizado para controlar el desempeño de un proceso. Un microcontrolador es un circuito integrado de una escala de integración muy elevada, lo que permite incluir dentro de él, todos los elementos de un controlador. Anteriormente, los controladores eran de tipo electromecánico, estaban contruidos a base de contactores y de relays. Estos controladores ocupaban demasiado espacio, tenían una baja confiabilidad por la cantidad de elementos y era muy difícil realizar una modificación al diseño.

Los microcontroladores han sido la base de diseño para los Controladores Lógicos Programables (PLC), que son el estándar en dispositivos para automatización de procesos. Un microcontrolador está constituido básicamente por:

- Procesador (Unidad central de procesos, CPU)
- Memoria de programa
- Memoria de datos
- Periféricos de entrada y salida
- Módulos periféricos
- Reloj

Un microcontrolador es el equivalente de una computadora en un solo circuito integrado, que incluye a todos los dispositivos periféricos que son conectados con el microprocesador. Los microcontroladores están orientados hacia el control de procesos. Algunas de las ventajas de utilizar microcontroladores en un sistema de control son:

- Bajo costo.
- Tamaño reducido.
- Se puede reprogramar en poco tiempo.
- Capacidad de almacenar datos por mucho tiempo, debido a sus memorias no volátiles.
- Puede guardar datos de calibración, sin necesidad de dispositivos de apoyo.

### **3.2 Fabricantes de microcontroladores**

En la actualidad el uso de microcontroladores está difundido para todos los campos de aplicación (automatización, instrumentos, telecomunicaciones, procesamiento digital de señales, etc.), lo que ha llevado a que una gran cantidad de empresas se dediquen a la fabricación de microcontroladores para satisfacer la demanda. Algunos de los principales fabricantes de microcontroladores en la actualidad son:

- Microchip
- Siemens
- Intel
- Motorola
- Texas Instruments

- Atmel
- Zilog

De los anteriores fabricantes, en la rama de microcontroladores y dispositivos periféricos para éstos, la compañía Microchip encabeza los mercados por las características que hacen a los microcontroladores PIC (Peripheral Interface Controller), los dispositivos más versátiles de utilizar para cualquier tipo de aplicación electrónica. Algunas de las razones por las cuales los PIC de Microchip, son los mejores controladores en el mercado son:

- Costos más bajos en comparación con otras empresas.
- Fáciles de programar.
- Amplia cantidad de información, software y soporte técnico, todo de forma gratuita, a través del fabricante por medio de su página en Internet ([www.microchip.com](http://www.microchip.com)).
- Fáciles de programar y de implementar en cualquier diseño.
- Poseen amplias características de acuerdo al modelo (tamaño de memoria, tipo de memoria, velocidad, ADC, comparadores analógicos, contadores, temporizadores, moduladores de ancho de pulso (PWM), interrupciones programables, tamaño físico, alimentación, controladores de memorias EEPROM, controladores de LCD, controladores de comunicación paralela, controladores de comunicación serial, controlador USB, entre otros.

### **3.2.1 Tipos de microcontroladores**

La clasificación de los microcontroladores puede volverse demasiado compleja si tomamos como base tamaño de memoria, tipo de memoria, velocidad, módulos periféricos, etc. Microchip ha dividido sus microcontroladores en tres gamas particulares, que toman como parámetro de clasificación el tamaño de las instrucciones de los microcontroladores PIC de registros de memoria de 8 bits. De acuerdo a esto, existen tres clasificaciones:

- Gama baja
- Gama media
- Gama alta

#### **3.2.1.1 Gama baja**

En esta gama se encuentran todos los PIC que cuentan con instrucciones de 12 bits de largo, poseen memoria flash, memoria RAM y memoria EEPROM. Dentro de esta clasificación se encuentran los PIC de las series PIC12CXX y PIC16CX5. Estos PICs cuentan hasta con 33 instrucciones.

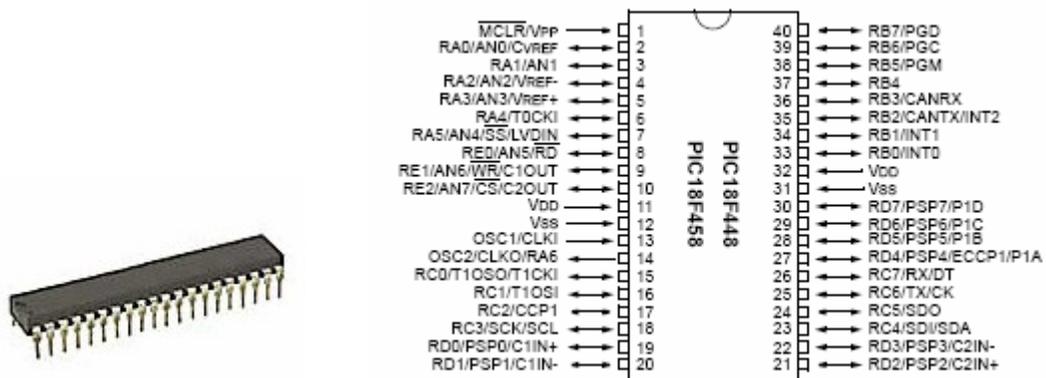
#### **3.2.1.2 Gama media**

Esta abarca todos los PICs de instrucciones de 14 bits de largo, también poseen memorias tipo flash, RAM y memoria tipo EEPROM. Estos PICs también cuentan con un juego de 35 instrucciones de 35 o más. Dentro de esta gama se encuentran los PIC16CXXX y los PIC16FXXX.

### 3.2.1.3 Gama alta

Esta es la última generación de microcontroladores construidos por Microchip, que poseen instrucciones de 16 bits de largo. Estos poseen una mayor capacidad de memoria que cualquier otra gama, además de un juego de 76 instrucciones. Los PIC17CXX, los PIC18CXXX y los PIC18FXXX se encuentran dentro de esta clasificación.

**Figura 6.** Microcontrolador PIC18F458. (www.microchip.com)



### 3.3 Arquitectura

Un microcontrolador PIC se han caracterizado por tener una estructura interna que permite incluir una gran cantidad de módulos y periféricos, que lo hacen un dispositivo de funcionamiento independiente. Algunas características de la arquitectura de los microcontroladores PIC son:

- Utilizan el modelo Harvard, en el cual, la memoria de datos y de programa se conectan al procesador por buses independientes, lo que permite que éste se comuniquen de forma simultánea con ambos.
- Se utiliza la segmentación de instrucciones para que mientras se ejecuta una instrucción, la siguiente ya se encuentra disponible en el procesador. Esto aumenta la velocidad del procesador por instrucción ejecutada.
- Todas las instrucciones poseen un código con la misma cantidad de bits.
- Utiliza instrucciones de código reducido (RISC), que permite una mayor velocidad de ejecución por instrucción, que si fuesen instrucciones de código complejo (CISC) o instrucciones de código específico (SISC), que requieren de mayor cantidad de ciclos de reloj para ejecutar una sola instrucción.
- Su arquitectura se basa en bancos de registro, es decir, mapeo de memoria. Todo módulo, entrada o salida del PIC, puede ser direccionado como si fuese una localidad de memoria.
- Todos los PICs poseen diferentes módulos especiales de acuerdo a la necesidad del proceso controlado (módulo ADC, módulo PWM, módulo USB, módulo LCD, etc.).

Por su tipo de arquitectura los microcontroladores se clasifican es:

- **Arquitectura cerrada:** estos microcontroladores poseen un procesador característico, un tamaño de memoria determinado, una cantidad de

instrucciones especificadas, un número específico de entradas / salidas y recursos auxiliares característicos de cada microcontrolador. Estos microcontroladores admiten muy pocas variaciones en su hardware. Estos son microcontroladores de tareas específicas. Los microcontroladores PIC poseen este tipo de arquitectura.

- **Arquitectura abierta:** son microcontroladores a los cuales se les puede ampliar la memoria, el número de entradas / salidas y agregar circuitos integrados externos. Estos microcontroladores tienen un funcionamiento similar al de un microprocesador, en el cuál, se pueden agregar los periféricos necesarios para la aplicación deseada.

La razón de utilizar una arquitectura de tipo cerrada en un microcontrolador, es la de poder utilizar un solo circuito integrado para resolver una aplicación. Esto disminuye los costos del controlador.

Los microcontroladores PIC, se dividen por su arquitectura en los siguientes elementos:

- Núcleo
- Periféricos
- Características especiales

### **3.3.1 Núcleo**

El núcleo de un microcontrolador está compuesto de todos aquellos elementos que hacen funcionar al dispositivo. Los elementos que conforman el núcleo de un microcontrolador PIC son:

- Procesador
- Unidad aritmética lógica
- Memoria
- Oscilador
- Operación de interrupción
- Instrucciones

Cada uno de estos elementos son partes fundamentales del núcleo, algunos son elementos físicos del microcontrolador, mientras que tan solo las instrucciones son elementos lógicos de operación.

### **3.3.1.1 Procesador**

Es el dispositivo encargado de direccionar la memoria de instrucciones, almacenar la instrucción a ejecutar, decodificar la instrucción y por último ejecutarla. El procesador es el que determina la capacidad de memoria a instalar (direcciones disponibles) y el tamaño de los registros de la memoria (bus de datos).

### **3.3.1.2 Unidad aritmética lógica**

La unidad aritmética lógica (ALU) es la encargada de efectuar todas las operaciones entre registros, ya sean operaciones de tipo aritmético o de tipo lógico. De acuerdo a la capacidad de operaciones a realizar por el ALU, así se puede determinar la cantidad de ciclos de reloj a utilizar por operación ejecutada.

### 3.3.1.3 Memoria

La memoria de un microcontrolador son registros digitales que almacenan el programa y los datos del usuario durante el funcionamiento normal de dispositivo, e incluso, pueden mantener estos registros, sin necesidad de estar energizado. Un microcontrolador se caracteriza por tener memorias dentro del mismo integrado, para todos sus procesos y carecen de medios de almacenamiento externo como los microprocesadores. Sin embargo, los microcontroladores pueden hacer uso de memorias externas, cuando el proceso no exige registrar una gran cantidad de datos.

Existen dos tipos fundamentales de memorias, la memoria no volátil (ROM) y la memoria volátil (RAM). Un PIC posee ambos tipos de memoria. En el caso de la memoria RAM, solo puede almacenar datos mientras el microcontrolador permanezca energizado y al suspenderse la alimentación, se pierden los datos almacenados. En el caso de las ROM, podemos almacenar datos, aún después de quitar la alimentación al microcontrolador. Las memorias no volátiles son importantes en el microcontrolador, para almacenar la memoria del programa y para almacenar datos de los procesos realizados.

Las memorias ROM disponibles en los microcontroladores son:

- **ROM con máscara:** los datos almacenados en esta memoria, se graban durante la fabricación del dispositivo. Es útil en el caso de producción a escala de grandes cantidades de microcontroladores para una misma aplicación.

- **OTP:** la memoria de programación única, permite al usuario grabar una sola vez en sus registros los datos deseados, utilizando circuitos sencillos, después de grabada, es imposible borrarla o escribir sobre los datos existentes.
- **EPROM:** la memoria ROM programable y borrable, puede ser grabada y borrada varias veces. Para programarla solo es necesario un circuito para direccionamiento de datos en las localidades deseadas. Para borrarla es necesaria la aplicación de luz ultravioleta, a través de una pequeña abertura que poseen los circuitos integrados para poder ser borrados.
- **EEPROM:** es una memoria ROM que se puede programar y borrar eléctricamente. Son ideales en aplicaciones donde deseamos almacenar tablar de valores o para registrar de forma permanente algunos datos de la aplicación del microcontrolador. Esta memoria posee un número finito de ciclos de lectura y escritura.
- **FLASH:** su funcionamiento es similar a la memoria EEPROM, pero posee las características de ser más veloz y de poder una mayor densidad por encapsulado, es decir que puede tener mayor capacidad que un encapsulado de cualquier otra memoria. Las memorias FLASH tienen aproximadamente un ciclo de vida de hasta 1 millón de ciclos de lectura / escritura.

Las memorias más utilizadas por un microcontrolador son las memorias OTP, FLASH, EEPROM y RAM. La memoria de un microcontrolador se divide en tres secciones: memoria de programa, RAM de datos y EEPROM de datos.

### **3.3.1.3.1 Memoria de programa**

La memoria de programa es aquella en la cual se almacenan las instrucciones del programa a utilizar, generalmente son del tipo FLASH y en algunos casos de aplicaciones industriales son del tipo OTP. La memoria de programa a su vez se divide en la pila (que almacena la dirección de una instrucción durante una interrupción o una llamada), memoria de usuario (donde se instala el programa) y los vectores de interrupción (son localidades de memoria donde se almacenan las rutinas de interrupción o su dirección). Estas memorias pueden abarcar tamaños de 1 kilo Byte (PIC de la gama baja) hasta 128 kilo Bytes (PIC de gama alta).

### **3.3.1.3.2 RAM de datos**

La memoria de programa se divide regularmente en bloques, que pueden ir desde algunos Bytes hasta algunos cientos de kilo Bytes. La memoria de datos es construida con una memoria RAM estática. La memoria de datos permite almacenar las variables creadas por el usuario en el programa, pero además posee los registros de memoria propios de cada PIC, donde se encuentran registros tan elementales como el de estado del ALU así como registros de los puertos y su configuración. La memoria de programa esta dividida en bloques de Bytes. En el caso de un PIC18F452 posee 16 bancos de 256 Bytes cada uno.

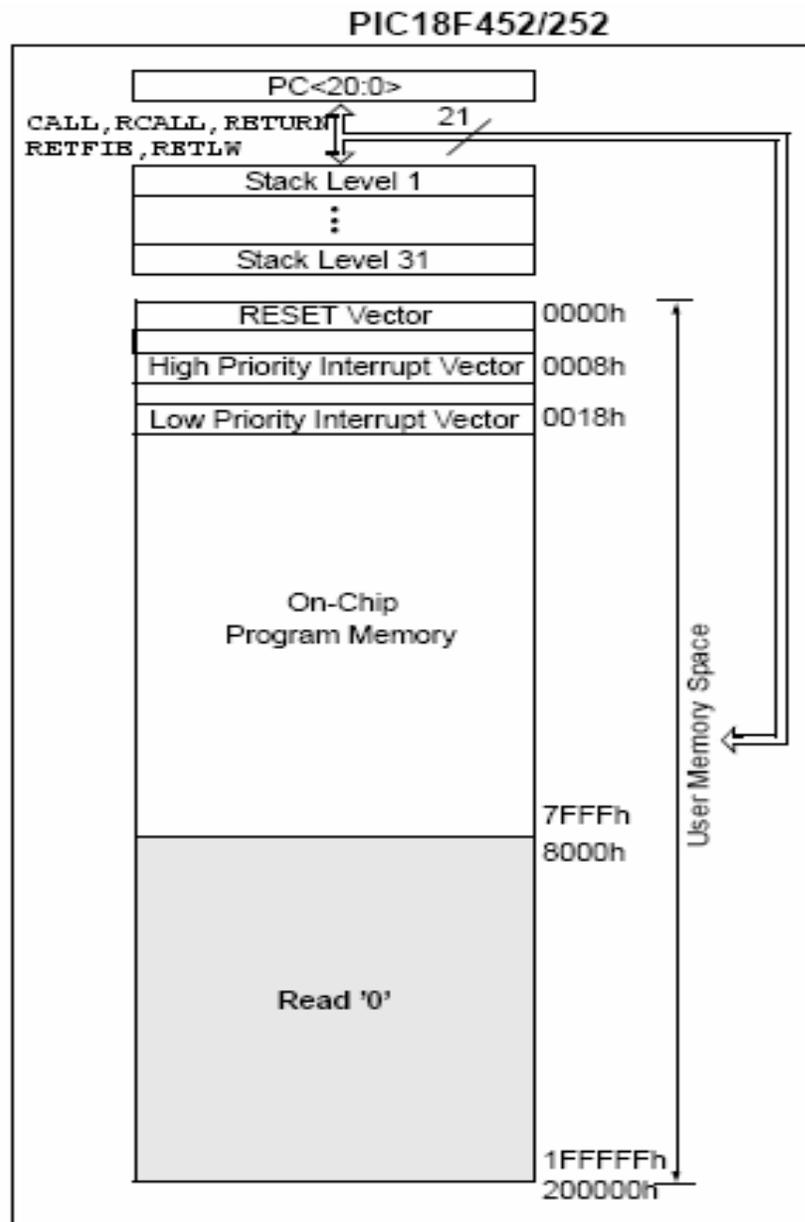
### **3.3.1.3.3 Memoria EEPROM**

Un PIC posee otra memoria para almacenar datos del usuario o tablas de valores para ser utilizadas por el programa durante su ejecución para comparar valores. La memoria EEPROM, posee la ventaja de ser una memoria no volátil, es decir que si el microcontrolador se reinicia o deja de ser energizado, los datos almacenados en la memoria EEPROM se mantendrán guardados. El único inconveniente de esta memoria, es que para los microcontroladores, se utilizan memorias integradas de hasta 1 kilo Byte. A pesar de eso, se puede hacer uso de las memorias EEPROM seriales y paralelas fabricadas por Microchip para aplicaciones donde sea necesario registrar datos o poseer tablar de datos muy grandes, y que son totalmente compatibles con los PIC, incluso algunos poseen módulos integrados para controlar su funcionamiento. Estas memorias externar pueden llegar hasta valores de 256 kilo Bytes.

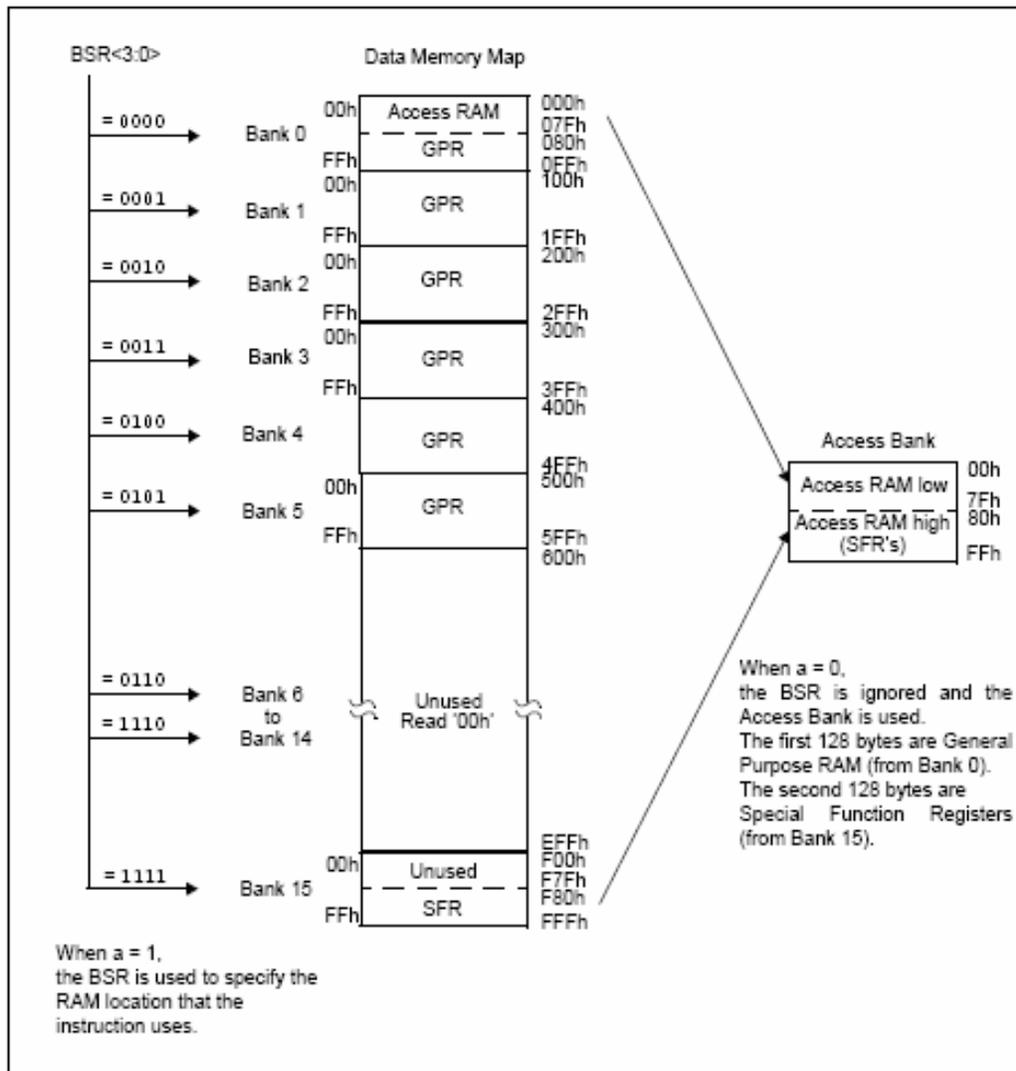
### **3.3.1.4 Oscilador**

Los microcontroladores poseen osciladores internos y es posible conectar un oscilador externo a éste. La función principal de un oscilador (reloj del sistema) es la de enviar los impulsos que servirán para sincronizar la ejecución de cada una de las instrucciones del programa. El oscilador es el que determina la velocidad de procesamiento del sistema, cada microcontrolador posee especificaciones de frecuencia máxima del oscilador con el cual puede ser alimentado. El consumo de un microcontrolador es directamente proporcional a la velocidad de ejecución de sus instrucciones.

**Figura 7.** Memoria de programa de un PIC18F452 ([www.microchip.com](http://www.microchip.com))



**Figura 8.** Memoria de datos del PIC18F452. ([www.microchip.com](http://www.microchip.com))



### **3.3.1.5 Operación de interrupción**

Los microcontroladores pueden ser programados para realizar ciertas rutinas en caso de presentarse algunas situaciones. Se le llama interrupción, al proceso de detener la ejecución normal de instrucciones y ejecutar una rutina determinada, de acuerdo al tipo de interrupción activada. Al finalizar la rutina de interrupción, el programa vuelve a la instrucción que iba a ser ejecutada en el momento de la interrupción.

Las interrupciones en un microcontrolador pueden ser de hardware (bajo voltaje de alimentación, flanco descendente, etc.) y de software (contadores, temporizadores, perro guardián, operaciones invalidas, etc.). El PIC cuenta con varias rutinas de interrupción que pueden ser habilitadas a medida que el usuario las necesite. De igual forma, el usuario puede programar sus propias rutinas de interrupción.

### **3.3.1.6 Instrucciones**

Las instrucciones son códigos binarios, que cuando son decodificadas por el procesador, dan como resultado la ejecución de una acción. Las instrucciones pueden ser de 12, 14 o 16 bits, de acuerdo a la gama utilizada. Las instrucciones utilizadas por los microcontroladores PIC, son el juego de instrucciones de código reducido (RISC), ya que éstas pueden ser ejecutadas por el procesador en un solo ciclo, o en algunos casos dos ciclos. Un ciclo de instrucción es equivalente a cuatro ciclos del oscilador que alimenta al microcontrolador.

Las instrucciones del microcontrolador son ortogonales, es decir que cualquier instrucción puede utilizar a cualquier registro como fuente o como destino, a excepción de los registros sólo de lectura del microcontrolador.

Las instrucciones de un microcontrolador son en lenguaje Assembler, lo que facilita su entendimiento y fácil uso. Las instrucciones en un microcontrolador se clasifican de la siguiente forma:

- Instrucciones orientadas a Bytes, que permiten utilizar registros de la memoria de datos para realizar operaciones de tipo lógico y aritmético.
- Instrucciones orientadas a bits, permiten utilizar los bits de un registro para una operación lógica.
- Instrucciones de constantes, estas nos permiten almacenar cualquier código en cualquier registro de la memoria de datos.
- Instrucciones de control, son todas aquellas que permiten el control de flujo lógico del programa. Nos permiten el uso de condicionales, llamadas e incluso saltos de posición dentro de la memoria de programa.
- Instrucciones de memoria, estas instrucciones nos permiten la escritura y lectura en memorias utilizadas para el registro de datos.

### 3.3.2 Periféricos

Los periféricos son todas aquellas características que diferencian a un microcontrolador de un microprocesador. Los periféricos son los dispositivos que realizan una interfaz entre el procesador y el mundo real. Estos nos sirven para llevar información del proceso o ejecutar acciones. Algunos de los periféricos disponibles en los microcontroladores PIC son:

- Entradas y salidas de propósito general.
- Temporizadores de 8 y 16 bits.
- Módulo de captura, comparación y modulación de ancho de pulso (PWM).
- Puerto serial síncrono.
- Módulo USB.
- Puerto serial síncrono maestro y esclavo.
- Receptor transmisor serial asíncrono síncrono.
- Voltajes de referencia.
- Comparador analógico.
- Convertidor de analógico a digital de 8 y 10 bits.
- Controladores de Pantalla de cristal líquido. (LCD)
- Puerto paralelo esclavo.

Dentro de los periféricos más importantes se encuentran las entradas y salidas de propósito general, que pueden ser configuradas como entradas o salidas digitales. Además también existen puertos configurables como entradas de tipo analógico.

### **3.3.3 Características especiales**

Son todas aquellas características de la arquitectura, que diferencian a los microcontroladores PIC de cualquier otro fabricado por otra compañía. Estas características disminuyen el costo del dispositivo, aumenta su rentabilidad y aumenta la flexibilidad de diseño del dispositivo. Algunas de las características especiales de los microcontroladores PIC son:

- Posee registros para configurar el dispositivo.
- Terminal para reiniciar el procesador.
- Reinicio en caso de voltajes bajos de alimentación.
- Modo de bajo consumo.
- Oscilador interno.
- Programación serial del dispositivo.

### **3.4 Herramientas de programación**

Los microcontroladores PIC poseen una amplia gama de herramientas para su programación, que pueden ser adquiridas gratuitamente en el sitio de Microchip ([www.microchip.com](http://www.microchip.com)). Dentro de las últimas herramientas desarrolladas para la programación de microcontroladores se encuentra el programa para desarrolladores y programadores de PICs MPLAB IDE versión 6.5.

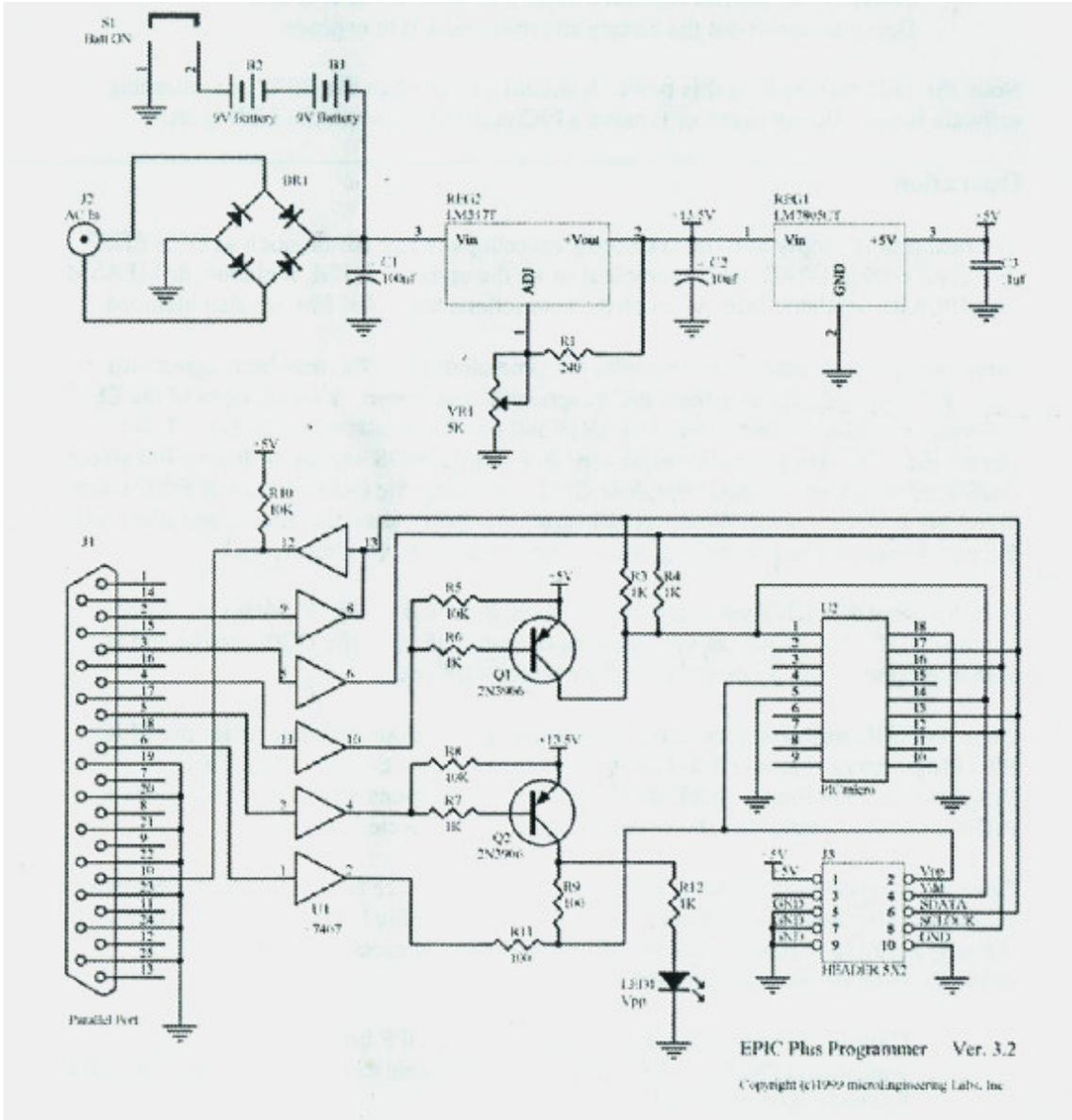
El MPLAB IDE: es un ambiente integrado de desarrollo que cuenta con un grupo de programas para utilizar un PIC. Dentro de las herramientas que posee el MPLAB IDE versión 6.5 se encuentran:

- **MPLAB Editor:** nos permite escribir, editar y crear archivos tipos Assembler y hexadecimal del código fuente del programa que deseamos instalar en un PIC.
- **MPASM Assembler:** nos permite depurar y compilar los códigos generados por el editor. Este programa es configurado de acuerdo al tipo de microcontrolador a utilizar.
- **MPLAB SIM:** es un programa que nos permite simular el funcionamiento del microcontrolador. Nos permite ver el mapa de memoria de programa, la memoria de datos y la memoria EEPROM. Además podemos ver el estado de cada uno de los registros del sistema, para así evaluar el desempeño del programa elabora y depurarlo.
- **MPLAB ICD 2 (2000 & 4000):** es un emulador que nos permite evaluar el funcionamiento del programa grabado en el PIC, por medio de un circuito de prueba que se conecta serialmente a la computadora. El PIC se coloca en este circuito para emular su comportamiento de acuerdo al cambio en sus entradas.
- **Microchip PRO MATE II:** este es una herramienta desarrollada por Microchip para programar, verificar programas, leer y borrar dispositivos. Se conectado de forma serial al PIC, y es controlado por el puerto serial de la computadora.
- **Visual Initializer:** es una interfase gráfica que nos permite observar el comportamiento del circuito integrado, este programa se utiliza en

conjunto con el MPLAB IDE. Nos permite configurar el dispositivo, agregar módulos (comparadores, contadores, entradas, etc.) y observar el comportamiento del microcontrolador ante los cambios en las entradas.

Uno de los circuitos utilizados para programar PICs, es el proporcionado por Micro Engineering Labs, el cuál es totalmente compatible con el programa Microchip PRO MATE II, después de realizar ciertas modificaciones en la configuración de la herramienta. Ya que el programador de MPLAB utiliza el puerto serial, debemos de configurar el programador para utilizar el puerto paralelo. El programa proporcionado para este circuito es el EPIC de Micro Labs.

**Figura 9.** Circuito del programador de PIC EPIC de Micro Engineering Labs.  
([www.melabs.com](http://www.melabs.com))



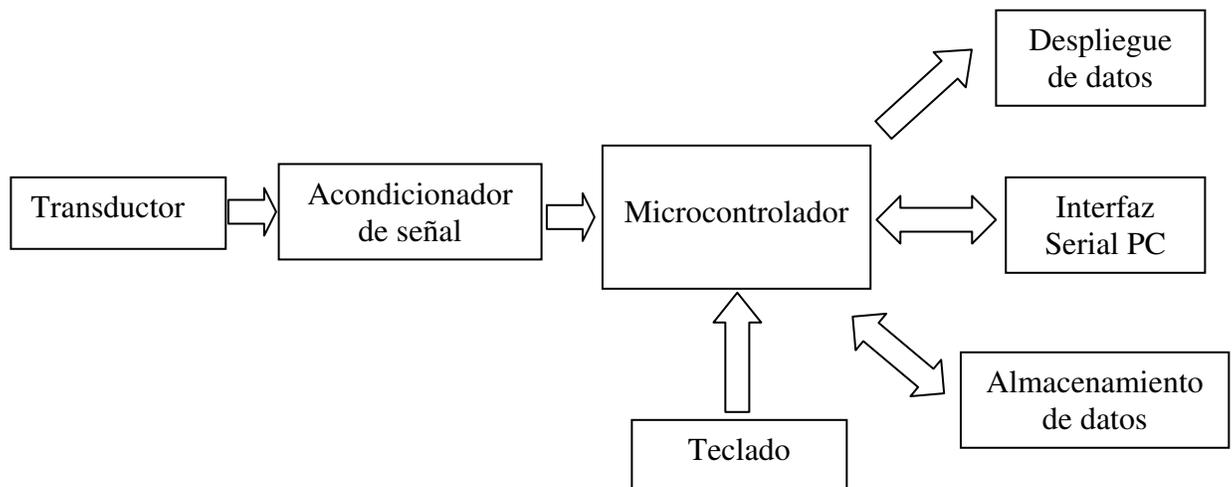


## 4. DISEÑO DE UN ANEMOGRAFO ELECTRONICO

### 4.1 Diagrama de Bloques

El dispositivo electrónico se ha dividido en los siguientes bloques para su etapa de diseño:

**Figura 10.** Diagrama de bloques que conforman el anemógrafo



### 4.2 Transductor

En la etapa de documentación bibliográfica, se ha determinado la utilización de un anemógrafo de copas para diseñar el dispositivo transductor mecánico.

El transductor del anemógrafo, será el dispositivo que estará en contacto con el aire y de acuerdo a la velocidad de éste, así será la velocidad con que girará el dispositivo de copas.

El transductor permitirá obtener una señal eléctrica proporcional a la velocidad mecánica del viento. Por medio de una etapa acondicionadora, podremos ajustar el nivel de la señal de salida del transductor para poder luego digitalizarla y poder analizarla, para determinar la velocidad promedio del viento.

#### **4.2.1 Diseño mecánico**

Para el diseño mecánico, nos basaremos en los transductores mecánicos utilizados por los anemómetros de copas, por su facilidad de acoplar a éste un transductor mecánico eléctrico.

El elemento mecánico consistirá en tres semiesferas metálicas separadas entre si por  $120^\circ$  sobre un mismo eje del mismo radio para cada una. El viento incidente en las semiesferas metálicas provocarán un torque que será proporcional a la velocidad del viento incidente. Además, este torque provocará que la hélice formada por las tres semiesferas gire a una velocidad angular proporcional a la velocidad del viento incidente. Este transductor mecánico nos permitirá tener como entrada una velocidad lineal (velocidad del viento) y como salida una velocidad angular (velocidad de giro de la hélice).

El Dr. Gary Johnson en su estudio 'Wind Energy Systems' publicado en noviembre de 2001, analiza el funcionamiento mecánico del transductor mecánico del anemómetro de copas.

El plantea, que el torque resultante en un sistema mecánico compuesto de tres semiesferas montadas sobre un mismo eje y espaciadas por 120°, será el resultado de el torque debido a la velocidad del viento menos el torque debido a la fricción mecánica de los cojinetes o del eje del dispositivo mecánico y el torque por la fricción del viento.

$$T_R = I \alpha = I \frac{\partial \omega}{\partial t} = T_V - T_{FV} - T_M \quad (\text{Ec. 5})$$

Donde

$I$  = inercia del sistema mecánico

$\alpha$  = aceleración angular del sistema mecánico

$\omega$  = velocidad angular del sistema mecánico

$T_V$  = torque debido al viento

$T_{FV}$  = torque debido a la fricción del viento

$T_M$  = torque debido a la fricción mecánica del sistema

Cada uno de los torques es definido por el Dr. Johnson de la siguiente manera:

$$T_V = K_1 \omega_o u \quad (\text{Ec. 6})$$

$$T_{FV} = K_1 u_o \omega \quad (\text{Ec. 7})$$

$$T_M = K_M \omega \quad (\text{Ec. 8})$$

Donde

$\omega_0$  = velocidad angular inicial del sistema mecánico.

$u_0$  = velocidad del viento inicial.

$\omega$  = velocidad angular del sistema mecánico.

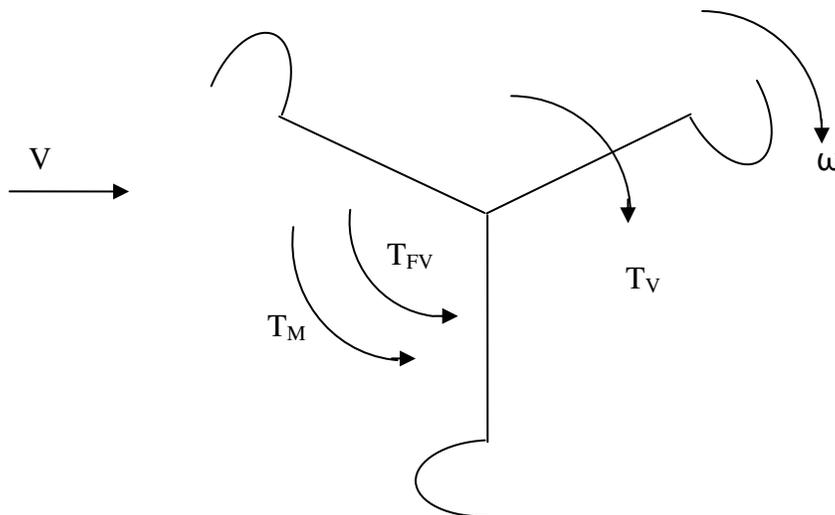
$u$  = velocidad del viento.

$K_1$  = constante que depende de las dimensiones de las copas.

$K_M$  = constante dependiente de las dimensiones del sistema mecánico.

Para determinar la solución del sistema que nos permitirá determinar la relación entre la velocidad del viento y la velocidad angular del sistema mecánico, basándose en aplicar un escalón unitario de cambio en la velocidad del viento, para determinar la respuesta del sistema en base al cambio en la velocidad angular.

**Figura 11.** Torques que existen en un anemómetro de copas.



La velocidad angular en función del cambio en la velocidad del viento y en la velocidad angular inicial del sistema se obtiene de resolver la ecuación diferencial No. 6.

$$\omega = \frac{\omega_o}{K_M + K_1 * u_o} (K_1 * u_1 + [K_M + K_1(u_o - u_1)] * e^{-t/\tau}) \text{ rad/s} \quad (\text{Ec. 9})$$

$$\tau = \frac{I}{K_M + K_1 * u_o} \quad (\text{Ec. 10})$$

Donde

$\tau$  = constante de tiempo del sistema mecánico

$u_1$  = velocidad final en intervalo de prueba

De acuerdo al análisis realizado por el Dr. Johnson, el comportamiento del sistema mecánico de hélices de copas tiene un comportamiento sobreamortiguado, igual que la curva de respuesta del voltaje durante la carga de un capacitor.

Experimentalmente se ha demostrado que torque por fricción debido al sistema mecánico, por la acción mínima en los cojinetes, es muy pequeña comparada respecto a los torque aerodinámicos, así que puede ser despreciada en el análisis. La ecuación No. 9 puede ser reducida a:

$$\omega = \omega_o * \frac{u_1}{u_o} + (u_o - u_1) * \frac{\omega_o}{u_o} e^{-t/\tau} \quad (\text{Ec. 11})$$

$$\tau = \frac{I}{K_1 * u_o} \quad (\text{Ec. 12})$$

Recordando las relaciones básicas de velocidad tangencial es igual a el radio de giro por la velocidad angular, podemos determinar que para el sistema mecánico del anemómetro, la relación entre la velocidad del viento y la velocidad angular de la hélice de copas es aproximadamente proporcional.

$$U = K_i * \varpi \quad (\text{Ec. 13})$$

Donde

$K_i$  = constante aproximadamente igual al radio de rotación de las copas montadas sobre la hélice.

Sustituyendo la ecuación No. 11 en la ecuación No. 13, obtenemos la relación existente entre la velocidad del viento y la velocidad angular de la hélice de copas:

$$U = K_i * \varpi = K_i * \left[ \omega_o * \frac{u_1}{u_o} + (u_o - u_1) * \frac{\omega_o}{u_o} e^{-t/\tau} \right] \quad (\text{Ec. 14})$$

Para determinar el valor de la velocidad del viento, la ecuación No. 14 se puede resolver numéricamente, obteniendo el valor para cada una de las constantes mecánicas y de tiempos de respuesta, colocando la hélice de copas dentro de una cámara de viento, debido a que la respuesta del sistema mecánico está en función de sus dimensiones físicas.

En el microcontrolador, utilizaremos el valor de la constante  $K_i$ , para poder obtener una relación lineal de la velocidad del viento y la velocidad angular del transductor mecánico.

#### **4.2.2 Diseño eléctrico**

El transductor mecánico será acoplado por su eje a un tacómetro para obtener una señal eléctrica de salida proporcional a la velocidad del viento.

Un tacómetro es un generador eléctrico, que al estar girando su eje da en su devanado del estator o de rotor (de acuerdo al tipo de tacómetro) un voltaje que es proporcional a la velocidad de giro del eje, de acuerdo a la ley de inducción de Faraday.

El tacómetro a utilizar en el diseño, será un generador de DC, debido a su gran disponibilidad en el mercado, además que existe una amplia variedad de información de las características eléctricas y mecánicas de éstos, proporcionados por los fabricantes en hojas técnicas.

El generador de corriente continua a considerar es uno de imán permanente en el estator y de devanado de salida en el rotor, con anillos rozantes en la salida. El voltaje de salida para un generador de este tipo depende de varios factores:

$$E = K_T \omega \quad (\text{Ec. 15})$$

Donde

$E$  = voltaje inducido (voltios)

$\omega$  = velocidad angular del rotor (rad/s)

$K_T$  = constante de construcción del tacómetro DC (voltios \* s)

La constante  $K_T$  es proporcionada por el fabricante en la hoja técnica de datos y depende del tamaño físico del motor, sus características geométricas, el número de vueltas del inducido, la densidad de campo magnético (si fuera con alimentación en el devanado de campo, consideraríamos la corriente de campo en lugar de directamente la densidad de campo magnético, al igual que el efecto de armadura) y la inercia del rotor.

Consideraremos la ecuación No. 13 y la ecuación No. 15, para determinar una relación directa entre la velocidad del viento y el voltaje de salida del transductor:

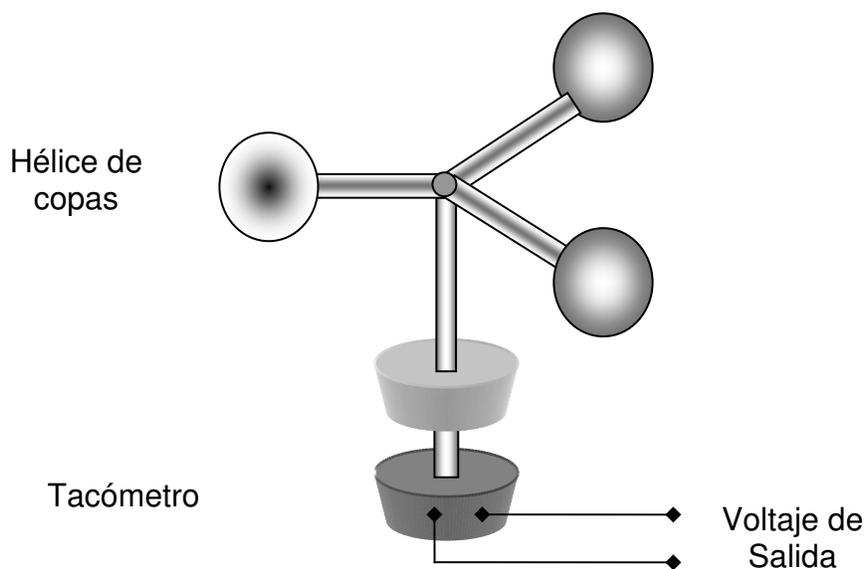
$$U = \frac{K_i}{K_T} * E \quad (\text{Ec. 16})$$

$$K_P = \frac{K_i}{K_T} \quad (\text{Ec. 17})$$

La constante  $K_P$  es la constante de proporcionalidad, que se determinará de acuerdo a las características que posean tanto la parte mecánica del

transductor (hélice de copas) y el transductor mecánico eléctrico (tacómetro). Esto permite una mayor versatilidad dentro del dispositivo, ya que esta constante, luego de ser determinada, podrá ser calibrada en el microcontrolador, por los cambios a sus características físicas, a los que están sometidos todos los dispositivos mecánicos durante su funcionamiento.

**Figura 12.** Diagrama físico del transductor del anemógrafo.



Se evaluó la opción de utilizar un encoder, pero debido al consumo continuo de energía por los diodos emisores de luz en la longitud del infrarrojo, disminuiría el tiempo de funcionamiento del dispositivo en mayor proporción que el transductor mecánico.

### **4.3 Acondicionador de señal**

El dispositivo para acondicionar señales será el encargado de aplicar un factor de escala a la señal de salida del transductor, esto será necesario en casos donde la velocidad del viento este distribuida en un rango de voltajes de salida muy pequeños, para lo cual se aplicará un factor de amplificación para aumentar la resolución de las señales de salida que serán luego digitalizadas. En algunos casos será necesario utilizar un factor de atenuación cuando los valores nominales de voltajes de salida sean elevados respecto al rango de voltaje a utilizar por el convertidor de analógico a digital.

Además debemos de considerar que eventualmente, existe la posibilidad de tener voltajes de salida elevados respecto a los valores nominales para los cuales se ha calibrado la etapa amplificadora, entonces será necesario utilizar un circuito limitador de picos de voltaje, para evitar daños en las siguientes etapas del diseño.

La etapa de acondicionamiento de voltaje estará dividida en dos partes:

- Etapa limitadora de sobrevoltajes.
- Etapa atenuadora / amplificadora.

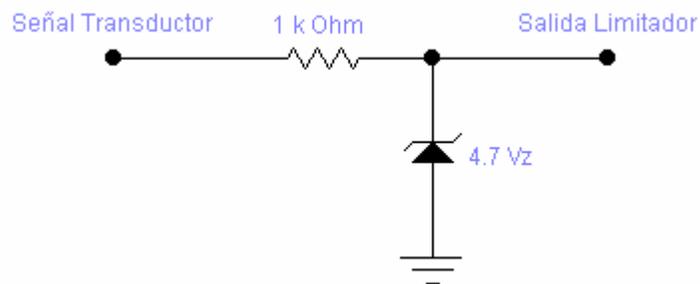
#### **4.3.1 Limitador de sobrevoltajes**

Esta etapa se encarga de limitar el voltaje máximo de salida, de la etapa proveniente del transductor de voltaje, hacia la etapa atenuadora / amplificadora. Para limitar el voltaje de salida a un nivel máximo, utilizaremos un dispositivo sencillo, un diodo tipo Zener. El circuito consistirá en una resistencia de 1 kOhm para limitar la corriente en que pasará por el Zener

cuando se encuentre en conducción y como divisor de voltaje, esto solamente será necesario cuando exista un sobrevoltaje. El Zener a utilizar tendrá un voltaje Zener de 4.7 Voltios, ya que durante la práctica, y debido a el proceso de fabricación, estos diodos están respondiendo a un voltaje Zener un poco superior que su voltaje nominal, durante varios proyectos, se obtuvieron mejor características en regulación de voltajes de 5 Voltios con diodos Zener de 4.7 Voltios que con diodos Zener de 5 voltios e incluso de 5.1 Voltios.

El funcionamiento de este circuito es bastante sencillo, mientras el voltaje de entrada sea menor a 5 voltios, el Zener no estará conduciendo, pero cuando existan voltajes mayores a 5 voltios, el Zener entrara en conducción, y mantendrá su caída de voltaje constante en 5 voltios, a pesar de existir voltajes mayores en la entrada. Cuando el voltaje caiga de nuevo a valores menores que 5 voltios, el Zener saldrá de conducción y permitirá nuevamente que el voltaje de salida sea igual al voltaje de la entrada.

**Figura 13.** Diagrama esquemático del limitador de sobrevoltaje.



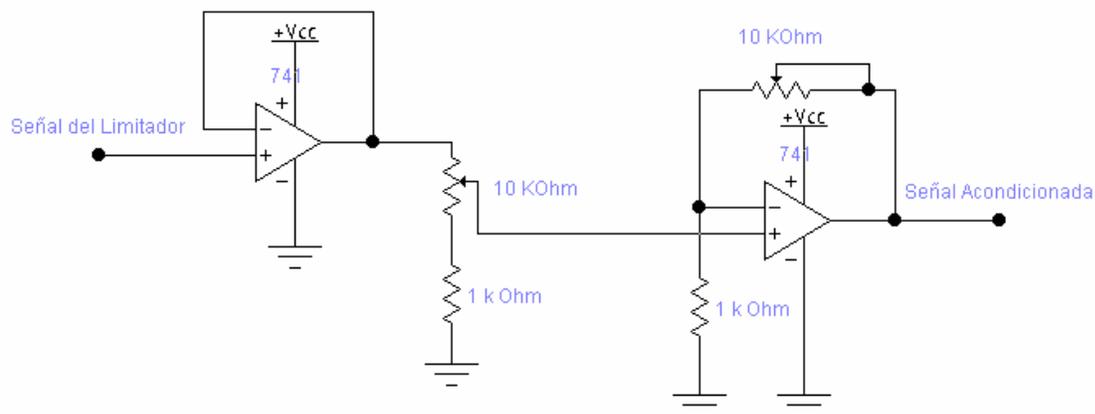
### 4.3.2 Etapa atenuadora / amplificadora

Esta etapa esta constituida por dos elementos, una etapa para acoplar voltaje y otra para amplificaci3n de la se1al.

La etapa acopladora de voltaje, se utilizar1 para que la etapa amplificadora no cargue al tac3metro durante la operaci3n normal del dispositivo y esto altere el factor  $K_P$  determinado en la etapa de calibraci3n del transductor, que se utilizar1 en el microcontrolador para procesamiento de la se1al de entrada. Para esto se utilizar1 un amplificador operacional configurado como seguidor de voltaje.

La salida del seguidor de voltaje, ser1 conectada a la etapa atenuadora / amplificadora. La etapa atenuadora / amplificadora estar1 constituida por un divisor de voltaje que nos permitir1 controlar el factor de atenuaci3n de la se1al, mientras que el factor de amplificaci3n ser1 controlado por la ganancia en un circuito amplificador no inversor con un amplificador operacional.

**Figura 14.** Diagrama esquem1tico del circuito de la etapa atenuadora / amplificadora.



El factor de amplificación puede ser controlador por el potenciómetro que se encuentra en la rama de retroalimentación del amplificador no inversor, éste puede llegar a ser hasta 10 veces el nivel original de la señal y su mínimo nivel de operación es el de una amplificación por un factor unitario.

El nivel de atenuación puede ser controlado hasta en una relación de 11 a 1, por el divisor de voltaje colocado a la salida del seguidor de voltaje, de la misma forma podemos anular el factor de atenuación de la señal colocando el cursor del potenciómetro en su posición de menor resistencia, es decir conectado directamente con la salida del amplificador operacional de la etapa anterior. Las resistencias, se han elegido de valores de 1 kOhm y de 10 kOhm para tener un bajo consumo de corriente, evitando así, cargar al tacómetro en cualquier etapa posterior y tener que recalibrar el sistema por cambio en la respuesta en el transductor. Los amplificadores operaciones serán alimentados con una referencia de voltaje positivo de 5 Voltios y una referencia negativa de 0 Voltios, utilizaremos un convertidor de analógico a digital que puede tener una amplitud de voltaje máxima de 5 Voltios, por estar integrado en el microcontrolador.

La salida del dispositivo acondicionador de señales estará conectada a un puerto analógico del microcontrolador, para poder procesar posteriormente, los datos de forma digital y además facilitar su almacenamiento. Se ha seleccionado la entrada analógica A0 del microcontrolador para el muestreo de la señal proveniente de ésta etapa.

#### **4.4 Dispositivo de almacenamiento de datos**

El dispositivo de almacenamiento de datos estará constituido por una memoria externa al microcontrolador, debido a las limitaciones de almacenamiento en memoria física no volátil que existen en los microcontroladores.

Se utilizará una memoria de tipo EEPROM, para que en algún momento, si no es renovada la fuente de alimentación del instrumento, podamos mantener los datos de mediciones realizadas. Las EEPROM permiten el almacenamiento de datos aún careciendo de fuente de alimentación, además son de suma ayuda, para contingencias como daños en el equipo, o si el equipo se ha reiniciado, poder guardar parámetros de configuración de este.

Dentro de las diferentes clasificaciones de memorias tipo EEPROM, podemos generalizar por el tipo de transmisión de su información: de forma serial o de forma paralela. La memoria paralela es bastante sencilla de implementar, pero tiene el inconveniente de necesitar demasiados pines de conexión y en algunos casos de dispositivos adicionales para multiplexación de los buses de comunicación. Las memorias de tipo serial, son de tamaño reducido en comparación con la memoria de tipo paralelo, pero su circuito de control es mucho más complejo y costoso. Debido a que los microcontroladores PIC de Microchip cuenta con módulos para conexión serial con dispositivos periféricos, el diseño de la interfaz serial no representará problema alguno para el diseño ni un inconveniente en relación a costos adicionales.

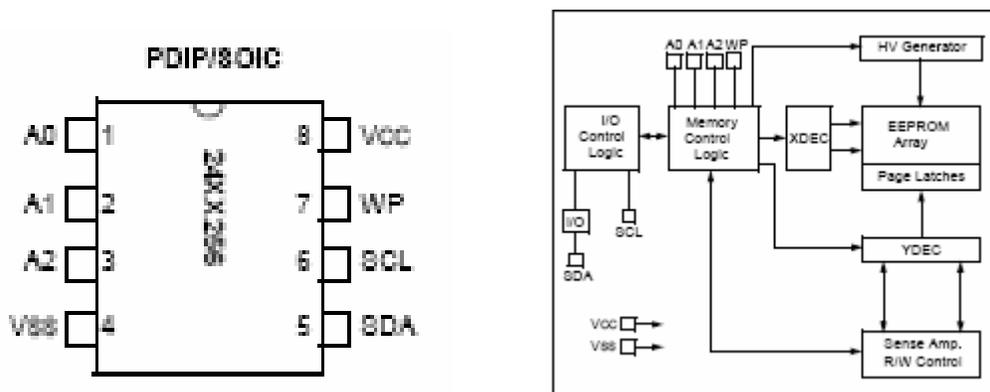
Las memorias EEPROM seriales a utilizar serán las 24AA512 de Microchip, por su costo bajo, su alta capacidad de almacenamiento, su facilidad de utilización y ante todo, que los microcontroladores PIC poseen módulos

especializados para la comunicación con periféricos seriales, que permitirán utilizar esta memoria únicamente configurando algunos registros.

La memoria EEPROM serial 24AA512 posee las siguientes características:

- 512 kbits, organizados en bloques de 64 bytes.
- Bajo consumo (3mA para escritura y 400  $\mu$  A para lectura)
- Solamente dos cables de conexión para comunicación (Datos y reloj)
- Se pueden conectar hasta 8 dispositivos en cascada.
- Ciclo de escritura / lectura máximo de 5ms.
- Entradas de dato y reloj Schmitt Trigger
- Hasta 1 millón de ciclos de escritura / lectura.
- Protección contra descargas electrostáticas mayores a 4000 V.
- Retención de datos mayor a 200 años.
- Temperatura de operación desde -40° C hasta +125° C.

**Figura 15.** Memoria EEPROM serial, encapsulado y diagrama interno.



#### 4.4.1 Funcionamiento de la memoria EEPROM

La memoria EEPROM 24AA512 tiene un funcionamiento muy sencillo, ya que solamente posee 8 terminales de conexión (Figura No. 15). Cada uno de las terminales posee una función específica:

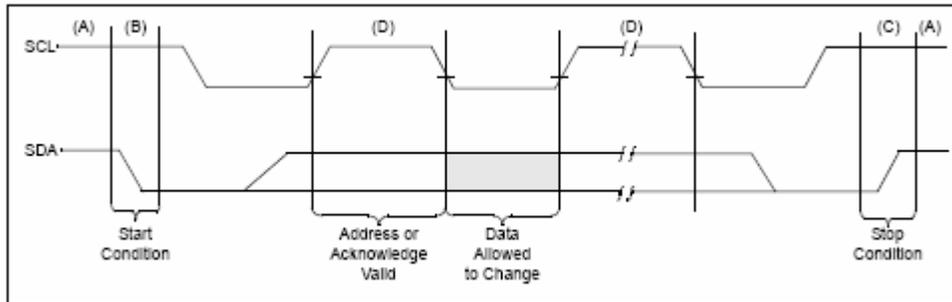
- **A2, A1 y A0:** estas terminales son entradas configuradas por el diseñador para asignar una dirección a cada una de las memorias seriales que se utilicen. Por ser tres terminales, nos permite únicamente direccionar hasta 8 dispositivos, si deseamos conectar más memorias, será necesario utilizar un multiplexor. La configuración de estas terminales se realiza conectando cada una de éstas a tierra para un '0' lógico y a voltaje para un '1' lógico.
- **Vss:** es la terminal de referencia negativa de voltaje, la cual se conecta a 0 voltios. Es llamada también tierra.
- **Vcc:** es la terminal de referencia positiva de voltaje, y puede ser conectada a voltajes desde 2.5 Voltios hasta 5.5 Voltios.
- **WP:** terminal de protección contra escritura (Write Protect Input), esta terminal nos permite habilitar o deshabilitar la escritura en la memoria, si está activada, no impide la lectura del dispositivo.
- **SDA:** datos seriales (Serial Data), es la terminal de comunicación entre la memoria y el dispositivo de control. Es bidireccional.

- **SCL:** Reloj serial (Serial Clock) es la terminal que sincroniza la transferencia de datos de la memoria con otros dispositivos.

La memoria EEPROM serial utiliza varias señales para poder comunicarse con otros dispositivos:

- La señal de reloj debe ser la misma tanto para la memoria como para el dispositivo que controla a ésta, para poder sincronizar la transferencia de comunicación.
- La Señal de Inicio de Transferencia de datos (START) es un flanco descendente que debe ocurrir durante el semiciclo positivo de la señal de reloj, sirve para indicar el inicio de la comunicación.
- Señal de Finalización de Transferencia de datos (STOP) es un flanco ascendente que debe ocurrir durante el semiciclo positivo de la señal de reloj, sirve para indicar el final de la comunicación.
- Señal de reconocimiento (Acknowledge) se da cuando se han enviado un byte, el transmisor envía un '0' lógico por SDA y para la recepción, es la memoria quien coloca la terminal SDA a '1' lógico.
- Señal de datos (DATA) es cada uno de los BIT que envía o recibe la memoria y son reconocidos como validos solamente en el semiciclo positivo de la señal de reloj. Estos pueden cambiar únicamente durante el semiciclo negativo del reloj, de lo contrario, un cambio en el otro semiciclo, causaría que se cambie el dato tomado como valido.

**Figura 16.** Secuencia de transferencia de datos en el bus serial. ([www.microchip.com](http://www.microchip.com))



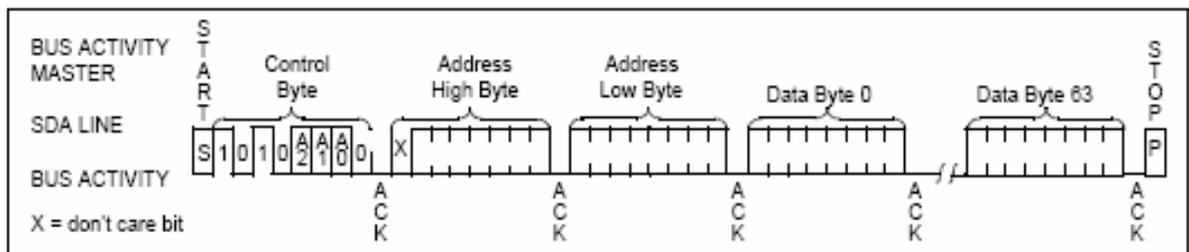
La memoria serial posee un protocolo de comunicación establecido para intercambiar información con otros dispositivos, el direccionamiento se basa en el envío de un byte de control para direccionar el dispositivo de memoria a utilizar y seleccionar un ciclo de lectura o escritura. Se envía inicialmente un byte de control, cuyos 4 bits más significativos es establecen con el código binario '1010'. Los tres siguientes bits (b3 al b1) constituyen los bits que permiten direccionar la memoria EEPROM a utilizar. Y el bit menos significativo indica si el ciclo es de lectura o escritura (1 es lectura y 0 escritura).

**Figura 17.** Formato del byte de control. ([www.microchip.com](http://www.microchip.com))



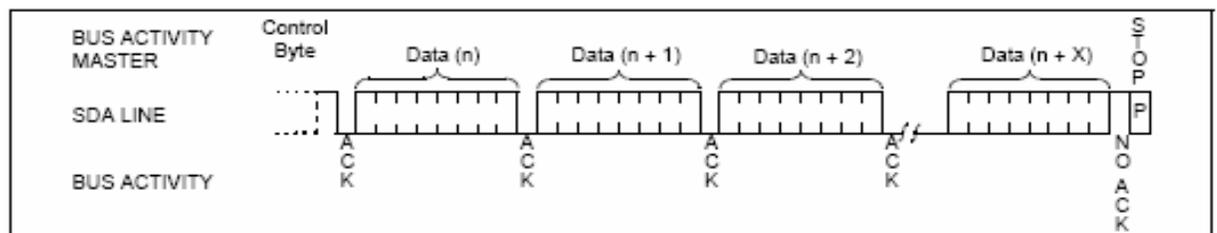
En el ciclo de escritura, luego de configurar el byte de control, se envían dos bytes que indicarán la dirección en la que se desea que se inicie la escritura y luego se envía los bytes que se deseen almacenar, intercalando después de cada byte una señal de reconocimiento. Luego del último byte de escritura se envía una señal de reconocimiento y una señal de parada, para finalizar la transferencia de datos.

**Figura 18.** Formato del ciclo de escritura. ([www.microchip.com](http://www.microchip.com))



En el ciclo de lectura, se envía el byte de control para habilitar la lectura. Posteriormente se envían dos bytes para indicar la posición inicial de acceso de la memoria. Luego de cada byte enviado por la memoria hacia el microcontrolador, es necesario generar una señal de reconocimiento para continuar con la transferencia. Para finalizar el ciclo, se envía una señal de parada luego del último byte, en lugar de la señal de reconocimiento.

**Figura 19.** Formato del ciclo de lectura. ([www.microchip.com](http://www.microchip.com))



#### **4.4.2 Diseño electrónico**

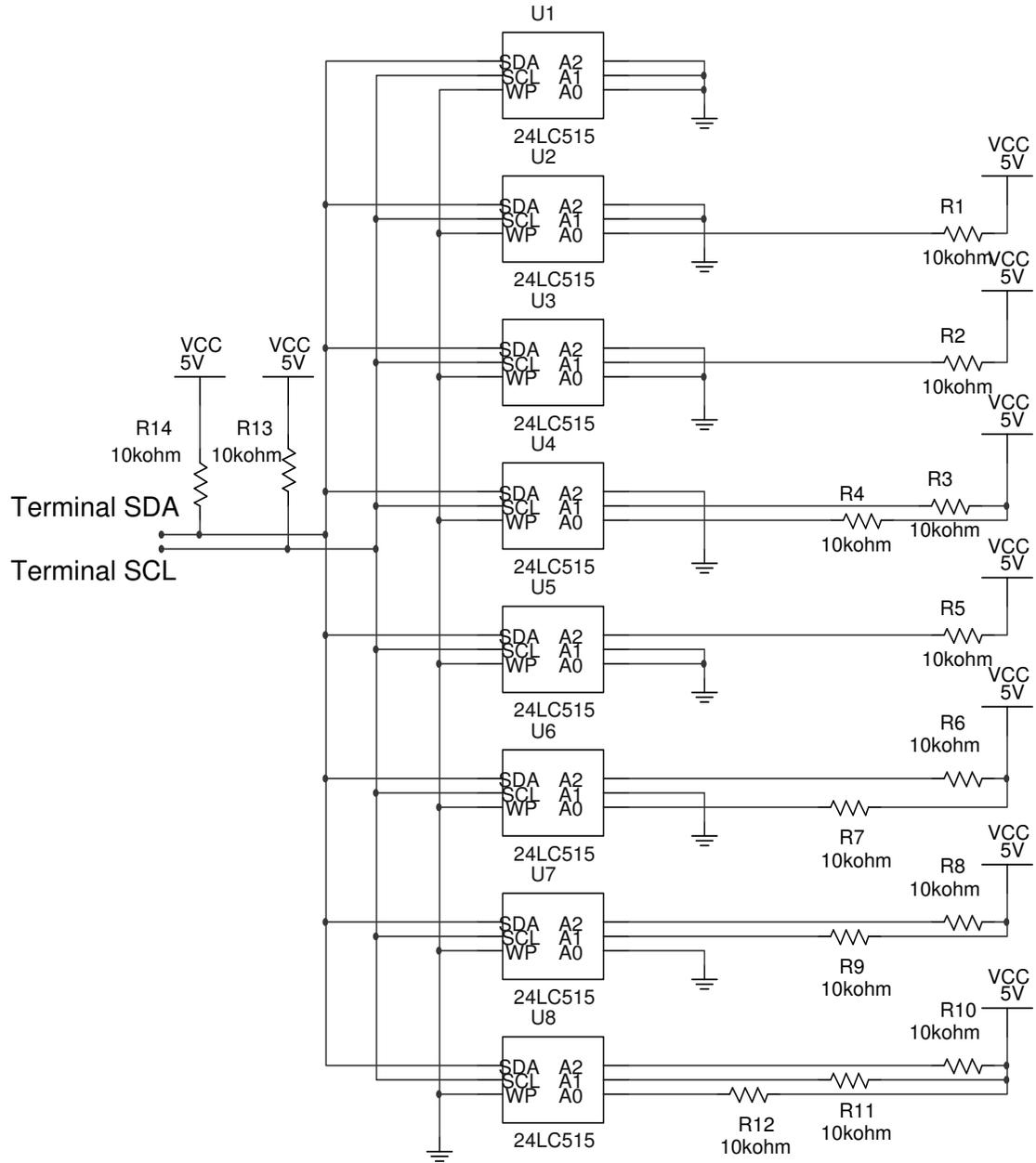
En la etapa de almacenamiento externo se utilizarán ocho memorias 24AA512 de Microchip en cascada, para permitir un mayor almacenamiento de datos y un mayor tiempo de operación. Económicamente se ha determinado utilizar un banco de memorias para ampliar el tiempo de operación del anemógrafo y evitar que los datos almacenados deban ser descargados a una computadora periódicamente. En principio, el diseño permite almacenar datos por períodos de tiempo mayores a un año, dependiendo esto del período de muestreo. A menor período de muestreo, será menor el tiempo que el dispositivo almacenará datos, debiendo ser descargados los datos almacenados con mayor frecuencia.

Las memorias se configurarán físicamente con resistencias para asignarles una dirección, utilizando sus terminales A0, A1 y A2. Todos los terminales seriales de datos (SDA) de cada memoria se conectan a un punto en común, que comunica con la terminal de datos del microcontrolador, y las de reloj serial (SCL) se conectan de la misma forma, ya que por medio del byte de control se direcciona la memoria serial a utilizar. Para los terminales de direccionamiento, el fabricante recomienda utilizar una resistencia de 2 k $\Omega$  a 10 k $\Omega$  para limitar la corriente cuando se conectan con la terminal de referencia positiva de voltaje. En el caso de los terminales conectados a tierra no es necesaria la utilización de una resistencia limitadora de corriente.

De la misma forma se conectarán en común las terminales Vcc y Vss, a la alimentación que especifica el fabricante, 5 voltios en Vcc y referencia a tierra en Vss. La terminal de protección de escritura será deshabilitada conectándola a tierra. Las terminales de datos serial y de reloj serial deben ser conectadas a

referencia de voltaje positivo utilizando resistencias de 10 kΩ para mantener siempre un estado alto aún cuando no se realice una transferencia de datos.

**Figura 20.** Diagrama esquemático del banco de memorias seriales EEPROM.



Cada memoria tiene capacidad para 512000 bits es decir, que podemos almacenar 64000 bytes. El formato para el almacenamiento de la información en la memoria se muestra en la siguiente tabla, que se ha diseñado para poder tener información de la fecha y hora en que es realizada la medición, y el valor promedio de las muestras tomadas en el período de tiempo asignado para obtener la velocidad promedio del viento.

**Tabla I.** Formato de muestra almacenada en memoria.

<b>B0</b>	1	0	1	0	1	0	1	0
<b>B1</b>	X	A6	A5	A4	A3	A2	A1	A0
<b>B2</b>	X	X	X	X	ME3	ME2	ME1	ME0
<b>B3</b>	X	X	X	D4	D3	D2	D1	D0
<b>B4</b>	X	X	X	H4	H3	H2	H1	H0
<b>B5</b>	X	X	M5	M4	M3	M2	M1	M0
<b>B6</b>	V7	V6	V5	V4	V3	V2	V1	V0
<b>B7</b>	0	1	0	1	0	1	0	1

Cada una de las muestras almacenadas en memoria estará constituida por 8 bytes, y cada uno de los bytes almacenará la siguiente información:

- **B0:** byte de reconocimiento de encabezado, el cual almacenará el código hexadecimal 0AAH, lo cual nos permitirá reconocer el inicio de una muestra durante el procesamiento de la información en la computadora.
- **B1:** los siete bits menos significativos almacenarán el año de la muestra, lo cual nos permite guardar la decena y la unidad de cada año en código hexadecimal, el bit menos significativo almacenará siempre un valor '0' binario. Durante el procesamiento de las muestras en la computadora el

bit más significativo del byte 1 (B1) y cualquier otro bit marcado con X en la tabla anterior, se asignará el valor de '0' binario.

- **B2:** los cuatro bits menos significativos de este byte permiten almacenar en formato numérico el mes en el cual se ha realizado la adquisición de la muestra.
- **B3:** los cinco bits menos significativos almacenarán el día en el cual se realizó la muestra.
- **B4:** los cinco bits menos significativos almacenan la hora de la muestra en formato de 24 horas.
- **B5:** los seis bits menos significativos almacenan el minuto en el cual se realizó la muestra.
- **B6:** este byte almacena el valor promedio de los registrados durante el período de muestras. Este byte registra el valor de salida promedio del convertidor de analógico a digital.
- **B7:** byte de reconocimiento, el cual almacenará el código hexadecimal 055H, lo cual nos permitirá reconocer el final de una muestra durante el procesamiento de la información en la computadora.

Al tener 8 bytes cada muestra, esto nos da la capacidad de almacenamiento de cada memoria que es de 8000 muestras, por lo que podemos almacenar hasta 64000 muestras en el banco de memorias. En el microcontrolador se almacenarán 8 muestras (64 bytes) en su memoria

EEPROM, antes de realizar una transferencia hacia la memoria externa durante los ciclos de escritura. Entre las opciones de escritura de la memoria, ésta nos permite realizar escritura de tres modos: escritura de un byte, escritura de una página y escritura total del dispositivo. Utilizaremos la escritura de páginas (64 bytes) dado a la capacidad de almacenamiento de la EEPROM de un PIC no excede 256 bytes y así se realizarán menos ciclos de escritura en la memoria externa.

De acuerdo a lo presentado anteriormente, podemos concluir que en cada memoria solamente podremos realizar 125 transferencias de páginas para poder utilizar cada una de éstas a plena capacidad.

#### **4.5 Interfaz serial computadora anemógrafo**

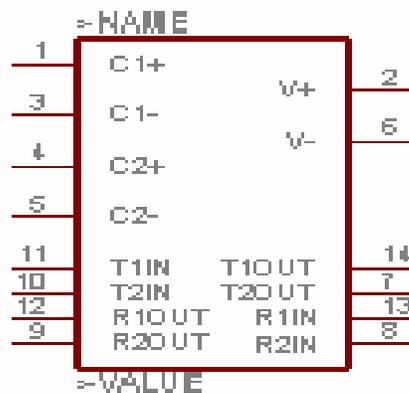
La etapa de interfaz serial permite la conexión física del instrumento con una computadora. Debido a que el instrumento a diseñar, utilizará voltajes TTL (0 V a 5 V) y la computadora en su puerto serial utiliza voltajes RS-232C ( $\pm 12$ ), no se puede realizar una conexión directa entre ambos dispositivos.

Para esta etapa del diseño se ha evaluado la opción de diseñar los dispositivos para cambiar los niveles de voltaje de TTL a RS-232C. Para el caso de la conversión de TTL a RS-232C es necesaria la utilización de un comparador analógico de alta frecuencia o un amplificador operacional para alta frecuencia, además de una fuente bipolar de voltaje. En el caso de la conversión de RS-232C a TTL, es necesaria la utilización de un diodo Zener para limitar el voltaje y luego de esto, se utiliza una compuerta digital inversora de voltaje.

En el mercado existe un circuito integrado llamado MAX 232, el cual necesita únicamente 4 capacitores de 4.7  $\mu\text{F}$  y alimentación TTL para funcionar.

Además posee dos terminales para conversión TTL a RS-232C y dos terminales para conversión RS-232C a TTL. La ventaja de este integrado es la utilizar capacitores para multiplicadores de voltaje y obtener una fuente bipolar a partir de una alimentación de corriente continua.

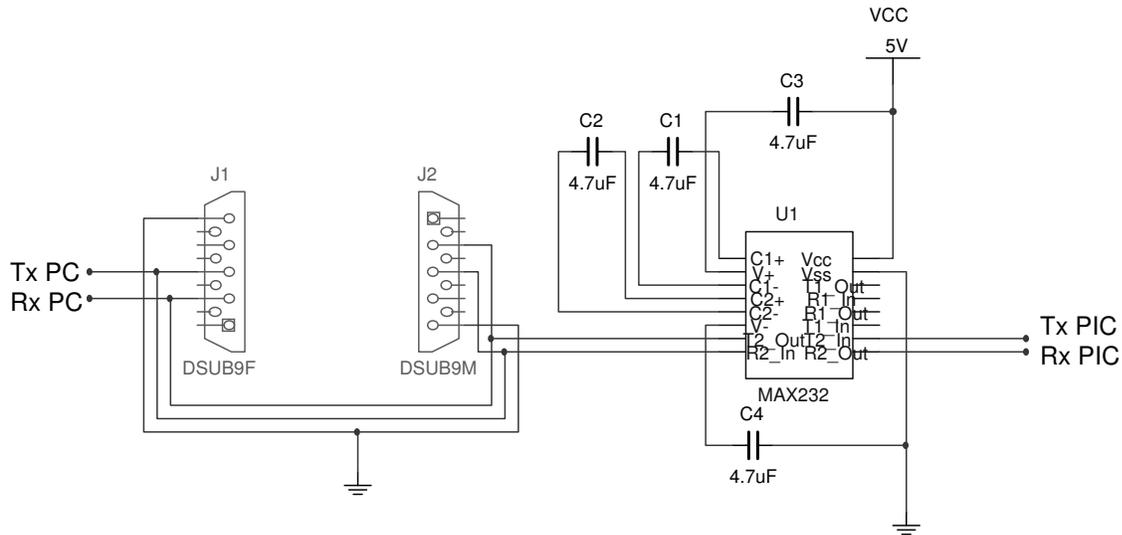
**Figura 21.** Configuración de terminales de MAX 232. ([www.cadsoft.com](http://www.cadsoft.com))



Además para conectar con la computadora es necesaria la utilización de conectores tipo DB-9. Para la comunicación, únicamente utilizaremos tres pines de este puerto: tierra (GND /5), transmisor (TX /3) y receptor (RX /2).

La interfaz serial debe realizar el acople de voltajes entre el microcontrolador y la computadora, permitiendo una comunicación bidireccional. La conexión se realizará utilizando un canal de conversión de niveles TTL a niveles RS-232C para la comunicación del microcontrolador a la computadora y un canal de conversión de niveles RS-232C a niveles TTL para la comunicación entre la computadora y el microcontrolador. Además se dispondrá de un conector DB-9 para el cable de comunicación serial con la computadora.

**Figura 22.** Diagrama esquemático de conexión serial del microcontrolador con la computadora.



#### 4.6 Despliegue de datos

El anemógrafo desplegará un valor promedio de la velocidad del viento, simultáneamente con el proceso de almacenamiento digital de estos. Además desplegará la fecha y la hora para poder controlar el funcionamiento del dispositivo y para facilitar el proceso de muestreo manual en lugares retirados.

Existen varias alternativas para desplegar datos, dentro de las cuales se encuentran el display led y el LCD. Debido a la versatilidad para desplegar una amplia cantidad de caracteres, bajo consumo de potencia, precio, sencillez de instalación y utilización utilizaremos un display de cristal líquido para el despliegue de datos en el anemógrafo. La alternativa de utilizar display de leds se ha descartado por el alto consumo de energía de los leds comparado con un LCD, la complejidad del circuito de control y por su poca versatilidad para despliegue de caracteres distintos a los numéricos.

En el LCD se ha de mostrar la velocidad promedio del viento, simultáneamente con la hora y la fecha. Debido a la cantidad de información a desplegar se puede utilizar un LCD de una sola línea y desplegar alternadamente la velocidad del viento con el tiempo y la hora. Pero al evaluar el costo y el tamaño de un display de una sola fila contra el de dos, no existe gran diferencia. Así que se utilizará un display de dos filas, lo que hará más sencillo el programa para el despliegue de datos. De acuerdo a los display disponibles en el mercado, he decidido utilizar un display de 16 columnas por 2 filas de la marca Polytronix, serie PVC-1602XX, debido a la gran cantidad de información proporcionada por el fabricante para la configuración y uso del dispositivo.

**Figura 23.** Distribución de la información a mostrar en LCD.



El LCD Polytronix PVC – 160204A a utilizar posee las siguientes características:

- Dos filas de caracteres y cada fila, 16 columnas (32 caracteres en total). Cada carácter esta constituido por un arreglo matricial de puntos que conforman 8 filas y columnas.
- Caracteres formados por arreglos matriciales de puntos ordenados en 8 filas y 5 columnas.
- Consumo menor a 5 mA.
- Memoria de hasta 80 caracteres en pantalla.
- 8 caracteres configurables.
- Luz para la pantalla y control de contraste de los caracteres.

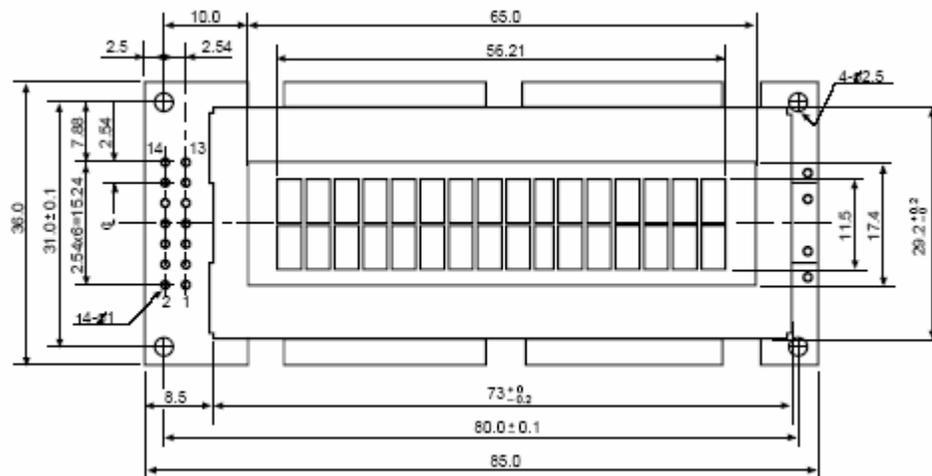
Este LCD al igual que cualquier otro, esta constituido por un circuito de control integrado en el impreso de la pantalla, pines de conexión y un banco de memoria RAM.

La memoria RAM de LCD se puede dividir en dos tipos:

- **DD RAM:** la memoria RAM de despliegue de datos, es la que almacena los caracteres que se despliegan en la pantalla LCD, esta constituida por 80 bytes, divididos en dos bloques de 40 bytes, uno por cada fila del display. Debido a que la pantalla solamente puede desplegar 16 caracteres por fila, se debe de indicar la posición de memoria donde se encuentra el primer carácter a mostrar de los 40 almacenados.
- **CG RAM:** la memoria RAM generadora de caracteres esta constituida por 64 posiciones de 5 bits cada una, las cuales se agrupan en bloques

contiguos de 8 posiciones para formar cada una un carácter personalizado por el diseñador.

**Figura 24.** Diagrama de pantalla PVC – 160204A. ([www.polytronix.com](http://www.polytronix.com))



La pantalla LCD consta de 14 pines de conexión, distribuidos de la siguiente forma:

- **Vss (#1):** referencia de voltaje positivo (5 V).
- **Vdd (#2):** referencia de voltaje positivo (0 V).
- **VO (#3):** control de ajuste de contraste.
- **RS (#4):** selección de registro.
- **R/W (#5):** selección de lectura o escritura.
- **E (#6):** habilitador.
- **D0... D7 (#7... #14):** terminales de datos.

De acuerdo a la distribución de sus terminales de conexión, se puede identificar dos tipos de buses en el LCD, el bus de datos y el bus de control. El

bus de datos esta constituido por las 8 terminales de datos, mientras que el bus de control está constituido por las terminales de Habilitación, Lectura / Escritura y la de Selección de registro.

El bus de control no permite controlar la lectura y escritura del registro seleccionado. Con la terminal RS podemos escoger entre el registro de datos (constituido por la memoria DD RAM y la CG RAM) y el registro de control, en el cual se ejecutan con ciclos de escritura los comandos de control de la pantalla y con los ciclos de lectura nos proporciona información del contador de posiciones en la pantalla.

Para los ciclos de escritura y lectura en cualquiera de los registros, debemos cumplir con la temporización requerida por el controlador de la pantalla. Esta se basa en iniciar seleccionando el registro a utilizar y seleccionado el tipo de ciclo (lectura o escritura), luego se envía un pulso en la terminal de habilitación (E), y su duración no debe ser menor de 0.5 microsegundos y durante el pulso se coloca el dato o comando en las terminales de datos.

El display a utilizar, posee una serie de comandos para su operación. Los comandos disponibles para este display se enumeran en la tabla II. Además se debe de considerar que es necesario un tiempo mínimo de espera por comando ejecutado, este tiempo se puede determinar realizando una lectura del registro de control, y revisando el estado del bit más significativo del registro que nos indica un estado de espera. Otra forma, es utilizar el tiempo mínimo de ejecución que recomienda el fabricante, 40  $\mu$ s.

Tabla II. Resumen de comandos de LCD. (Manual de LCD)

	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
Borrar display	0	0	0	0	0	0	0	0	0	1
Cursor a inicio	0	0	0	0	0	0	0	0	1	*
Establecer modo de funcionamiento	0	0	0	0	0	0	0	1	I/D	S
Control On / Off	0	0	0	0	0	0	1	D	0	D
Desplazamiento del cursor / display	0	0	0	0	0	1	S/C	R/L	*	*
Modo de transferencia	0	0	0	0	1	DL	1	0	*	*
Acceso a memoria CG RAM	0	0	0	1	Dirección de la CG RAM					
Acceso a memoria DD RAM	0	0	1	Dirección de la DD RAM						
Lectura de dirección y del flag de ocupado	0	1	BF	Contador de Dirección						
Escritura de datos en CG RAM / DD RAM	1	0	Dato a escribir							
Lectura de datos en CG RAM / DD RAM	1	1	Dato leído							

I/D = 1: Incrementa contador de direcciones	I/D = 0: Decrementar contador de direcciones
S = 1: Desplazamiento del display	S = 0: Display quieto
D = 1: Display On	D = 0: Display Off
C = 1: Cursor On	C = 0: Cursor Off
B = 1: Parpadeo del carácter en la posición del cursor	B = 0: No hay parpadeo
S/L = 1: Desplazar el display	S/L = 0: Desplazar el cursor
R/L = 1: Desplazamiento a la derecha	R/L = 0: Desplazamiento a la izquierda
DL = 1: Configurar display a 8 bits	DL = 0: Configurar el display a 4 bits
BF = 1: Display ocupado	BF = 0: Display lista para ejecutar otra operación

#### 4.6.2 Diseño electrónico

Para el diseño electrónico de la pantalla únicamente es necesario tomar en consideración la alimentación del dispositivo, el control del contraste, las conexiones del bus de datos y las del bus de control.

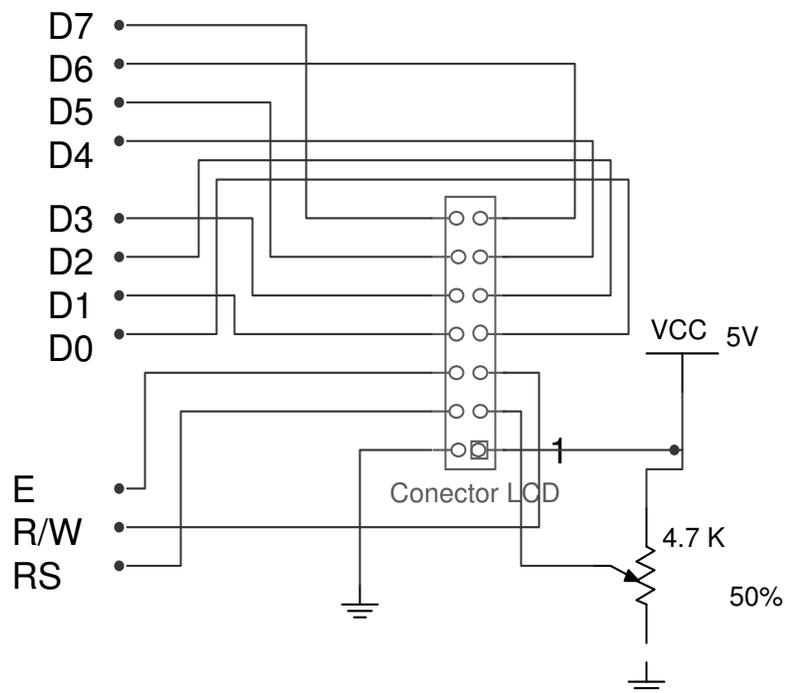
Para la alimentación de la pantalla LCD PVC-160204A solamente es necesario conectar la terminal Vdd a 5 voltios y la terminal Vss a 0 voltios o la referencia del circuito.

Para el control del contraste es necesario únicamente utilizar un divisor de voltaje que permite poseer un rango de variación desde 0 voltios hasta 5

voltios. Para lo cual utilizaremos un potenciómetro de 4.7 K $\Omega$  de acuerdo a las especificaciones del fabricante.

Tanto para las conexiones del bus de datos y del bus de control se pueden realizar directamente entre el microcontrolador y la pantalla, no es necesario la utilización de resistencias conectadas a la referencia de voltaje positivo.

**Figura 25.** Diseño esquemático de conexión de pantalla LCD.



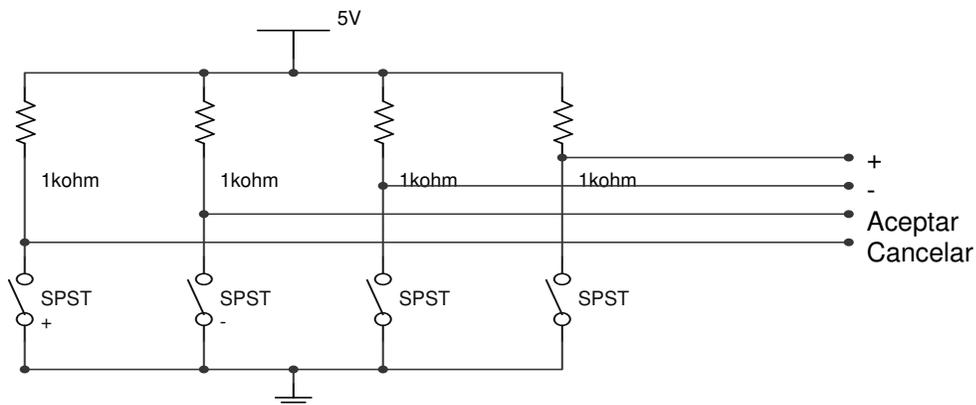
## 4.7 Teclado

El teclado nos permitirá configurar el reloj, la fecha y la constante de proporcionalidad del diseño de acuerdo a pruebas y a cambios en el transductor. Para eso será necesaria además una contraseña para evitar que sean alterados los parámetros del dispositivo por personas ajenas. El teclado constará de 4 teclas de control, una para aumentar el valor del número a ingresar, otra para disminuir el valor a ingresar, una para aceptar el ajuste realizado y otra para cancelar el ajuste o salir del menú.

### 4.7.1 Diseño electrónico

Para el diseño, se ha considera utilizar interruptores normalmente abiertos. Cada una de las terminales de los interruptores estarán conectados a tierra en un extremo y en el otro conectados a la terminal correspondiente para cada tecla en el microcontrolador, a la vez se conectará cada terminal del microcontrolador a 5 voltios mediante una resistencia de 10 K $\Omega$  para mantener un estado lógico alto cuando no sea pulsada ninguna de las teclas.

**Figura 26.** Diseño esquemático del teclado del anemógrafo.



## 4.8 Microcontrolador

Para el control del anemógrafo se utilizará un microcontrolador PIC 18F458 de encapsulado tipo DIP. Este uno de los modelos más recientes de microcontroladores disponibles en Microchip. Este microcontrolador posee las siguientes características lógicas:

- Arquitectura RISC.
- 76 instrucciones disponibles.
- Memoria tipo Flash e EEPROM.
- Memoria de programa de 2 Mbytes.
- Memoria de datos de 4 kbytes.
- Operación con reloj de hasta 40 MHz (10 MIPS)
- Bus de datos de 8 bits e instrucciones de 16 bits.
- Interrupciones programables de hardware y software.
- Reloj externo para temporizadores.

El microcontrolador PIC 18F458 posee las siguientes características periféricas:

- Corriente de hasta 25 mA en terminales de salida / entrada.
- Tres terminales de interrupciones externas.
- Un contador de 16 bits y tres contadores de 8 bits.
- Modulo de captura / comparación / modulación por ancho de pulso.
- Puerto serial síncrono.
- Modulo serial universal asíncrono.
- Modulo de conversión analógica digital de 8 canales y cada uno con resolución de 10 bits.

- Comparador analógico.
- Detección de bajo voltaje.

#### 4.8.1 Diseño electrónico

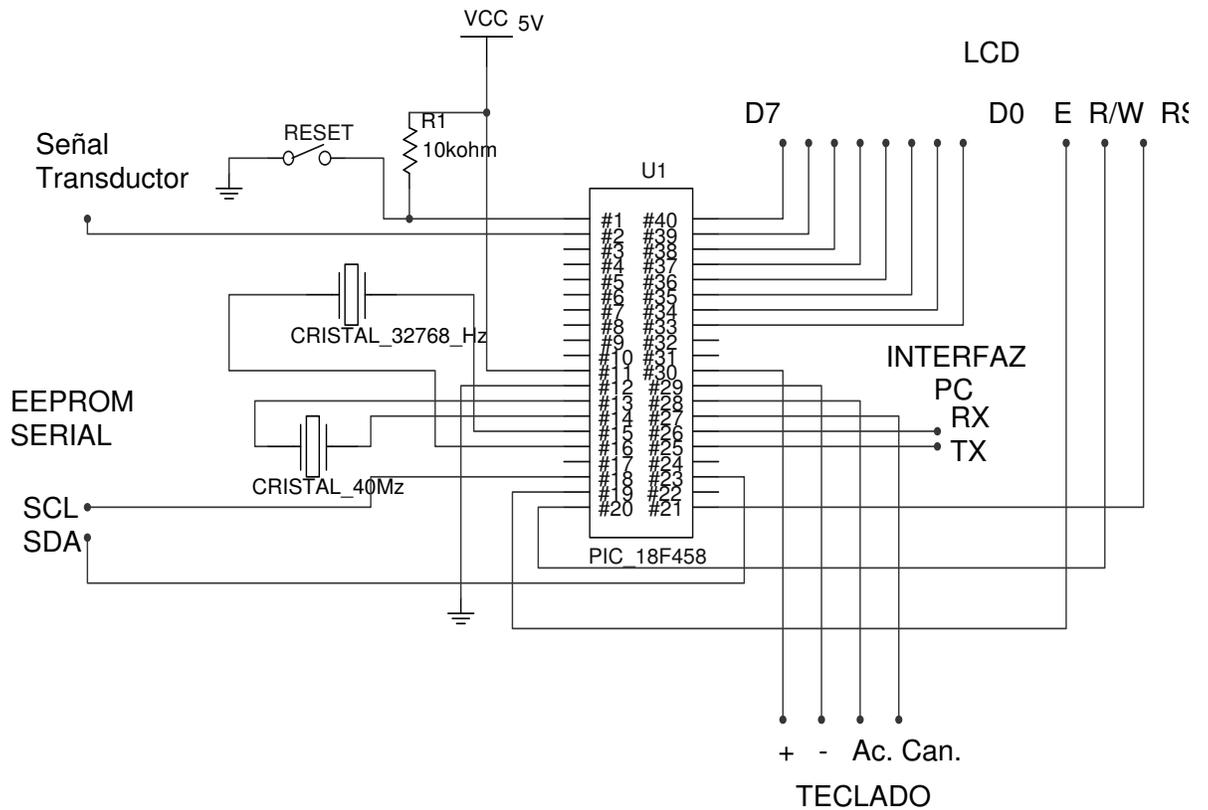
Para el diseño electrónico, asignaremos cada una de las terminales de comunicación con los demás dispositivos que conforman al anemógrafo. Las terminales del microcontrolador se conectarán de la siguiente forma:

- **MCLR (#1):** en esta terminal se colocará un interruptor normalmente abierto con una resistencia Pull-up a la terminal de voltaje. Se emplea para reiniciar el microcontrolador.
- **RA0 (#2):** terminal de entrada analógica para la señal proveniente del transductor.
- **Vdd (#11 y #32):** terminales de alimentación de voltaje positivo de 2.5 hasta 5 voltios.
- **Vss (#12 y #31):** terminales de referencia a tierra.
- **Reloj (#13 y #14):** terminales de conexión para el reloj de temporización del microcontrolador.
- **RC1 / T1OSI (#16):** terminal digital para señal de reloj externa para contador Timer0.

- **RC3 / SCL (#18):** terminal digital para señal de reloj en comunicación serial de periféricos con la memoria EEPROM.
- **RC4 / SDA (#23):** terminal digital para señal de datos con el banco de memorias seriales EEPROM.
- **RC6 / TX (#25):** terminal digital para transmisión de datos hacia una computadora.
- **RC7 / RX (#26):** terminal digital para recepción de datos de la computadora.
- **RB0... RB7 / D0... D7 (#33... #40):** terminales de salida digital hacia la pantalla LCD para la transmisión de datos a desplegar y de comandos de operación.
- **RD0 / E (#19):** terminal de salida digital para habilitar la pantalla LCD.
- **RD1 / R/W (#20):** terminal de salida digital para determinar un ciclo de lectura o de escritura en la pantalla LCD.
- **RD2 / RS (#21):** terminal de salida digital para seleccionar el registro de datos o el registro de control de la pantalla LCD.
- **RD4 / + (#27):** terminales de entrada digital para la tecla de aumentar.
- **RD5 / - (#28):** terminal de entrada digital para la tecla de disminuir.

- **RD6 / Aceptar (#29):** terminal de entrada digital para la tecla de aceptar.
- **RD7 / Cancelar (#30):** terminal de entrada digital para la tecla de cancelar.

**Figura 27.** Diagrama esquemático de conexión de microcontrolador PIC 18F458.



## 4.8.2 Programación del microcontrolador

Luego de terminar el diseño electrónico del anemógrafo, se debe de programar al microcontrolador para controlar los dispositivos que forman el anemógrafo y luego permitirnos obtener lecturas de la velocidad del viento, almacenar éstas y transferirlas a una computadora para su posterior análisis.

Para utilizar el microcontrolador de forma apropiada, debemos de configurar cada uno de los registros de los módulos a emplear. Los registros a configurar son los puertos, el convertidor analógico digital, las interrupciones, los contadores, el módulo USART para comunicación con la computadora, el módulo MSSP para comunicación con las memorias EEPROM seriales, configuración de la pantalla LCD e inicialización de las variables a utilizar.

### 4.8.2.1 Programación de puertos

Para utilizar cada uno de los puertos debemos configurar al menos dos registros y en otros casos tres o más registros. Los registros a utilizar son:

- **TRIS:** estos registros permiten configurar cada una de las terminales de los puertos como entrada o como una salida.
- **PORT:** estos registros nos permiten acceder a la información de los terminales en el caso de estar configurados como entradas y colocar información si son salidas.
- **LATB:** este registro almacena datos cuando están configurados los puertos como salidas.

#### 4.8.2.1.1 Configuración del puerto analógico y el convertidor analógico digital

Del puerto A se ha utilizar únicamente la terminal A0 (#2), como una entrada de la señal del transductor. Además de poder ser configurado como una entrada o una salida, este puerto se puede configurar como de entrada digital o analógica. Hemos de configurar la terminal A0 en el registro TRISA como una entrada y debemos configurar también el funcionamiento del modulo analógico digital.

El modulo digital posee cuatro registros:

- **ADRESH:** registro que almacena los bits más significativos de la salida del convertidor de analógico a digital.
- **ADRESL:** registro que almacena los bits menos significativos de la salida del convertidor analógico digital.
- **ADCON0:** es uno de los registros de control del convertidor. En este configuramos el canal de conversión, el reloj del convertidor, inicio y estado de la conversión y habilitamos el modulo del ADC.
- **ADCON1:** en este registro configuramos junto con los bits del ADCON0 la velocidad del reloj del ADC y configuramos al puerto A Y E para seleccionar si tienen un comportamiento como terminales analógicas o digitales. También configuramos la referencia de voltaje del convertidor, podemos seleccionar entre la alimentación del microcontrolador o poder configurar dos terminales para colocar nuestras referencias de voltaje.

Permite configurar si deseamos que el resultado del ADC sea almacenado justificado a la izquierda (almacena los 8 bits más significativos en el registro ADRESH) o justificado a la derecha (los 8 bit menos significativos son almacenados en el registro ADRESL).

Para el registro TRISA únicamente se utilizará el bit A0, para configurarlo como una entrada. Los demás pines del puerto no serán utilizados.

**Figura 28.** Registro TRISA y su configuración.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTA	—	RA6	RA5	RA4	RA3	RA2	RA1	RA0

-	0	0	0	0	0	0	1
A6	A5	A4	A3	A2	A1	A0	

Cada uno de los bits del registro ADCON0 tiene la siguiente función:

- **ADCS1 y ADSC0:** permiten configurar la velocidad de conversión. De acuerdo al fabricante, mientras más grande sea el tiempo de conversión, más preciso será el valor digital de salida. Utilizaremos el valor (11) que configura la velocidad del ADC a 64 ciclos de instrucción por conversión.

- **CHS0, CHS1 y CHS2:** permiten seleccionar entre los 8 canales de conversión. Estableceremos el valor de 000 para seleccionar como canal de conversión el bit A0.
- **GO / DONE:** cuando se coloca el bit en '1' da inicio a la conversión analógica digital, se mantiene así hasta que termina ésta. El estado de la conversión también puede ser verificado por una bandera en los registros de interrupciones. La configuraremos como '0' al inicio.
- **ADON:** este bit permite encender o apagar el ADC. La activaremos cada vez que utilicemos el convertidor y cuando termine la conversión la desactivaremos, esta es una opción del microcontrolador para ahorrar energía. La configuraremos inicialmente con '0'.

**Figura 29.** Registro ADCON0 y su configuración.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

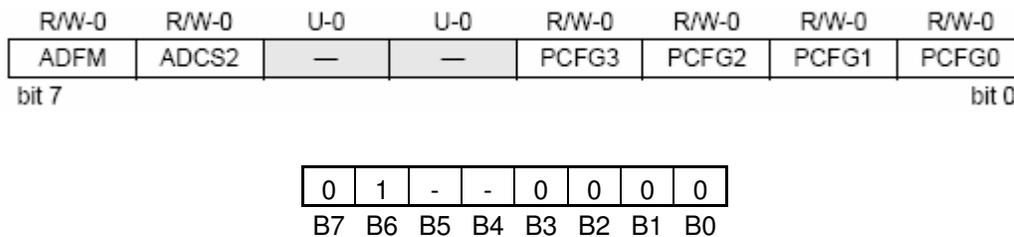
1	0	0	0	0	-	0	0
B7	B6	B5	B4	B3	B2	B1	B0

En el caso del registro ADCON1, cada uno de sus bits tienen la siguiente función:

- **ADFM:** este permite seleccionar la justificación del resultado de la conversión. Seleccionaremos justificación a la izquierda ('1') para que los 8 bits más significativos sean almacenados en el registro ADRESH.

- **ADCS2:** este junto con ADCS1 y ADSC0, configuran la velocidad del reloj de conversión. Este se configura en '1' para tener la velocidad de 64 ciclos de instrucción.
- **PCFG3, PCFG2, PCFG1 y PCFG0:** estos permiten configurar a los terminales del puertos A y E como analógicos o digitales, además de configurar la referencia de voltaje para el ADC. Seleccionaremos el código '0000' que configura al bit RA0 como analógico y a las fuente de alimentación como referencia de voltaje.

**Figura 30.** Registro ADCON1 y su configuración.



#### 4.8.2.1.2 Configuración de los puertos digitales

Para los demás puertos, al ser todos digitales, es necesario únicamente configurar el registro TRIS de cada uno, para determinar cuales funcionarán como salidas y cuales como entradas.

Para el caso del puerto B, el cual se conecta con el bus de datos de la pantalla LCD, se configura todos sus bits como salidas, almacenado el código '00000000'.

**Figura 31.** Configuración del registro TRISB.

0	0	0	0	0	0	0	0
B7	B6	B5	B4	B3	B2	B1	B0

El puerto C será utilizado para reloj externo para el temporizador, los módulos de comunicación USART y el IIC. Para cada uno de los pines deberemos referirnos al funcionamiento de cada pin para poder configurarlos.

- **TOSI (RC1):** entrada para la señal del reloj externo. ('1')
- **SCL (RC3):** es la salida del reloj serial. ('0')
- **SDA (RC4):** es la terminal de datos bidireccional. Se configura inicialmente como entrada, pero durante la operación, el modulo lo configura como entrada o salida de acuerdo a la operación. ('1')
- **TX (RC6):** se configura como una salida. ('0')
- **RX (RC7):** se configura como una entrada. ('1')

Las demás terminales no serán conectadas físicamente, así que no tiene importancia su configuración, de cualquier forma, las configuraremos como salidas.

**Figura 32.** Configuración del registro TRISC

1	0	0	1	0	0	1	0
C7	C6	C5	C4	C3	C2	C1	C0

En el caso del puerto D, tenemos 3 terminales que funcionan como salida (bus de control del LCD), 4 terminales que funcionan como entradas (teclado) y una terminal sin conexión. Para las terminales E (RD0), R/W (RD1) y RS (RD2) estableceremos el código '000' por ser salidas. Para las terminales Aumentar (RD4), Disminuir (RD5), Aceptar (RD6) y Cancelar (RD7), estableceremos el código '1111' por ser entradas. El registro TRISD deberá ser configurado con el código '11110000'.

**Figura 33.** Configuración del registro TRISD

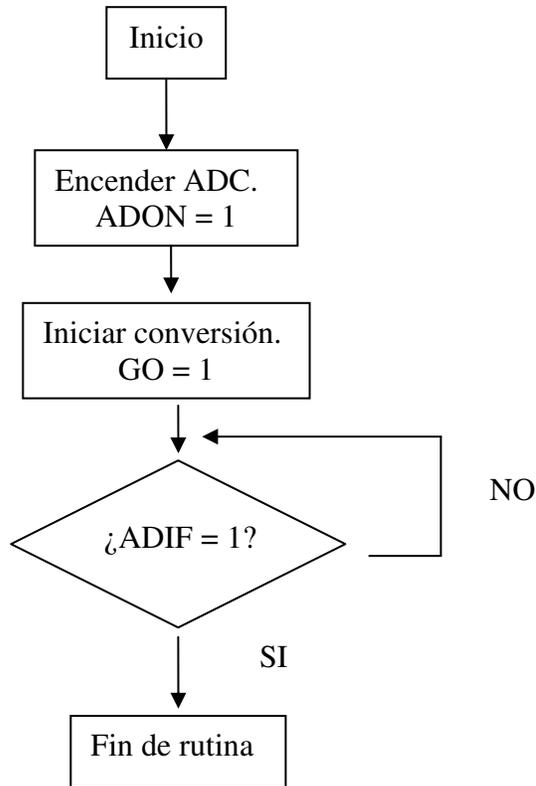
1	1	1	1	0	0	0	0
D7	D6	D5	D4	D3	D2	D1	D0

#### 4.8.2.1.3 Adquisición de señales del transductor

Luego de configurar el convertidor de analógico a digital, debemos de realizar una rutina para el control de la conversión. Esta rutina se debe de encargar de habilitar el ADC, luego iniciar la conversión y por último permanecer en espera de que concluya la conversión para poder almacenar el resultado. Cuando el ADC ha realizado una conversión, la bandera ADIF

(Bandera de interrupción del ADC) se coloca en '1'. La bandera ADIF se encuentra en el registro PIR1 (registro de interrupciones periféricas).

**Figura 34.** Algoritmo de rutina de adquisición de datos.



#### 4.8.2.2 Programación de Comunicación serial con memoria EEPROM

Para la comunicación serial de las memorias con el microcontrolador utilizaremos el módulo MSSP (transmisión maestro serial síncrona con periféricos). Debemos de recordar que el banco de memorias seriales consta de 8 memorias EEPROM, y cada una, esta configurada con una dirección física distinta que se puede acceder configurando correctamente el bit de control.

Para utilizar la comunicación serial con periféricos es necesario configuración IIC (Comunicación entre circuitos integrados). Para hacer uso de este modo de comunicación es necesario configurar 4 registros y hacer uso de otro.

#### **4.8.2.2.1 Configuración de registros para comunicación IIC**

Los registros a configurar para la comunicación IIC son:

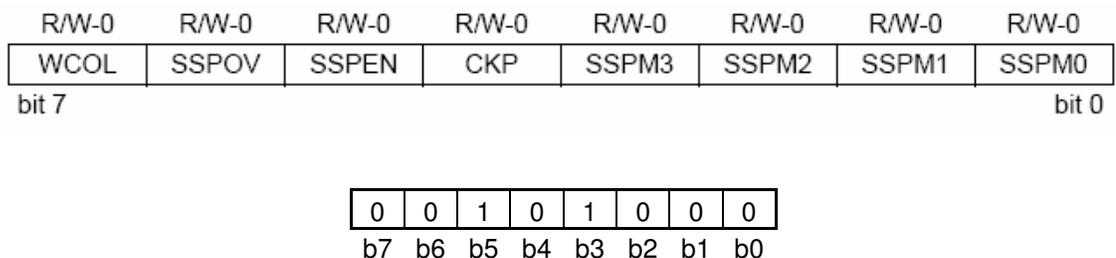
- **SSPCON1:** registro 1 de control de comunicación maestro serial síncrona con periféricos (MSSP).
- **SSPCON2:** registro 2 de control de comunicación MSSP.
- **SSPSTAT:** registro de estado de comunicación MSSP.
- **SSPADD (BRG):** registro de velocidad de transmisión.

El registro SSPCON1 permite habilitar el puerto serial, la velocidad de éste y el estado del puerto. Cada uno de los bits de este registro posee la siguiente función:

- **WCOL:** detección de colisión en escritura, nos permite determinar si el estado del puerto es valido para iniciar la transmisión. Se desactiva ('0') para poder estar listos para una transmisión, también nos puede servir para detectar si el puerto esta listo par usarse.

- **SSPOV:** indica condición de sobre flujo en la recepción de datos, se está recibiendo un dato nuevo y el buffer aún contiene el dato anterior. Se desactiva ('0') cuando se inicia la comunicación.
- **SSPEN:** habilita el puerto serial ('1').
- **CKP:** se utiliza solamente en el modo de esclavo para esperas. No tiene aplicación en el modo de maestro.
- **SSPM0...SPM3:** modo y velocidad de operación. El modo de funcionamiento del microcontrolador será maestro y utilizaremos la velocidad de operación más alta que proporciona el modulo, 400 Khz. El código a programar debe ser '1000'.

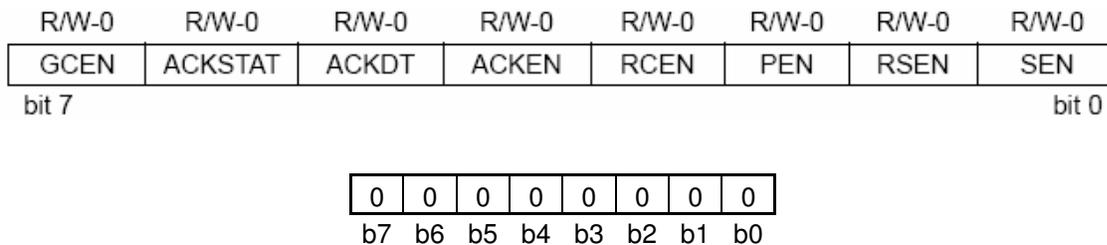
**Figura 35.** Registro SSPCON1 y su configuración.



El registro SSPCON2 permite generar las señales necesarias durante la operación del puerto, por lo cual, es un registro de trabajo. Para su configuración inicial se limpia el registro. Cada uno de los bits de este registro posee la siguiente función:

- **GCEN:** solo se utiliza en modo esclavo, habilita interrupciones cuando existe llamada a direcciones de acceso.
- **ACKSTAT:** solo funciona en modo maestro de transmisión, nos indica si existe una señal de reconocimiento del esclavo.
- **ACKDT:** solo funciona en modo maestro de recepción, genera las señales de reconocimiento.
- **ACKEN:** solo funciona en modo maestro de recepción, sirve para habilitar una señal de reconocimiento automática durante la recepción.
- **RCEN:** activa el modo de recepción.
- **PEN:** genera una señal de parada.
- **RSEN:** repite una señal de inicio.
- **SEN:** genera una señal de inicio.

**Figura 36.** Registro SSPCON2 y su configuración.



El registro SSPSTAT nos permite configurar la velocidad de operación y habilita los terminales de uso en la comunicación serial. Los demás bits son indicadores de estado. Cada uno de los bits de este registro tiene la siguiente función:

- **SMP:** selecciona entre velocidades menores a 400 Khz. y mayores a ésta.
- **CKE:** permite que los terminales del puerto C sean configurados para la transmisión serial.
- **D/A:** indica si la información recibida es un dato o es una dirección.
- **P:** indicador de señal de parada recibida.
- **S:** indicador de señal de inicio recibida.
- **R/W:** solo para recepción, indica el modo de operación, lectura o escritura.
- **UA:** indica que es necesario actualizar la dirección de acceso. Solo en el modo esclavo.
- **BF:** indica que el buffer esta lleno.

**Figura 37.** Registro SSPSTAT y su configuración.

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit 7							bit 0

0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0

El registro SSPADD se utiliza para configurar la velocidad de operación de la comunicación serial. De acuerdo al fabricante, la velocidad más alta de operación con un oscilador de 40 MHz. es de 400 KHz., y para obtener esta velocidad es necesario programar el registro con el código 18h.

**Figura 38.** Configuración de registro SSPADD.

0	0	0	1	1	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0

El registro SSPBUF es un registro de lectura / escritura, de acuerdo al modo de operación. Este registro es el registro de trabajo del puerto, es donde se almacena los datos recibidos y donde se colocan los datos por enviar.

#### 4.8.2.2 Programas de comunicación

Para establecer la comunicación en los dispositivos, se debe de seleccionar si el ciclo es de lectura o de escritura de datos, el proceso es similar para ambos casos.

Para la escritura de datos en la memoria desde el microcontrolador, únicamente se debe de configurar el modulo IIC del microcontrolador. Luego se envía una señal de inicio, a la que prosigue el bit de control, donde se indica la memoria a la cual se desea acceder y la operación a efectuar, en este caso, escritura de datos. A continuación se espera una señal de reconocimiento de la memoria hacia el microcontrolador, al recibir la señal podemos continuar con el proceso.

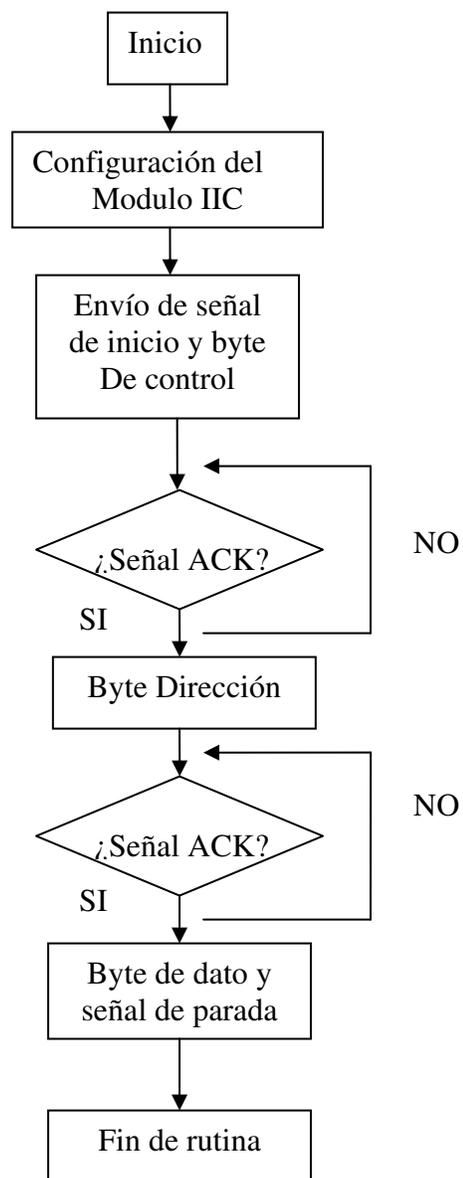
El siguiente byte a enviar es el byte más significativo de la dirección en donde se desea acceder, se espera una señal de reconocimiento, luego se envía el byte menos significativo de la dirección de memoria y se espera de nuevo una señal de reconocimiento de la memoria. Después de estos, podemos empezar a enviar la información a escribir en la memoria, cada byte debe ir acompañado de un estado de espera de una señal de reconocimiento de la memoria para continuar, la mayor cantidad de bytes posibles a escribir es una página (64 bytes). Para terminar la escritura, solamente se debe de enviar una señal de parada a continuación de recibir la señal de reconocimiento de la memoria.

Para la lectura el procedimiento es similar, enviamos una señal de inicio acompañada del byte de control. Ahora generamos una señal de reconocimiento y enviamos a continuación el byte más significativo de la posición de memoria a leer, otra señal de reconocimiento hacia la memoria y luego el byte menos significativo de la dirección, acompañado de una señal de reconocimiento.

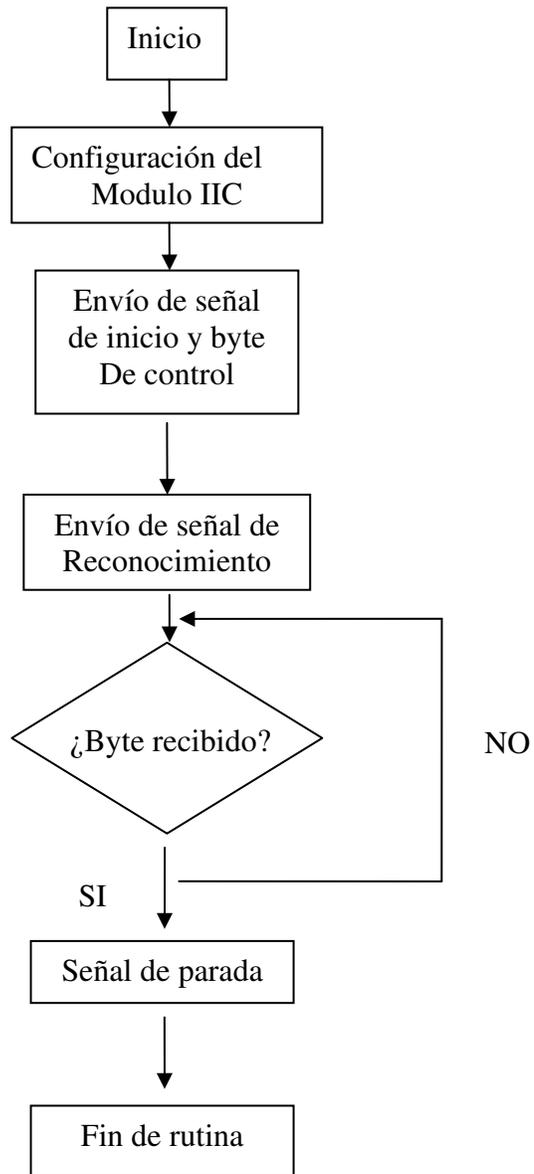
La lectura se efectúa, después de enviar el último byte de dirección y su señal de reconocimiento, enviado una señal de inicio nuevamente. Ahora empezamos a recibir bytes, después de cada byte recibido debemos de enviar

una señal de reconocimiento para continuar la lectura. Para finalizar el ciclo de lectura, en lugar de enviar una señal de reconocimiento, después del último byte recibido, se genera una señal de parada.

**Figura 39.** Algoritmo del ciclo de escritura del microcontrolador en la memoria serial EEPROM.



**Figura 40.** Algoritmo de lectura de memoria EEPROM por el microcontrolador.



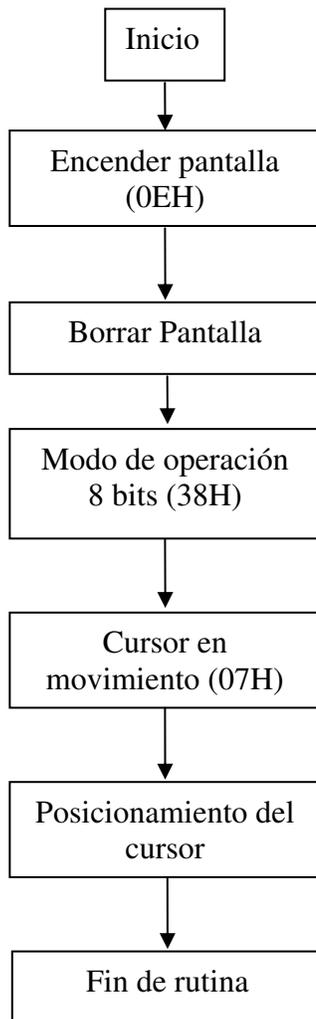
#### **4.8.2.3 Programación de pantalla de cristal líquido**

La pantalla de cristal líquido se debe de configurar antes de cada ciclo de escritura. El bus de datos se conecta al microcontrolador, para enviar los comandos y los datos a desplegar, y el bus de control se encarga de la configuración de la pantalla y de su temporización durante los ciclos de escritura y lectura. Para la utilización de la pantalla se debe de considerar la utilización de señales de control temporizadas que serán generadas por el microcontrolador.

La pantalla deberá de ser configurada cada vez que se encienda al dispositivo, cuando los datos desplegados en ella sean actualizados (fecha, hora y velocidad del viento) y para desplegar mensajes (memoria insuficiente, batería baja, menú de configuración y transferencia de datos hacia la computadora).

Para la configuración de la pantalla, se debe enviar el código (0EH) para encenderla. Luego se envía el comando de modo de operación, para indicar que se estará trabajando con 8 bits en el bus de datos (38H). A continuación borramos la memoria RAM de la pantalla, en caso de que existan datos en ella, y así evitamos que despliegue datos al azar durante su inicialización, con el código 01H. Se debe de indicar si el cursor avanzará cada vez que exista un ciclo de lectura o permanezca en su posición, enviando el código 07H para el primer caso y el código 06H en el segundo. Por último se envía un comando para indicar la posición del cursor, 02H para enviarlo a la posición inicial y en caso de querer posicionarlo en una dirección específica de la DDRAM, se debe de enviar el código binario '1XXXXXX', donde los últimos 7 bits indican la posición deseada.

**Figura 41.** Algoritmo de configuración de pantalla de cristal líquido.



Para escribir en la pantalla se debe de enviar únicamente la posición en la cual deseamos imprimir el carácter y luego se envía el dato a desplegar.

#### 4.8.2.4 Programación de la comunicación serial con la computadora

Para la comunicación serial con la computadora únicamente debemos de definir la velocidad de operación del puerto serial, cantidad de bits a enviar y la cantidad de bits de parada.

Para la estructura de nuestro dato, se ha decidido enviar byte por byte. Solamente utilizaremos un bit de parada. En el caso de la velocidad del puerto serial del microcontrolador, nos referiremos a la máxima velocidad a utilizar con un oscilador de cristal de 40 MHz, para la cual el error es mínimo. De acuerdo al fabricante para la velocidad de 96 kbps se tiene un porcentaje de error de tan solo 0.16%, donde el error se calcula en base de la diferencia porcentual de la velocidad deseada y la velocidad real de transferencia del microcontrolador, respecto a la velocidad deseada.

El microcontrolador consta de un modulo para comunicación serial síncrona asíncrona con computadora (USART). Para poder utilizarlo es necesario utilizar 9 registros del microcontrolador:

- **INTCON:** es el control de interrupciones que habilita las interrupciones globales y las periféricas.
- **PIR1:** contiene las banderas de las interrupciones de periféricos.
- **PIE1:** habilitador de interrupciones periféricas.
- **IPR1:** registro de asignación de prioridad a interrupciones periféricas.

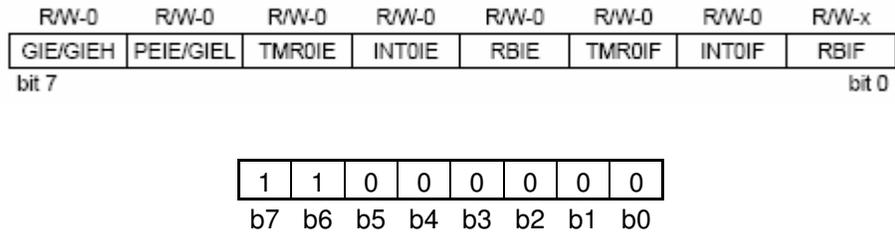
- **RCSTA:** registro de control y estado de la recepción.
- **TXSTA:** registro de control y estado de la transmisión.
- **SPBRG:** registro generador de tasa en baudios.
- **TXREG:** registro de transmisión del USART.
- **RCREG:** registro de recepción del USART.

#### 4.8.2.4.1 Configuración de registros para la comunicación serial

El registro INTCON nos permite habilitar las interrupciones globales, las interrupciones de periféricos y las interrupciones de algunos módulos (TIMER0 y el cambio de flanco). Solamente utilizaremos los bits más significativos que habilitan las interrupciones globales y periféricas, los demás serán deshabilitados, a excepción del TIMER0, que controlará la temporización del dispositivo. Los bits a configurar poseen la siguiente configuración:

- **GIE:** habilita las interrupciones globales, por lo que será configurado con un '1' para habilitar las interrupciones no enmascaradas de hardware.
- **PEIE:** habilita las interrupciones no enmascaradas de los módulos periféricos, configurado en '1' para utilizar las interrupciones generadas por el módulo USART durante la transmisión y recepción de datos.

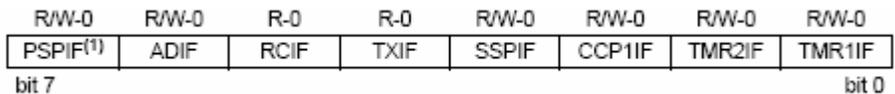
**Figura 42.** Registro INTCON y su configuración.



El registro de lectura PIR1 contiene las banderas de las interrupciones generadas por los módulos periféricos, respecto al módulo USART, este contiene las banderas que indican el estado de la transmisión y recepción serial. Los bits a monitorear en este registro son:

- **RCIF:** esta bandera nos indica que un dato ha sido recibido en el puerto serial.
  
- **TXIF:** esta bandera nos indica el estado de la transmisión, '0' para un dato transmitido y '1' para indicar que hay una transmisión en progreso.

**Figura 43.** Registro PIR1.

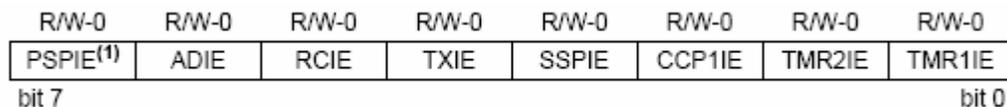


El registro PIE1 permite la habilitación de interrupciones no enmascarables de los módulos periféricos, entre estas interrupciones, se encuentran la habilitación de la interrupción de transmisión y recepción del USART. Las interrupciones de recepción y transmisión serán habilitadas

alternamente. Cuando el microcontrolador este funcionando, se habilitará la interrupción de recepción para detectar la petición de transmisión de datos hacia una computadora. Cuando se encuentre transmitiendo datos hacia la computadora se habilitará únicamente la interrupción de transmisión para determinar cuando se ha transmitido un dato y poder continuar enviando datos. Los bits a utilizar en el registro PIE1 son:

- **RCIE:** habilita la interrupción generada al ser recibido un dato en el módulo USART.
- **TXIE:** habilita la interrupción generada cuando se ha transmitido un dato por el módulo USART.

**Figura 44.** Registro PIE1.

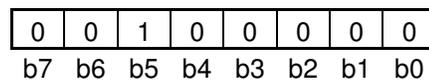
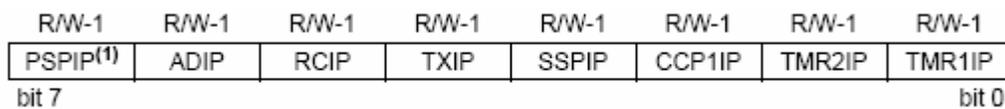


El registro IPR1 permite asignar la prioridad de una interrupción, En este caso, se tomará a la interrupción por recepción de datos como máxima prioridad. Los bits a configurar en este registro son:

- **RCIP:** determina el nivel de prioridad de la interrupción por recepción de datos en el módulo USART. Se le asigna '1', para determinar de forma inmediata una petición de datos hacia una computadora, así se accede a la rutina de interrupción para la comunicación de datos seriales.

- **TXIP:** determina el nivel de prioridad de la interrupción por transmisión de datos en el módulo USART. Se le asigna '0', ya que únicamente se monitoreará la bandera de interrupción para poder continuar con la transmisión de datos.

**Figura 45.** Registro IPR1 y su configuración.



Las interrupciones generadas durante la utilización del módulo USART, después de ser habilitadas, utilizan la dirección 08H como vector para las rutinas de interrupción.

El registro TXSTA nos permite configurar el funcionamiento del módulo USART durante la transmisión, habilita la transmisión y permite monitorear el estado de ésta. Los bits del registro de transmisión poseen la siguiente función:

- **CRSC:** este bit selecciona la fuente del reloj, en el caso de la transmisión asíncrona no es necesario configurarlo.
- **TX9:** este habilita la comunicación de 9 bits de datos, se deshabilitará.

- **TXEN:** este bit permite habilitar el módulo de transmisión de datos seriales del USART. Será fijado en '1' cada vez que se utilice el módulo de transmisión.
- **SYNC:** selecciona el modo de funcionamiento del módulo USART, síncrono o asíncrono. Estableceremos el valor de '0' para seleccionar el modo de transmisión asíncrono.
- **BRGH:** permite seleccionar la velocidad alta o baja del generador de la tasa de baudios, fijaremos el valor en '1' para tomar el valor más alto.
- **TRMT:** nos permite monitorear el estado del buffer de transmisión.
- **TX9D:** noveno dato de transmisión, que puede ser utilizado como dato o como bit de paridad.

**Figura 46.** Registro TXSTA y su configuración.

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

0	0	1	0	0	1	X	0
b7	b6	b5	b4	b3	b2	b1	b0

El registro RCSTA permite configurar el funcionamiento del módulo de recepción USART, el estado de la recepción y habilita el funcionamiento del módulo USART. Los bits del registro se configuran de la siguiente forma:

- **SPEN:** habilita el módulo USART. Se establecerá el valor de '1' para mantenerlo funcionando de forma continua.
- **RX9:** habilita la recepción de 9 bits de datos. Se deshabilitará esta función.
- **SREN:** habilitación de recepción simple, modo de funcionamiento no empleado en la configuración asíncrona.
- **CREN:** habilitación del modo de recepción continua. Se establecerá en valor '1' al inicio del programa.
- **ADDEN:** modo de detección de dirección, el cual se deshabilitará.
- **FERR:** error en recepción de datos.
- **OERR:** error de sobre flujo.
- **RX9D:** almacena el noveno bit para transmisión de 9 bits de datos.

**Figura 47.** Registro RCSTA y su configuración.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

1	0	0	1	0	X	X	0
b7	b6	b5	b4	b3	b2	b1	b0

El registro SPBRG nos permite configurar la velocidad de transferencia de la comunicación serial en kbps, se puede determinar el valor del registro en el manual de referencia del fabricante, que proporciona además información para poder configurar la velocidad a valores deseados por el programador. De acuerdo al fabricante el valor almacenado en el registro SPBRG para obtener una velocidad de 96 kbps debe ser 19H.

**Figura 48.** Configuración de registro SPBRG.

0	0	0	1	1	0	0	1
b7	b6	b5	b4	b3	b2	b1	b0

El registro TXREG almacena el dato a ser enviado por el puerto serial del módulo USART. Además, este registro habilita la transmisión datos seriales, cada vez que almacenemos un valor, se dará inicio a la transferencia.

El registro RCREG almacena el dato recibido por el puerto serial de módulo USART. Cada vez que se recibe un dato, la bandera RCIF se levanta

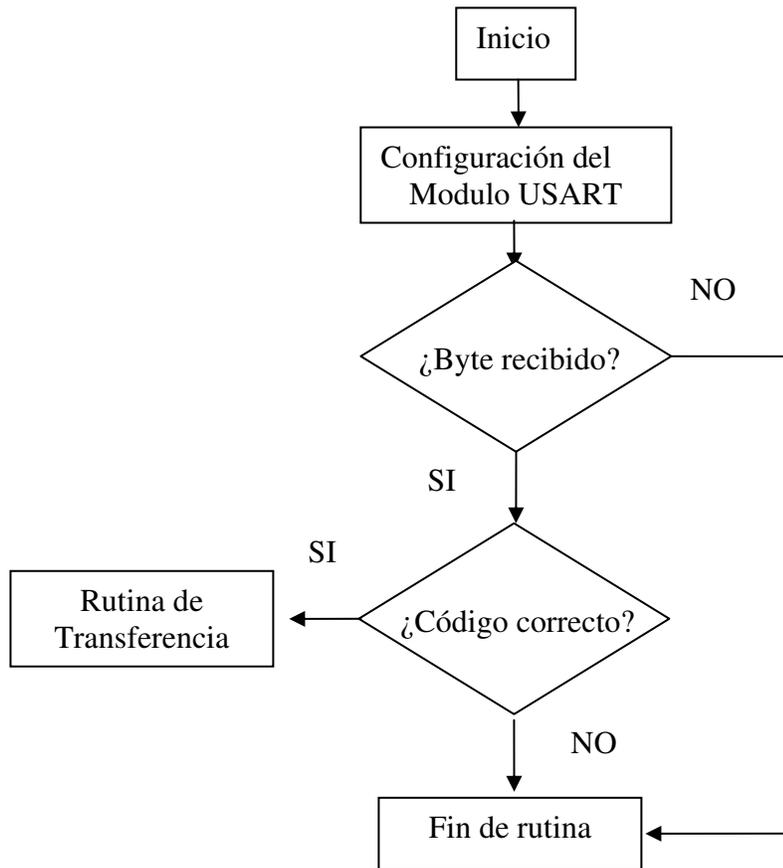
para indicar un dato recibido correctamente, en caso de una recepción errónea, se habilitan los indicadores OERR y FERR, los cuales debemos de borrar para poder continuar con la recepción de datos.

#### **4.8.2.4.2 Programa para la comunicación serial**

Para la transferencia de datos seriales, se utilizará un sencillo algoritmo basado en la detección de una petición de transferencia de datos. Cuando sea necesario descargar los datos almacenados en la memoria serial del anemógrafo en una computadora para su posterior análisis, se hará una petición de la computadora al microcontrolador, cuando la petición sea reconocida, se dará inicio a la transferencia de información del microcontrolador a la computadora, para finalizar la descarga el microcontrolador enviará una serie de datos de reconocimiento para finalizar la transmisión y borrará los datos contenidos en la memoria EEPROM y reiniciará los contadores de almacenamiento. Para controlar la descarga de la memoria EEPROM se utilizará un contador de memoria, que funciona tanto para el almacenamiento como para la descarga de datos.

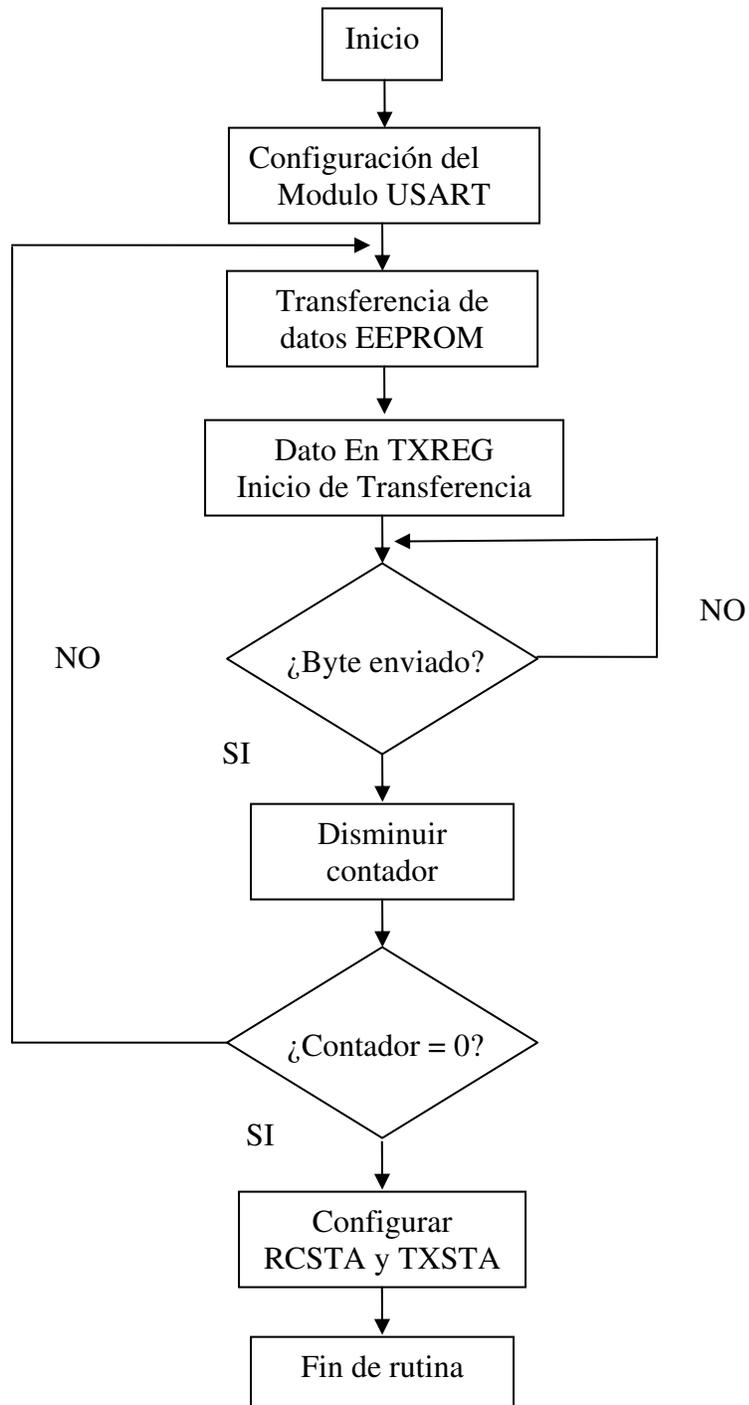
Para el control de la petición se utilizará una interrupción no enmascarable de hardware. Cada vez que un dato sea recibido en el puerto serial se genera una interrupción por recepción de información. Cada vez que se genere está interrupción, se llama a una rutina de verificación de petición de transferencia de información, si el byte recibido es igual al código de reconocimiento se inicia la transferencia, de lo contrario regresa al estado de espera.

**Figura 49.** Algoritmo de inicio de transferencia de información.



Después de ser reconocido el código de transferencia se debe de volver a configurar el registro TXSTA, para habilitar la transmisión de datos y en el caso del módulo RCSTA, se debe deshabilitar la recepción de datos. A finalizar la transmisión de datos, demos de deshabilitar la transmisión de datos y habilitar de nuevo la recepción continua de datos en espera de una siguiente petición de transferencia. Se enviará el código 5AH para indicar que se han enviado todos los datos y se esperará el código de reconocimiento 0F0H del computador para reiniciar la memoria serial EEPROM.

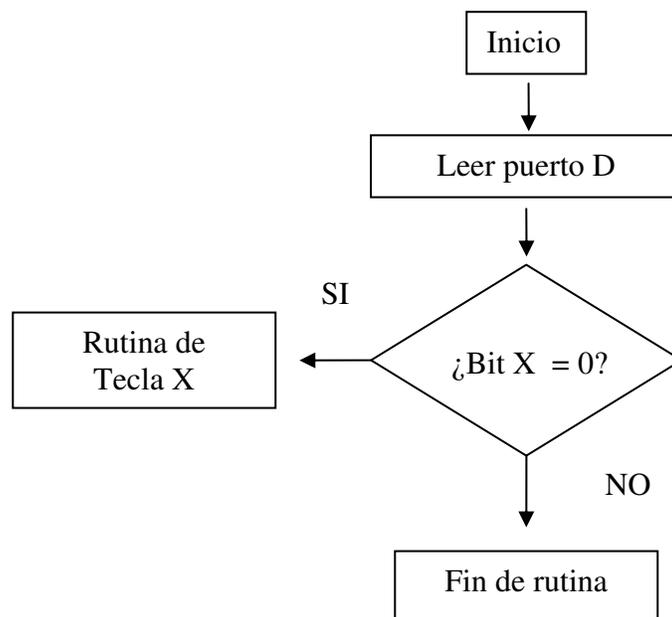
**Figura 50.** Algoritmo de transferencia de información.



#### 4.8.2.5 Programación de teclado

El programa del teclado se basa en la lectura periódica de los puertos en donde se encuentran conectadas cada una de las teclas, en cada ciclo del programa se verifica si hay cambio de estado en los puertos. Si existe cambio de estado, se llama a una rutina de acuerdo a la función de cada tecla. De acuerdo a la configuración de las teclas en el hardware, éstas mantienen un estado alto cuando no son pulsadas.

**Figura 51.** Algoritmo de funcionamiento del teclado



#### 4.8.2.6 Programación de temporización

Esta rutina se encargará de controlar el tiempo de muestreo del microcontrolador. Para esto se utilizará un temporizador interno del microcontrolador, el TIMER0, el cual es un registro de 16 bits que puede ser controlado por el reloj del microcontrolador o un reloj externo. Entre sus características resalta la capacidad de poder ser configurado como contador o como temporizador.

La ventaja de utilizar un temporizador en lugar de una rutina para el control del tiempo, se basa es que estos temporizadores funcionan en paralelo con cualquier otro proceso. Además la habilitación de interrupciones permite que el temporizador controle el tiempo de funcionamiento mediante una rutina que se efectuará de forma automática cada vez que termine su conteo, además posee la ventaja que al llegar a su conteo no se detiene, sino que continua de forma independiente a cualquier otro proceso que se este llevando a cabo.

El temporizador TIMER0 consta de varios registros que debemos de configurar para poder utilizarlo:

- **T0CON:** registro que permite configurar el funcionamiento del temporizador.
- **INTCON:** este registro de control de interrupciones posee los bits que permiten habilitar la interrupción del temporizador y la bandera de interrupción de éste.

- **TMR0H & TMR0L:** estos registros permiten configurar el valor inicial de conteo.

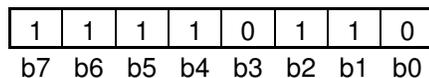
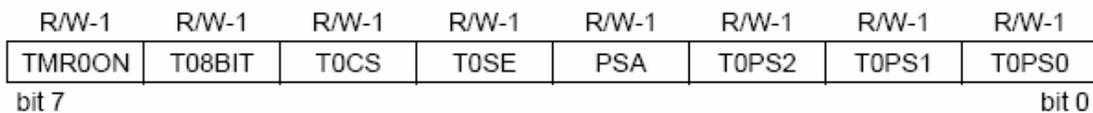
#### 4.8.2.6.1 Configuración de registro del temporizador TMR0

El registro T0CON, nos permite seleccionar la fuente del reloj, el funcionamiento a 16 bits, además de la relación entre el conteo y los pulsos de reloj. Cada uno de los bits posee la siguiente función:

- **TMR0ON:** este permite encender o apagar el contador. Este se encenderá desde el inicio del programa para llevar un control del tiempo.
- **T08BIT:** este permite configurar entre un contador de 8 o 16 bits. Para nuestro caso utilizaremos el contador de 8 bits, para lo cual se establece el valor de '1'.
- **TOCS:** permite seleccionar entre el reloj del microcontrolador y un reloj externo. Estableceremos el valor de '1' para una fuente de reloj externo de 32.768 Khz.
- **T0SE:** este bit permite seleccionar el flanco del reloj a partir del cual se realizará el conteo. Estableceremos el valor de '1' para un flanco negativo.
- **PSA:** el factor de escala del conteo del temporizador contra los pulsos de reloj. Permite configurar si habrá un factor de escala para el conteo o si el conteo se realizará cada pulso de reloj. Utilizaremos el factor de escala estableciendo el valor de '0'.

- **T0PS2, T0PS1 & T0PS0:** estos nos permiten configurar el factor de escala de acuerdo a valores preestablecidos por el fabricante. Para este caso utilizaremos el factor de escala de 128, que implica que se contará cada 128 pulsos del reloj externo. El valor a establecer será '110'. Se ha elegido este valor para que cada vez que el contador termine su conteo en cero y se genere la interrupción, el tiempo transcurrido sea exactamente de un segundo. Esto nos evita problemas de establecer un valor inicial de conteo.

**Figura 52.** Registro T0CON y su configuración.



Para el registro INTCON que configuramos anteriormente para el USART, únicamente debemos de prestar atención a los bits TMR0IE y TMR0IF. El bit TMR0IE es el bit que permite la habilitación de la interrupción del temporizador TIMER0 y el bit TMR0IF es el que nos indica que se ha terminado el conteo. En el caso del bit TMR0IF, este debe ser borrado cada vez que se llame a una interrupción por finalización de conteo en el temporizador, de lo contrario, el microcontrolador asumirá que existe una condición para mantener la interrupción.

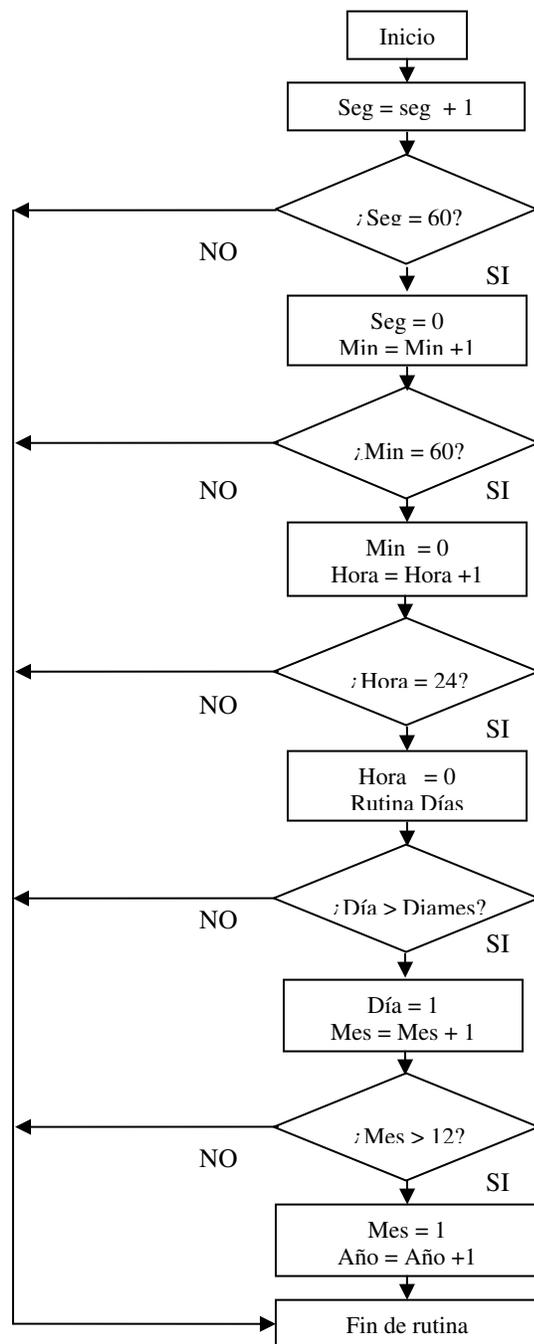
Los registros TMR0H y TMR0L son registros que permiten configurar el valor inicial de conteo para el temporizador. En este caso no será necesario configurarlos durante la ejecución de programa, únicamente serán borrados al inicio del programa para iniciar la cuenta desde 0.

#### **4.8.2.6.2 Programa para el temporizador**

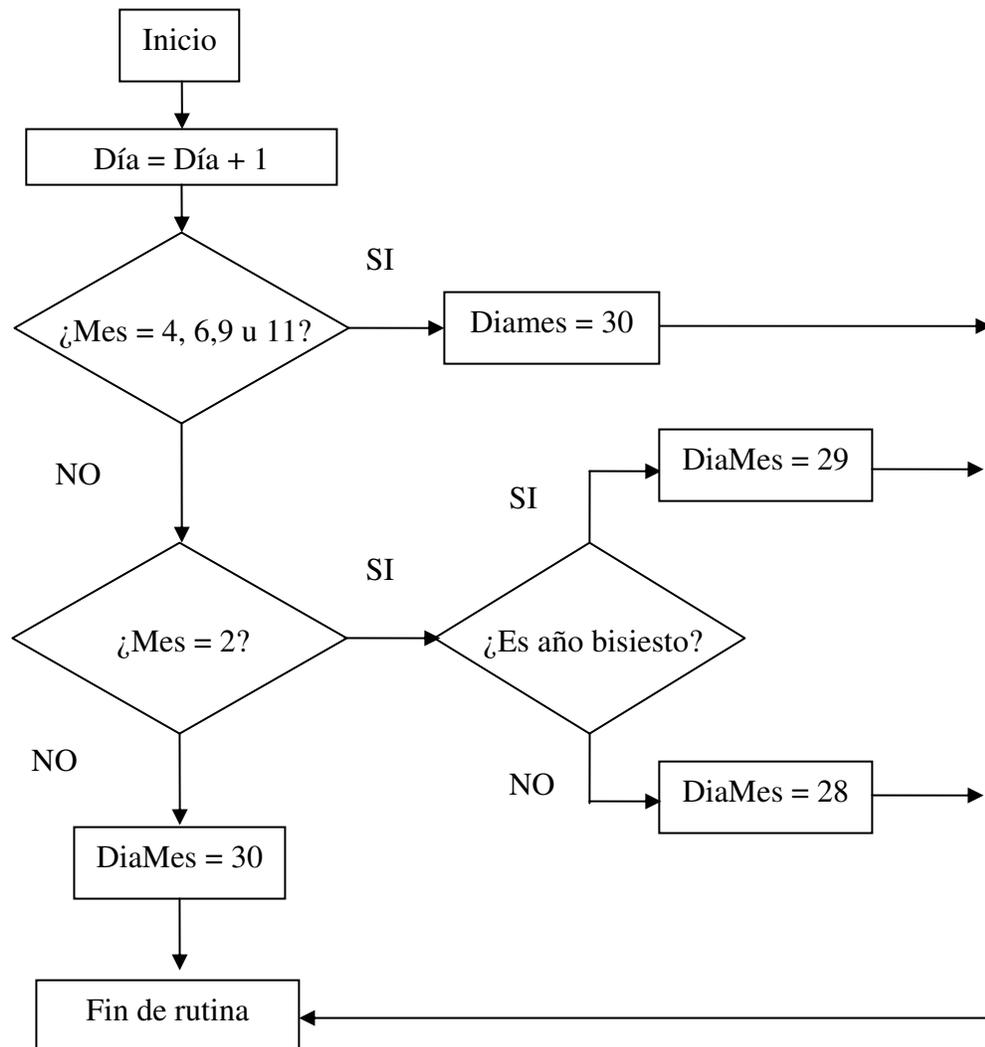
Este programa controla el tiempo de muestreo del anemógrafo, basado en una rutina de interrupción que se ejecuta exactamente cada segundo. La rutina de temporización esta controlada por el temporizador TMR0, que de acuerdo a la programación de conteo, genera una interrupción cada segundo. El programa debe de aumentar el contador de segundos y verificar si se ha llegado el conteo a 60, si esto ocurre, debe de borrar el contador y aumentar el contador de los minutos. Si el contador de minutos llega a sesenta, este se de borrar y aumentar el contador de las horas. El contador de horas, realizar un conteo de 24 horas, y luego aumentará el contador de los días.

Para el conteo de los días se lleva a cabo el uso de otra rutina encargada de verificar el mes en que se realiza el conteo, para determinar el número de días de ese mes. Los meses de Abril, Junio, Septiembre y Noviembre tienen 30 días, Febrero tiene 28 días y 29 días cuando el año es bisiesto y los demás meses tienen 31 días. Cuando el conteo ha sobrepasado el número máximo de días, se establece el contador en '1' y el contador de meses aumenta. Con el contador de meses es más sencillo, cada vez que el contador sobrepasa el valor de '12', el contador de meses se reestablece en '1' y aumentamos el contador de años.

Figura 53. Algoritmo de control de hora y fecha.



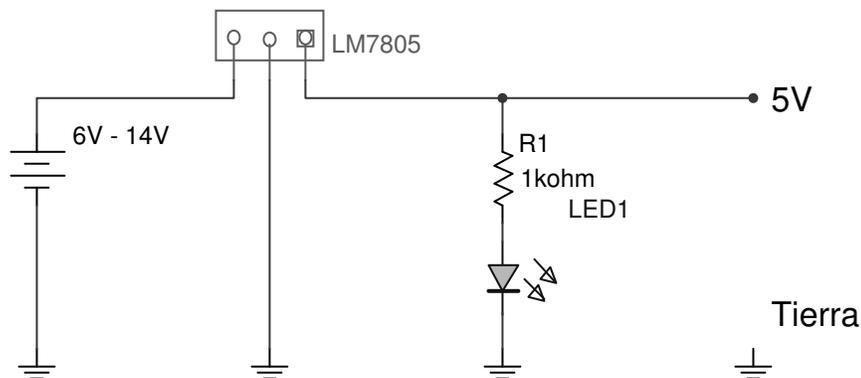
**Figura 54.** Algoritmo para determinar el número de días por mes.



## 4.9 Alimentación

Al momento, se dispone de todos los componentes electrónicos que conforman el diseño del anemógrafo, para determinar la fuente de alimentación ideal para el dispositivo. Todas las etapas necesitan como máximo una alimentación TTL, para lo cual hemos de utilizar una fuente de alimentación externa constituida por baterías, para permitir que el dispositivo trabaje de forma independiente. Pero el diseño puede fácilmente ser modificado para ser alimentado por 110 Voltios, en lugares donde este disponible el servicio eléctrico. Para esto se utilizará únicamente un transformador de 110V / 12V. Para el diseño se utilizará un regulador de voltaje LM7805, que permite regular 5 Voltios en su salida con una entrada que varíe desde 6 voltios hasta 14 voltios.

**Figura 55.** Diagrama esquemático del circuito de alimentación.



#### 4.10 Programa de Adquisición de datos

Para el diseño de este programa, utilizaremos programación orientada a objetos, por permitir una mayor facilidad para el diseño de programas, además de proporcionar interfaces amigables para el usuario. El lenguaje de programación a utilizar es Delphi, basado en pascal orientado a objetos. Este compilador nos proporciona herramientas de trabajo que permiten una mayor rapidez en la interfaz con dispositivos conectados a los puertos de la computadora. La herramienta a utilizar será Asyncfree, la cual nos proporciona control del puerto serial, para su configuración y utilización.

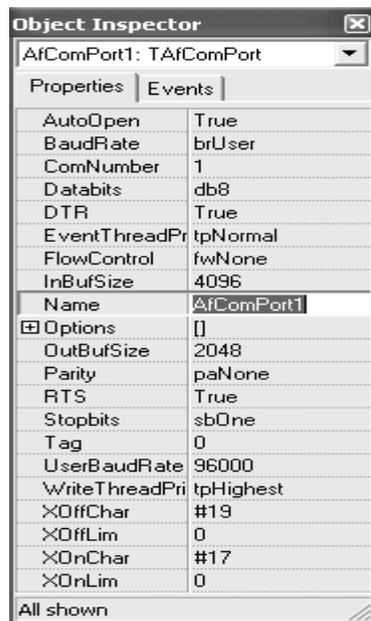
Las principales funciones del programa de adquisición e interpretación de datos son:

- Adquisición de datos almacenados en la memoria EEPROM del anemógrafo y almacenarlos en un archivo.
- Configurar la constantes del anemógrafo (Constante mecánica del transductor ( $K_i$ ), la constante del tacómetro ( $K_T$ ) y la de factor de escala).
- Procesar la información descargada del microcontrolador para determinar velocidades de viento en los períodos deseados.
- Graficar las diferentes velocidades del viento a través del tiempo de muestre.

### 4.10.1 Adquisición de datos

En el programa a diseñar únicamente debemos de configurar en la herramienta Asyncfree, la velocidad del puerto serial a 96 kbps, la cantidad de datos a 8 bit y el bit de parada a uno. Además debemos de indicar el puerto serial a utilizar, en este caso es el COM1. Esta herramienta del Asyncfree se llama AfComPort, el cual es un objeto que permite la configuración del puerto serial a configurar.

**Figura 56.** Configuración de puerto serial en AsyncFree



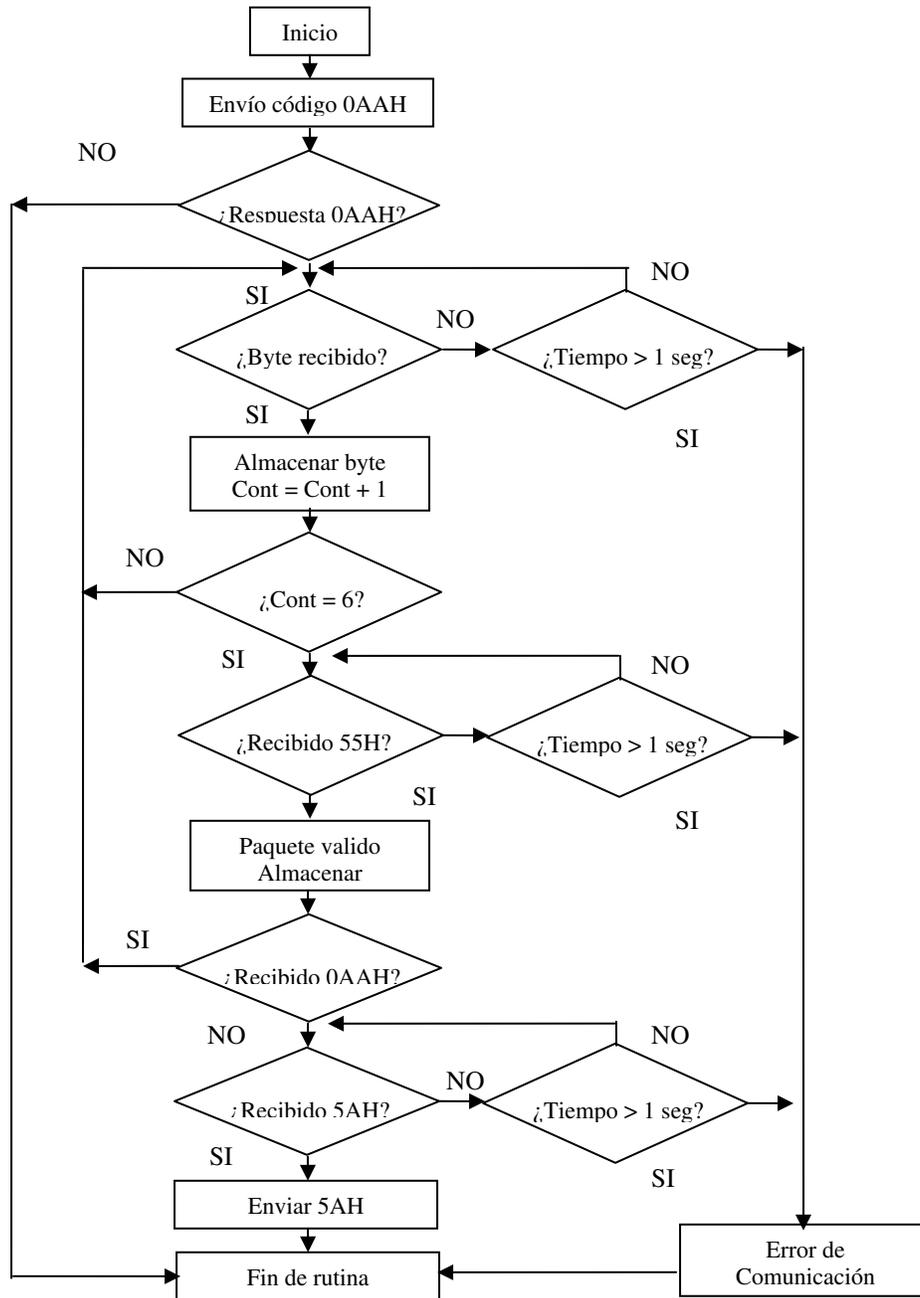
Para utilizar el puerto serial es necesario utilizar la herramienta llamada AfDataDispatcher, está herramienta nos permite controlar el puerto serial seleccionado. Con está herramienta es posible escribir datos al puerto serial y leerlos. Para la escritura de datos, es necesario utilizar un objeto para habilitar la escritura, en nuestro caso será un botón para petición de transferencia de datos. Para la lectura, utilizaremos el evento 'En dato recibido' del

AfDataDispatcher, para controlar la llegada de información proveniente del microcontrolador.

La rutina de adquisición de datos se basa en presionar el botón de transferencia de información, enviando el código 0AAH para que el microcontrolador inicie la transferencia de información hacia la computadora. Después se recibirá un dato cada vez que se active la interrupción de recepción de datos del puerto serial que ejecuta un pequeño ciclo para controlar la llegada de cada uno de las muestras del anemógrafo en paquetes de 8 bytes. Se iniciará con la detección del código de inicio de cada paquete que debe ser 0AAH, para poder decodificar cada paquete. Los seis bytes siguientes poseen la información de la fecha, hora y muestra. El último byte a esperar debe ser 55H, entonces se dará como válido el paquete recibido. El proceso continuará hasta que el microcontrolador envíe todos los datos de la memoria EEPROM. Para dar finalizada la comunicación enviará el código 5AH. Además el programa poseerá un tiempo de espera para la comunicación, en caso de existir algún problema con la conexión física. Se asignará un segundo como el tiempo máximo de espera entre bytes de datos, antes de dar finalizada la transferencia de datos. Al finalizar la transferencia, la computadora enviará el código de reconocimiento para 0F0H al microcontrolador, para que este considere la transferencia de datos correcta y reinicie la memoria EEPROM serial.

Para cada uno de los paquetes válidos, el programa almacenará en una tabla dividida en seis campos (columnas) cada uno de los datos. Este registro es un archivo tipo 'txt', ya que facilita la transferencia de los datos a diferentes programas para su posterior análisis. Cada uno de los campos está separado por el símbolo '|' para indicar datos de un mismo paquete, para dar inicio a otro campo se iniciará en la siguiente fila. Programas como Excel y MATLAB poseen la capacidad para poder convertir archivos 'txt' a tablas de datos.

**Figura 57.** Algoritmo de recepción de datos del microcontrolador.



**Figura 58.** Pantalla de inicio del programa de registro eólico.



En la figura anterior se muestra la pantalla inicial del programa de transferencia de información del microcontrolador a la computadora. El botón 'Transferencia' efectúa el algoritmo anterior para recuperar la información del anemógrafo. El botón 'Registro' nos permite manipular la información recuperada del dispositivo de medición.

#### **4.10.2 Configuración de constantes del anemógrafo**

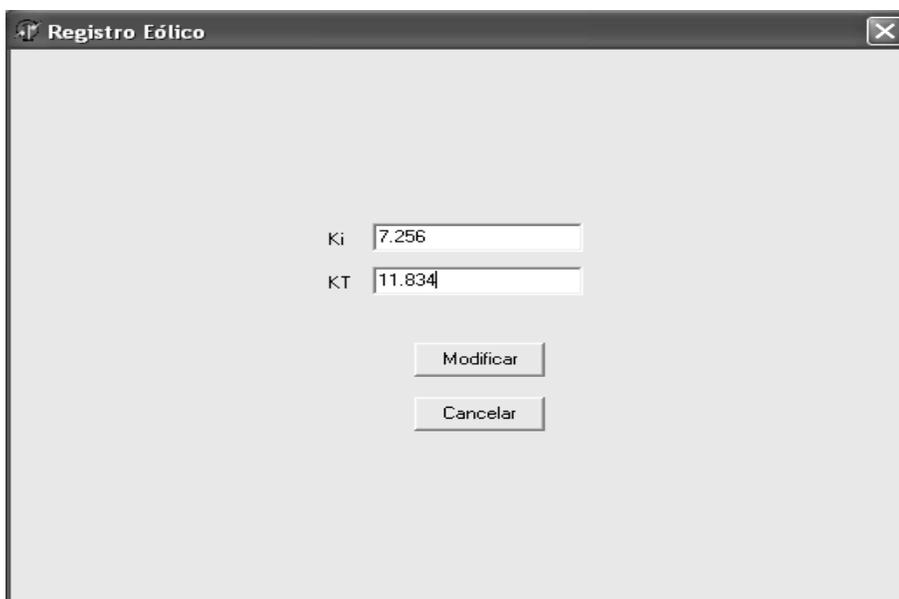
Para el análisis de los datos en la computadora es necesario configurar las constantes de proporcionalidad del transductor del anemógrafo. El objetivo de volver a configurarlas en el programa de análisis, reside en la cantidad de cantidades decimales que podemos operar con la computadora para realizar un mejor análisis. Para acceder a las constantes de proporcionalidad, debemos de pulsar el botón de registro, donde aparecerá un menú, que nos permitirá seleccionar entre graficar los datos registrados y cambiar las constantes de proporcionalidad del dispositivo.

**Figura 59.** Menú del programa de Registro Eólico.



Seleccionamos la opción 'Constantes', donde inicialmente aparecerá el valor actual de las constantes de  $K_i$  y  $K_T$ . Si deseamos mantener el valor actual solamente pulsamos el botón 'Cancelar' y regresaremos a la pantalla anterior. Para cambiar su valor, únicamente escribimos el valor deseado en cada una de las casillas de texto y pulsamos 'Modificar'.

**Figura 60.** Configuración de constantes del anemógrafo.



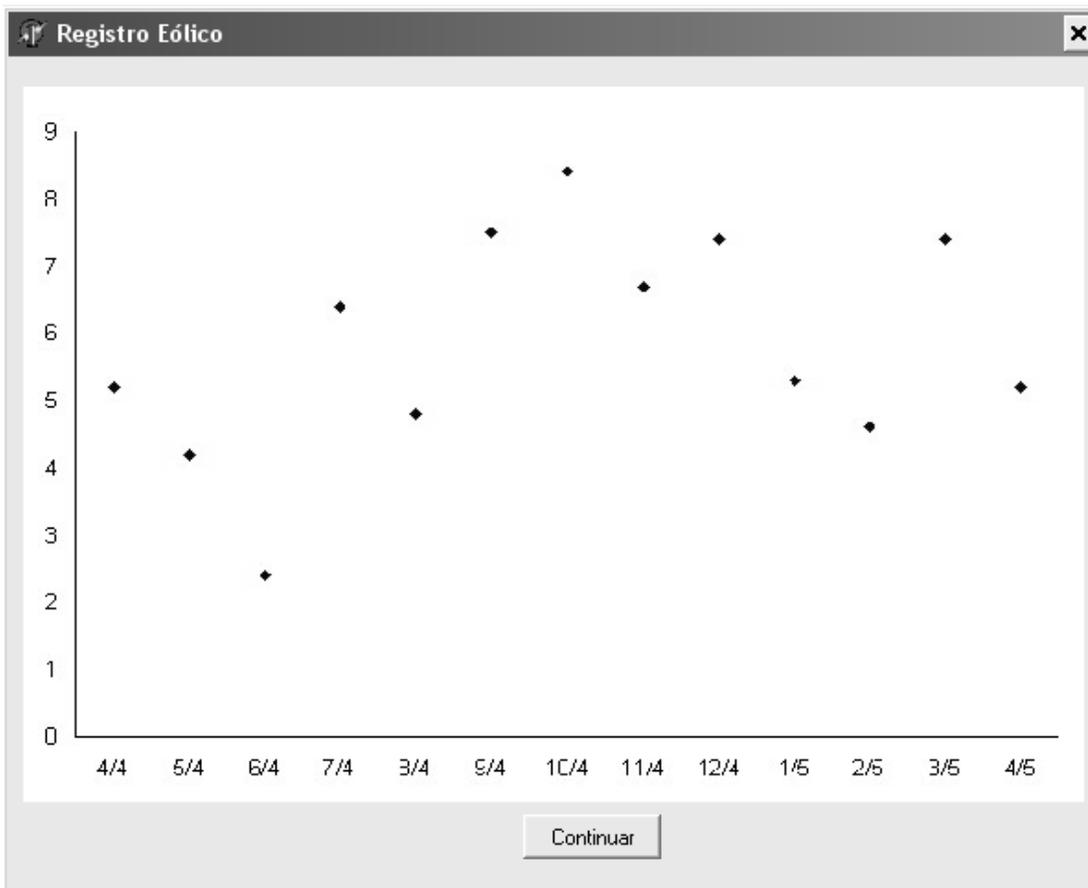
### 4.10.3 Análisis de registros

Para el análisis de la información se requiere determinar un período de evaluación, para determinar las velocidades promedios de esos períodos. En el menú, podemos seleccionar entre un período diario y un período mensual, luego seleccionamos graficar y obtenemos los valores promedios diarios o mensuales de la velocidad del viento. En esta ventana también podemos obtener información de cuantos días de información posee el dispositivo. La herramienta utilizada para graficar en Delphi es Canvas. Para un análisis más profundo se puede hacer uso del archivo de texto 'registro.txt' creado por el programa 'Registro Eólico', que se almacena en la misma carpeta donde se encuentre el archivo ejecutable del programa.

**Figura 61.** Menú de configuración de análisis de la información.



**Figura 62.** Gráfica mensual de velocidades promedio del viento.



En la figura anterior se muestra la gráfica generada de acuerdo a registros para velocidades promedio mensual. Para esta gráfica, se debe determinar la cantidad de meses disponibles en los datos registrados, a continuación, se establece un promedio diario para cada una de las muestras. Al obtener el promedio de las muestras diarias, se obtiene el promedio mensual en base al promedio de las anteriores.

#### **4.11 Funcionamiento del dispositivo**

El anemógrafo posee la principal característica de muestrear de acuerdo al período de tiempo programa por el usuario, de acuerdo al diseño de la memoria, el anemógrafo es capaz de almacenar hasta 64000 muestras de forma independiente. Si programamos un período de muestreo de 15 minutos, el anemógrafo podrá registrar datos hasta por 667 días. A mayor período de muestreo, mayor tiempo de operación.

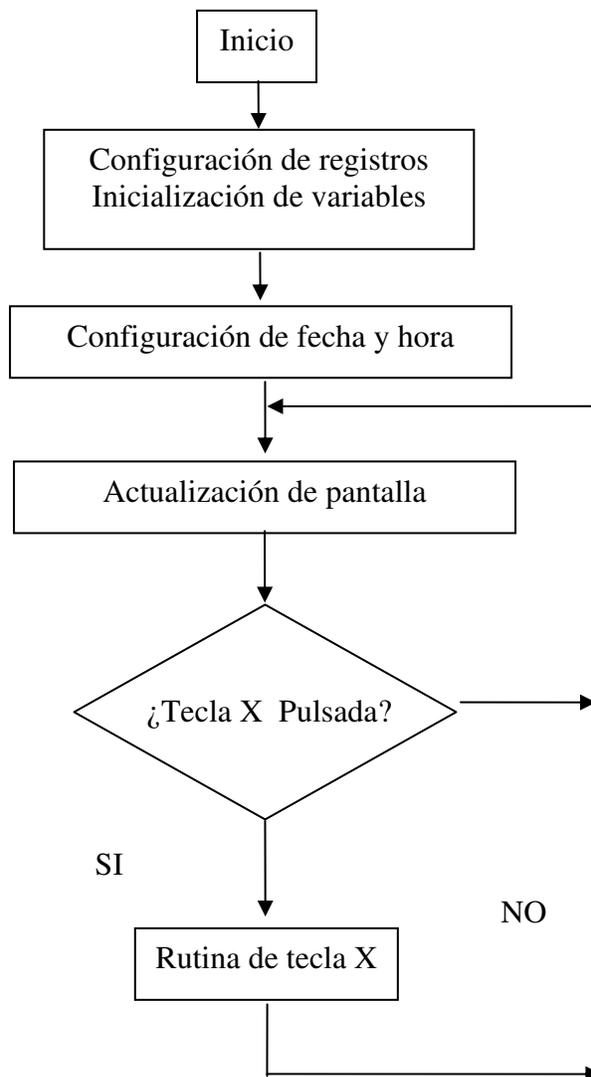
El anemógrafo toma muestras cada minuto y de acuerdo al período programado así serán la cantidad de datos almacenados por muestra, antes de obtener un valor promedio a almacenar. El dispositivo no almacena valores instantáneos de la velocidad del viento, sino valores promedio en un período de tiempo.

Al momento de energizar el dispositivo, se configurarán cada uno de los registros del microcontrolador de acuerdo a lo visto en el apartado anterior, luego se inicializarán las variables de control y se procederá a solicitar al usuario que actualice la fecha, hora, constantes de proporcionalidad y el período de muestreo. En el caso de las constantes de proporcionalidad el dispositivo estará configurado inicialmente a los valores de laboratorio. Para el período de muestreo, se ha elegido el de 15 minutos por muestras.

El algoritmo de control principal es bastante sencillo, ya que las principales funciones (muestreo, temporización y transferencia de datos) habilitarán una interrupción cada vez que se cumplan sus condiciones de funcionamiento, luego de inicializar el dispositivo, este se mantendrá anidado en un ciclo de muestreo de puertos, para determinar si el usuario ha presionado algunas de las teclas para acceder al menú de configuración del dispositivo.

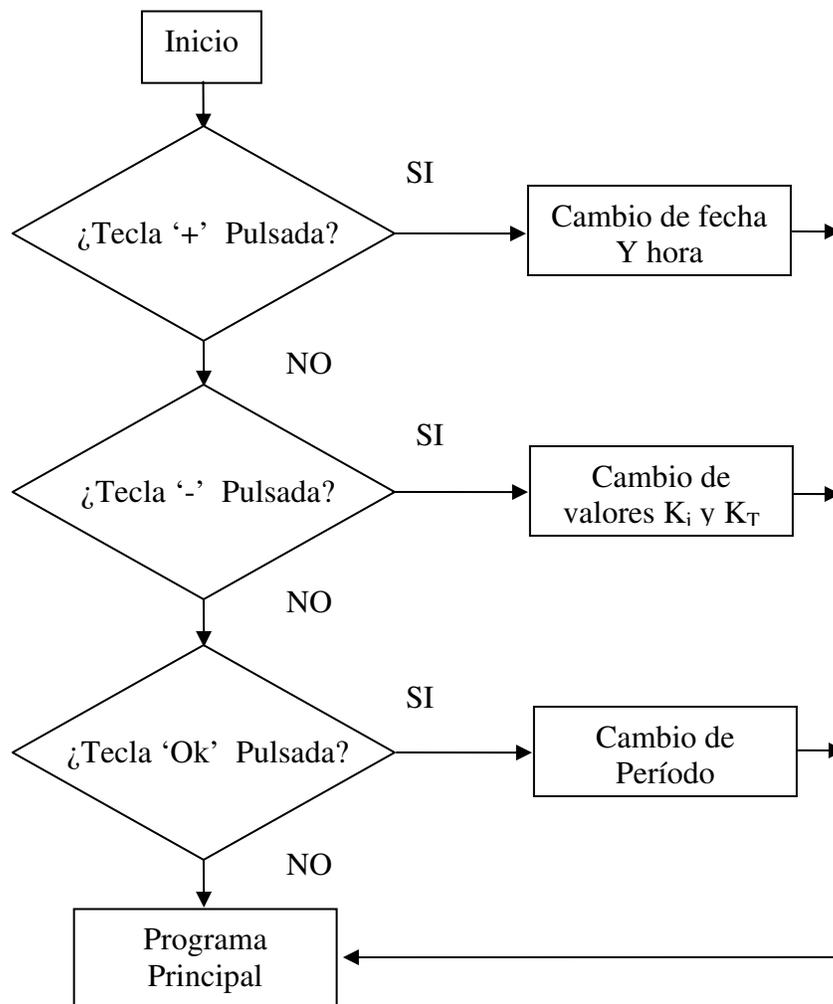
Para el caso de las interrupciones se consta de una rutina específica para cada uno de los procesos del microcontrolador.

**Figura 63.** Algoritmo de programa principal del microcontrolador.



Cada una de las teclas posee distinta función en el programa principal, a excepción de la tecla de 'Cancelar'. La tecla '+' nos permite ingresar a la subrutina para configurar la fecha, la tecla '-' permite configurar las constantes de proporcionalidad del dispositivo y la tecla 'Aceptar' accede a la subrutina para establecer el tiempo de muestreo. Al ingresar a cualquier subrutina, podemos salir sin realizar cambio alguno pulsando la tecla 'Cancelar'.

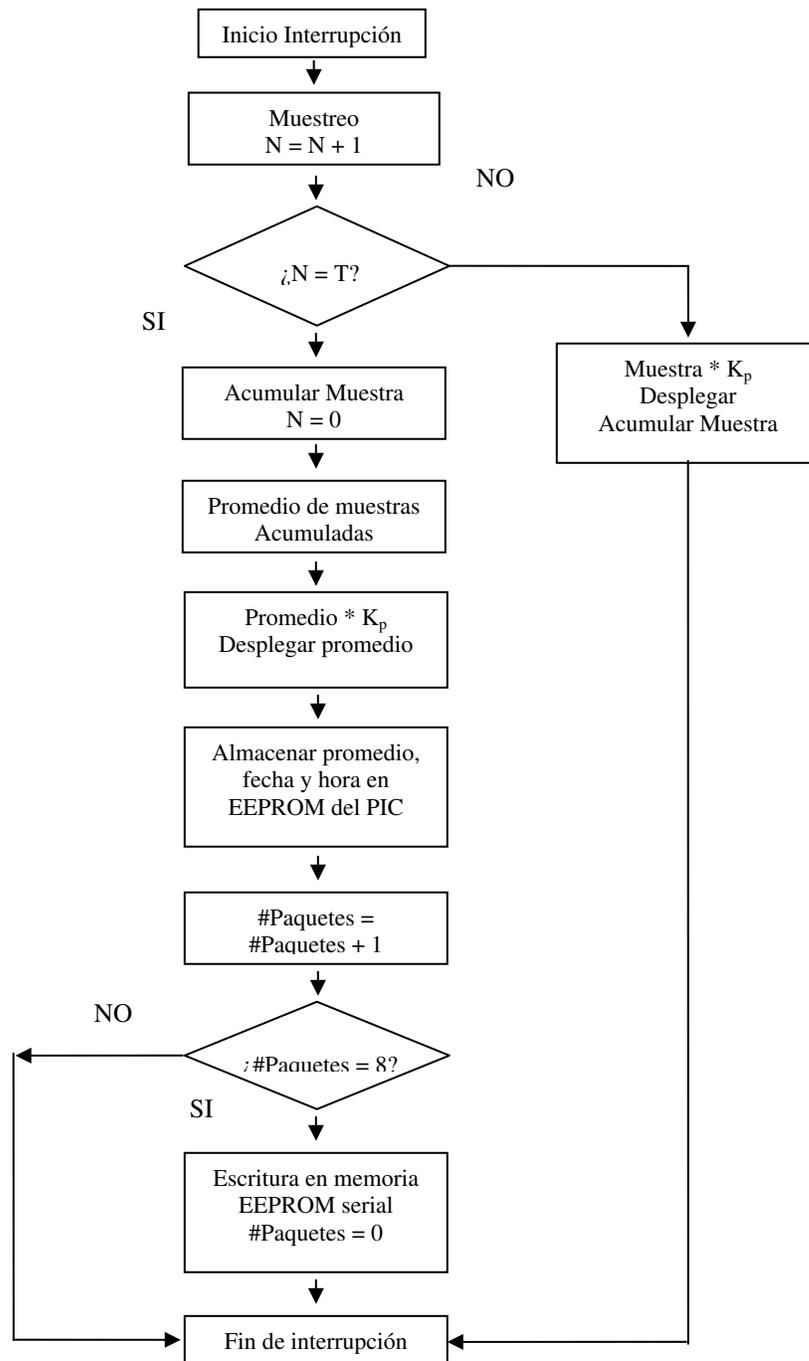
**Figura 64.** Algoritmo de funcionamiento del teclado.



La rutina de interrupción más importante es la generada por el temporizador TMR0, cada segundo para controlar la hora, fecha y el tiempo de muestreo del dispositivo. El muestreo es realizado cada minuto, pero estas muestras son acumuladas, para al terminar el período de muestreo, se promedian para obtener un valor medio de este intervalo de tiempo. En el caso del anemógrafo, este siempre desplegará el valor actual de la velocidad del viento y el valor promedio. El funcionamiento consiste en adquirir datos cada minuto y comparar el número de muestra contra el período, que también define el número máximo de muestras. Si no se alcanza el valor máximo de muestras, se acumula la muestra tomada en la memoria EEPROM del microcontrolador y se despliega, el valor registrado por ésta multiplicado por la constante de proporcionalidad del dispositivo. Al alcanzar el número de muestras, el acumulador se divide dentro del período y así obtenemos el valor promedio, el cual almacenamos, y luego aplicamos el factor de proporcionalidad del dispositivo y así obtenemos el valor promedio a desplegar durante el siguiente período de muestreo.

Para almacenar los paquetes de datos, se utilizará la memoria EEPROM del microcontrolador, en ésta se almacenarán como máximo 64 bytes, antes de enviarlos a la memoria serial EEPROM. Los valores de las constantes de proporcionalidad se almacenarán siempre en la memoria EEPROM, para evitar que al carecer de alimentación el microcontrolador, pierda sus valores de calibración.

Figura 65. Algoritmo de muestreo de datos y almacenamiento.





## CONCLUSIONES

1. Un transductor eólico-eléctrico se diseña basándose en los siguientes criterios:
  - Señal de salida (voltaje o corriente)
  - Rango de mediciones
  - Lugar de medición
  - Factibilidad de fabricación de componentes en el medio nacional.
  
2. Un microcontrolador PIC 18F458 se puede programar para:
  - Capturar datos analógicos generados por el transductor eólico-eléctrico y calcular la velocidad media del viento.
  - Hacer un muestreo periódico de la velocidad del viento, registrando fecha y hora de ésta.
  - Almacenar datos en memorias EEPROM seriales y transferir de la memoria hacia una computadora.
  - Configurar parámetros de funcionamiento del dispositivo (fecha, hora, constantes de proporcionalidad y período de muestreo).

3. Un microcontrolador reduce el tiempo de diseño y costos de instrumentos y otros dispositivos, al poseer módulos periféricos integrados con diversas aplicaciones.
4. El programa 'Registro Eólico' permite analizar la velocidad media del viento y graficar el comportamiento de ésta durante un período de muestreo programable, para determinar el potencial eólico de la región evaluada.
5. Para calibrar un instrumento de medición eólico es necesario determinar la constante de proporcionalidad del transductor eólico-eléctrico en una cámara de viento o utilizando un patrón disponible, y así determinar la confiabilidad de las mediciones.

## RECOMENDACIONES

1. La fabricación del transductor eólico-eléctrico disminuye el costo de producción del anemógrafo, pero aumenta el costo de la etapa de diseño y calibración.
2. Desarrollar algoritmos de los procesos desempeñados por el microcontrolador, reduce el tiempo de la etapa de programación.
3. Automatizar las estaciones de medición eólica del país, para disponer de registros para uso meteorológico y de análisis de generación de energía eólica.
4. Utilizar microcontroladores para reducir los diseños electrónicos y aumentar la confiabilidad de éstos.
5. Implementar un sistema inalámbrico para intercambio de datos, desde puntos remotos.



## REFERENCIA

- Jonson, Gary. **Wind Energy System**. Estados Unidos. 2001.



## BIBLIOGRAFÍA

1. CAMPOS CERDA, Manuel Fernando y otros. **Implementación de un sistema de desarrollo de los microcontroladores PIC Microchip Technology**. Ingeniería en Comunicaciones y Electrónica. México, Universidad de Guadalajara, Centro Universitario de Ciencias Exactas e Ingeniería. 1998.
2. DONN, William. **Meteorology**. 3ra. Ed., Ed. McGraw-Hill, Estados Unidos. 484 pp.
3. GIPE, Paul. **Wind power for home & business: Renewable energy for the 1990s and beyond**. Ed. Chelsea Green, Estados Unidos, 1993. 414 pp.
4. KUO, Benjamín. **Sistemas de control automático**. Ed. Prentice Hall Hispanoamericana, México, 1996. 897 pp.
5. Micro Engineering Labs: [www.melabs.com](http://www.melabs.com)
6. Microchip: [www.microchip.com](http://www.microchip.com)
7. MORALES MANCHAME, Edward. **El viento en Centro América**. Publicado por NRECA/CARES, Guatemala.

8. OCHOA GARCÍA, Humberto Alonzo. **Características a considerar en el diseño de una central eólica en Guatemala.** Tesis, Ing. Mecánico Electricista. Guatemala, Universidad de San Carlos Facultad de Ingeniería, 1998.
  
9. OLA GARCÍA, José Luis. **Elementos para analizar la factibilidad de instalar plantas eólicas en el sistema eléctrico de Guatemala.** Tesis, Ing. Electricista. Guatemala, Universidad de San Carlos Facultad de Ingeniería, 1998.
  
10. SOTO MIRANDA, Moisés Iván. **Diseño de un limnógrafo electrónico.** Tesis, Ing. Electrónico. Guatemala, Universidad de San Carlos Facultad de Ingeniería, 2003.
  
11. WALTER, John y Nicholas Jenkins. **Wind Energy Technology.** Ed. John Wiley & Sons, Inglaterra, 2000. 161 pp.

## APÉNDICE A

### Código fuente del programa del microcontrolador

```
;-----  
;  
;Programa de control de Anemógrafo digital  
;Utilizando PIC 18F458  
;Autor: Pedro Obregón  
;  
;  
list p=18f458 ;procesador PIC 18F458  
#include "p18f458.inc" ;referencia dirección de registros del PIC 18F458  
  
;Declaración de variables a utilizar en memoria de registros  
  
temp10 equ 0x00 ;variable utilizada por la rutina de espera 1  
temp11 equ 0x01 ;variable utilizada por la rutina de espera 1  
comando equ 0x02 ;variable utilizada para almacenar comando de LCD  
Dir_EEPROM equ 0x03 ;variable utilizada para almacenar dirección de EEPROM PIC  
Dato_EEPROM equ 0x04 ;variable utilizada para almacenar dato de EEPROM del PIC  
dir_LCD equ 0x05 ;dirección de la memoria DDRAM del LCD  
dato_LCD equ 0x06 ;carácter a imprimir en LCD  
dia equ 0x07 ;día  
dia_1 equ 0x08 ;unidad del día  
dia_2 equ 0x09 ;decena del día  
mes equ 0x0A ;mes  
mes_1 equ 0x0B ;unidad del mes  
mes_2 equ 0x0C ;decena del mes  
anno equ 0x0D ;año  
anno_1 equ 0x0E ;unidad del año  
anno_2 equ 0x0F ;decena del año  
hora equ 0x10 ;hora
```

hora_1	equ	0x11	;unidad de la hora
hora_2	equ	0x12	;decena de la hora
minuto	equ	0x13	;minutos
min_1	equ	0x14	;unidad de los minutos
min_2	equ	0x15	;decena de los minutos
segundo	equ	0x16	;segundos
dato_2	equ	0x17	;decena del dato a desplegar
dato_1	equ	0x18	;unidad del dato a desplegar
dato_0	equ	0x19	;decimal del dato a desplegar
ki_2	equ	0x1A	;decena de constante ki
ki_1	equ	0x1A	;unidad de constante ki
ki_0	equ	0x1B	;decimal de constante ki
kt_2	equ	0x1C	;decena de constante kt
kt_1	equ	0x1D	;unidad de constante kt
kt_0	equ	0x1E	;decimal de constante kt
ki	equ	0x1F	;constante ki
kt	equ	0x20	;constante kt
T_3	equ	0x21	;centena de período de muestreo
T_2	equ	0x22	;decena de período de muestreo
T_1	equ	0x23	;unidad de período de muestreo
T	equ	0x24	;período de muestreo
temp_0	equ	0x25	;variable de almacenamiento temporal
temp_1	equ	0x26	;variable de almacenamiento temporal
temp_2	equ	0x27	;variable de almacenamiento temporal
temp_3	equ	0x28	;variable de almacenamiento temporal
temp_4	equ	0x29	;variable de almacenamiento temporal
temp_5	equ	0x2A	;variable de almacenamiento temporal
temp_6	equ	0x2B	;variable de almacenamiento temporal
temp_7	equ	0x2C	;variable de almacenamiento temporal
temp_8	equ	0x2D	;variable de almacenamiento temporal
temp_9	equ	0x2E	;variable de almacenamiento temporal
temp_10	equ	0x2F	;variable de almacenamiento temporal
diames	equ	0x30	;dias en un mes
ki_dec	equ	0x31	;decimal de la constante ki

```

kt_dec      equ    0x32    ;decimal de la constante kt
n           equ    0x33    ;número muestras tomadas
m_2        equ    0x34    ;byte alto de muestra acumulada
m_1        equ    0x35    ;byte bajo de muestra acumulada
muestra     equ    0x36    ;muestra a desplegar
muestra_m  equ    0x37    ;muestra promedio
mserial     equ    0x38    ;memoria serial EEPROM seleccionada
DHEEPROM   equ    0x39    ;byte más significativo de dirección EEPROM serial
DLEEPROM   equ    0x3A    ;byte menos significativo de dirección EEPROM serial
D_SEEPROM  equ    0x3B    ;dato de la memoria serial EEPROM
k           equ    0x3C    ;constante de proporcionalidad
datum      equ    0x3D    ;registro temporal de la interfaz serial

```

```

;-----

```

```

;-----

```

```

;Inicio del programa del anemógrafo digital

```

```

    org 0x000

```

```

    goto  Rutina_principal ; salto a rutina principal

```

```

    org 0x008      ;vector de interrupción

```

```

    goto  interrupciones

```

```

    org 0x020

```

```

;A partir de esta localidad de memoria inicia el programa

```

```

;-----

```

```

;-----

```

```

;Espera de tiempo

```

```

;Espera equivalente a 2 ms.

```

```

Espera1

```

```

    movlw .21

```

```

    movwf temp10

```

```

Esp11  movlw .237

```

```

    movwf temp11

```

```

Esp12

```

```

    nop
    decfsz temp11,1
    goto Esp12
    decfsz temp10,1
    goto Esp11
    goto Esp13
Esp13 nop
    return
;-----
;-----
;Rutinas para control y configuración de LCD

;Envío de comando a LCD
comando_LCD
    bcf    PORTD,1
    bcf    PORTD,2
    nop                                ;espera mínima de un 0.2 us
    nop
    bsf    PORTD,0
    nop
    movff  comando,PORTB
    movwf  LATB
    nop
    bcf    PORTD,0
    nop
    nop
    call   Espera1    ;tiempo de espera de 2ms. para ejecución de comando
    return

```

;Escritura de datos en LCD

Escribir\_LDC

```
    movlw 0x80           ;dirección de acceso a DDRAM
    addwf dir_LCD,0
    movwf comando
    call comando_LCD
    bcf PORTD,1
    bsf PORTD,2
    nop                 ;espera mínima de un 0.2 us
    nop
    bsf PORTD,0
    nop
    movff dato_LCD,PORTB
    movwf LATB
    nop
    bcf PORTD,0
    nop
    nop
    bcf PORTD,2
    call Espera1       ;tiempo de espera de 2ms. para ejecución de comando
    return
```

;Despliegue fecha y hora

Desplegar\_fecha\_hora

```
    movlw 0x01
    movwf dir_LCD
    movlw dia_2
    movwf dato_LCD
    call Escribir_LDC
    movlw 0x02
```

```
movwf dir_LCD
movlw día_1
movwf dato_LCD
call Escribir_LDC
movlw 0x03
movwf dir_LCD
movlw 0x2F
movwf dato_LCD
call Escribir_LDC
movlw 0x04
movwf dir_LCD
movlw mes_2
movwf dato_LCD
call Escribir_LDC
movlw 0x05
movwf dir_LCD
movlw mes_1
movwf dato_LCD
call Escribir_LDC
movlw 0x06
movwf dir_LCD
movlw 0x2F
movwf dato_LCD
call Escribir_LDC
movlw 0x07
movwf dir_LCD
movlw anno_2
movwf dato_LCD
call Escribir_LDC
movlw 0x08
movwf dir_LCD
movlw anno_1
movwf dato_LCD
call Escribir_LDC
```

```
movlw 0x0A
movwf dir_LCD
movlw hora_2
movwf dato_LCD
call Escribir_LDC
movlw 0x0B
movwf dir_LCD
movlw hora_1
movwf dato_LCD
call Escribir_LDC
movlw 0x0C
movwf dir_LCD
movlw 0x2F
movwf dato_LCD
call Escribir_LDC
movlw 0x0D
movwf dir_LCD
movlw min_2
movwf dato_LCD
call Escribir_LDC
movlw 0x0E
movwf dir_LCD
movlw min_1
movwf dato_LCD
call Escribir_LDC
return
```

;Despliegue de datos

Desplegar\_datos

```
movlw 0x41
movwf dir_LCD
movlw 0x56
```

```
movwf dato_LCD
call  Escribir_LDC
movlw 0x43
movwf dir_LCD
movlw 0x4D
movwf dato_LCD
call  Escribir_LDC
movlw 0x44
movwf dir_LCD
movlw 0x45
movwf dato_LCD
call  Escribir_LDC
movlw 0x45
movwf dir_LCD
movlw 0x44
movwf dato_LCD
call  Escribir_LDC
movlw 0x47
movwf dir_LCD
movlw dato_2
movwf dato_LCD
call  Escribir_LDC
movlw 0x48
movwf dir_LCD
movlw dato_1
movwf dato_LCD
call  Escribir_LDC
movlw 0x49
movwf dir_LCD
movlw 0x2E
movwf dato_LCD
call  Escribir_LDC
movlw 0x4A
movwf dir_LCD
```

```
movlw dato_0
movwf dato_LCD
call  Escribir_LDC
movlw 0x4C
movwf dir_LCD
movlw 0x4D
movwf dato_LCD
call  Escribir_LDC
movlw 0x4D
movwf dir_LCD
movlw 0x2F
movwf dato_LCD
call  Escribir_LDC
movlw 0x4E
movwf dir_LCD
movlw 0x53
movwf dato_LCD
call  Escribir_LDC
return
```

;Cambio de constante

Constante

```
movlw 0x04
movwf dir_LCD
movlw 0x4B
movwf dato_LCD
call  Escribir_LDC
movlw 0x05
movwf dir_LCD
movlw 0x69
movwf dato_LCD
```

```
call    Escribir_LDC
movlw  0x06
movwf  dir_LCD
movlw  0x3D
movwf  dato_LCD
call    Escribir_LDC
movlw  0x08
movwf  dir_LCD
movlw  ki_2
movwf  dato_LCD
call    Escribir_LDC
movlw  0x09
movwf  dir_LCD
movlw  ki_1
movwf  dato_LCD
call    Escribir_LDC
movlw  0x0A
movwf  dir_LCD
movlw  0x2E
movwf  dato_LCD
call    Escribir_LDC
movlw  0x0B
movwf  dir_LCD
movlw  ki_0
movwf  dato_LCD
call    Escribir_LDC
movlw  0x44
movwf  dir_LCD
movlw  0x4B
movwf  dato_LCD
call    Escribir_LDC
movlw  0x45
movwf  dir_LCD
movlw  0x74
```

```
movwf dato_LCD
call  Escribir_LDC
movlw 0x46
movwf dir_LCD
movlw 0x3D
movwf dato_LCD
call  Escribir_LDC
movlw 0x48
movwf dir_LCD
movlw kt_2
movwf dato_LCD
call  Escribir_LDC
movlw 0x49
movwf dir_LCD
movlw kt_1
movwf dato_LCD
call  Escribir_LDC
movlw 0x4A
movwf dir_LCD
movlw 0x2E
movwf dato_LCD
call  Escribir_LDC
movlw 0x4B
movwf dir_LCD
movlw kt_0
movwf dato_LCD
call  Escribir_LDC
return
```

;Cambio de Período

Período

```
    movlw 0x01
    movwf dir_LCD
    movlw 0x54
    movwf dato_LCD
    movlw 0x02
    movwf dir_LCD
    movlw 0x3D
    movwf dato_LCD
    movlw 0x04
    movwf dir_LCD
    movlw T_3
    movwf dato_LCD
    movlw 0x05
    movwf dir_LCD
    movlw T_2
    movwf dato_LCD
    movlw 0x06
    movwf dir_LCD
    movlw T_1
    movwf dato_LCD
    movlw 0x08
    movwf dir_LCD
    movlw 0x4D
    movwf dato_LCD
    movlw 0x09
    movwf dir_LCD
    movlw 0x49
    movwf dato_LCD
    movlw 0x0A
    movwf dir_LCD
    movlw 0x4E
```

```
movwf dato_LCD
movlw 0x0B
movwf dir_LCD
movlw 0x55
movwf dato_LCD
movlw 0x0C
movwf dir_LCD
movlw 0x54
movwf dato_LCD
movlw 0x0D
movwf dir_LCD
movlw 0x4F
movwf dato_LCD
movlw 0x0E
movwf dir_LCD
movlw 0x53
movwf dato_LCD
return
```

;transferencia en progreso

transferencia

```
movlw 0x00
movwf dir_LCD
movlw 0x54
movwf dato_LCD
call Escribir_LDC
movlw 0x01
movwf dir_LCD
movlw 0x54
```

```
movwf dato_LCD
call  Escribir_LDC
movlw 0x02
movwf dir_LCD
movlw 0x41
movwf dato_LCD
call  Escribir_LDC
movlw 0x03
movwf dir_LCD
movlw 0x4E
movwf dato_LCD
call  Escribir_LDC
movlw 0x04
movwf dir_LCD
movlw 0x53
movwf dato_LCD
call  Escribir_LDC
movlw 0x05
movwf dir_LCD
movlw 0x46
movwf dato_LCD
call  Escribir_LDC
movlw 0x06
movwf dir_LCD
movlw 0x45
movwf dato_LCD
call  Escribir_LDC
movlw 0x07
movwf dir_LCD
movlw 0x52
movwf dato_LCD
call  Escribir_LDC
movlw 0x08
movwf dir_LCD
```

```
movlw 0x45
movwf dato_LCD
call  Escribir_LDC
movlw 0x09
movwf dir_LCD
movlw 0x4E
movwf dato_LCD
call  Escribir_LDC
movlw 0x0A
movwf dir_LCD
movlw 0x43
movwf dato_LCD
call  Escribir_LDC
movlw 0x0B
movwf dir_LCD
movlw 0x49
movwf dato_LCD
call  Escribir_LDC
movlw 0x0C
movwf dir_LCD
movlw 0x41
movwf dato_LCD
call  Escribir_LDC
movlw 0x0E
movwf dir_LCD
movlw 0x45
movwf dato_LCD
call  Escribir_LDC
movlw 0x0F
movwf dir_LCD
movlw 0x4E
movwf dato_LCD
call  Escribir_LDC
movlw 0x43
```

```
movwf dir_LCD
movlw 0x50
movwf dato_LCD
call  Escribir_LDC
movlw 0x44
movwf dir_LCD
movlw 0x52
movwf dato_LCD
call  Escribir_LDC
movlw 0x45
movwf dir_LCD
movlw 0x4F
movwf dato_LCD
call  Escribir_LDC
movlw 0x46
movwf dir_LCD
movlw 0x47
movwf dato_LCD
call  Escribir_LDC
movlw 0x47
movwf dir_LCD
movlw 0x52
movwf dato_LCD
call  Escribir_LDC
movlw 0x48
movwf dir_LCD
movlw 0x45
movwf dato_LCD
call  Escribir_LDC
movlw 0x49
movwf dir_LCD
movlw 0x53
movwf dato_LCD
call  Escribir_LDC
```

```

movlw 0x4A
movwf dir_LCD
movlw 0x4F
movwf dato_LCD
call  Escribir_LDC
movlw 0x4B
movwf dir_LCD
movlw 0x2E
movwf dato_LCD
call  Escribir_LDC
movlw 0x4C
movwf dir_LCD
movlw 0x2E
movwf dato_LCD
call  Escribir_LDC
return

```

```
;------
```

```
;------
```

```
;Rutinas de lectura y escritura en memoria EEPROM del PIC
```

```
Leer_EEPROM_PIC
```

```
;rutina para leer la memoria EEPROM del PIC
```

```

movff  Dir_EEPROM,EEADR

bcf    EECON1, EEPGD
bcf    EECON1, CFGS
bsf    EECON1, RD
movff  EEDATA, Dato_EEPROM
return

```

```
Escribir_EEPROM_PIC
```

```
;rutina para escribir en la memoria EEPROM
```

```

movff  Dir_EEPROM,EEADR

```

```

movff Dato_EEPROM,EEDATA
bcf   EECON1, EEPGD
bcf   EECON1, CFGS
bsf   EECON1, WREN
bcf   INTCON, GIE
movlw 0x55
movwf EECON2
movlw 0x0AA
movwf EECON2
bsf   EECON1, WR
bsf   INTCON, GIE
Escritura_EEPROM_En_Progreso
btfss PIR2, EEIF
goto  Escritura_EEPROM_En_Progreso
bcf   PIR2, EEIF
bcf   EECON1, WREN
return

```

```

;-----
;-----

```

;Determinar días en un mes

dia\_mes

```

movlw .30
movwf temp_0
movlw .4
cpfseq mes
goto  m1
goto  salir_diames
m1   movlw .6
      cpfseq mes
      goto  m2
      goto  salir_diames
m2   movlw .9

```

```

        cpfseq mes
        goto m3
        goto salir_diames
m3     movlw .11
        cpfseq mes
        goto m4
        goto salir_diames

m4     movff anno,temp_1
rep    movff temp_1,temp_2
        movlw .4
        subwfb temp_1,1
        btfs STATUS,C
        goto rep
        clrf WREG
        cpfseq temp_2
        goto m5
        movlw .29
        movwf temp_0
        goto m6
m5     movlw .28
        movwf temp_0
m6     movlw .2
        cpfseq mes
        goto m7
        goto salir_diames
m7     movlw .31
        movwf temp_0
salir_diames
        movff temp_0,diames
        return

```

;-----

;rutina de control de hora y fecha

```
fecha_hora
    incf    segundo,1
    movlw  .60
    cpfseq segundo
    goto   salir_rutina
    clrf   segundo
    incf   minuto,1
    movlw  .60
    cpfseq minuto
    goto   salir_rutina
    clrf   minuto
    incf   hora,1
    movlw  .24
    cpfseq hora
    goto   salir_rutina
    clrf   hora
    incf   dia
    call   dia_mes
    movlw  diames
    cpfseq dia
    goto   salir_rutina
    movlw  .1
    movwf  dia
    incf   mes,1
    movlw  .12
    cpfseq mes
    goto   salir_rutina
    movlw  .1
    movwf  mes
    incf   anno,1
```

;En esta parte de la rutina se almacena las variables en la memoria EEPROM del PIC

;Se le asigna un valor a cada una de las variables a desplegar en la pantalla

```

salir_rutina
    movlw 0x40
    movwf Dir_EEPROM
    movff minuto,Dato_EEPROM
    call  Escribir_EEPROM_PIC
    movlw 0x41
    movwf Dir_EEPROM
    movff hora,Dato_EEPROM
    call  Escribir_EEPROM_PIC
    movlw 0x42
    movwf Dir_EEPROM
    movff dia,Dato_EEPROM
    call  Escribir_EEPROM_PIC
    movlw 0x43
    movwf Dir_EEPROM
    movff mes,Dato_EEPROM
    call  Escribir_EEPROM_PIC
    movlw 0x44
    movwf Dir_EEPROM
    movff anno,Dato_EEPROM
    call  Escribir_EEPROM_PIC

    movff minuto,temp_0
    clrf  temp_1
asignar_minuto
    incf  temp_1,1
    movlw .10
    movff temp_0,temp_2
    subwfb temp_0,1
    btfss STATUS,C
    goto  asignar_minuto
    movff temp_2,min_1
    movff temp_1,min_2

```

```

        movff  hora,temp_0
        clrf   temp_1
asignar_hora
        incf   temp_1,1
        movlw  .10
        movff  temp_0,temp_2
        subwfb temp_0,1
        btfss  STATUS,C
        goto   asignar_hora
        movff  temp_2,hora_1
        movff  temp_1,hora_2

```

```

        movff  dia,temp_0
        clrf   temp_1
asignar_dia
        incf   temp_1,1
        movlw  .10
        movff  temp_0,temp_2
        subwfb temp_0,1
        btfss  STATUS,C
        goto   asignar_dia
        movff  temp_2,dia_1
        movff  temp_1,dia_2

```

```

        movff  mes,temp_0
        clrf   temp_1
asignar_mes
        incf   temp_1,1
        movlw  .10
        movff  temp_0,temp_2
        subwfb temp_0,1
        btfss  STATUS,C
        goto   asignar_mes

```

```

        movff  temp_2,mes_1
        movff  temp_1,mes_2

        movff  anno,temp_0
        clrf   temp_1
asignar_anno
        incf   temp_1,1
        movlw  .10
        movff  temp_0,temp_2
        subwfb temp_0,1
        btfss  STATUS,C
        goto   asignar_anno
        movff  temp_2,anno_1
        movff  temp_1,anno_2
        return

;-----
;-----
;Rutinas para cambiar el valor de las variables con el teclado
inc_dec.      ;Esta rutina incrementa el valor de la variable
        incf   temp_0,1
        movlw  .10
        cpfseq temp_0
        goto   salir_inc
        clrf   temp_0
salir_inc
        return

dec_dec       ;Esta rutina disminuye el valor de la variable.
        decf   temp_0,1
        btfss  STATUS,C
        goto   salir_dec
        movlw  .9
        movwf  temp_0

```

salir\_dec

return

cambio\_valor ;Control de teclado en cambio de valor de variables.

btfs PORTD,4

goto inc\_dec

btfs PORTD,5

goto dec\_dec

btfs PORTD,6

goto salir\_valor

btfs PORTD,7

goto cambio\_valor

salir\_valor

return

;------

;------

;Rutina de cambio de fecha y hora.

cambio\_fecha\_hora

call Desplegar\_fecha\_hora

ff1

call cambio\_valor

movlw .1

movwf dir\_LCD

movff temp\_0,temp\_1

movff temp\_0,dato\_LCD

call Escribir\_LDC

btfs PORTD,7

goto salir\_fecha\_hora

btfs PORTD,6

goto ff1

ff2

```
call    cambio_valor
movlw  .2
movwf  dir_LCD
movff  temp_0,temp_2
movff  temp_0,dato_LCD
call   Escribir_LDC
btfsc  PORTD,7
goto   salir_fecha_hora
btfsc  PORTD,6
goto   ff2
```

ff3

```
call    cambio_valor
movlw  .4
movwf  dir_LCD
movff  temp_0,temp_3
movff  temp_0,dato_LCD
call   Escribir_LDC
btfsc  PORTD,7
goto   salir_fecha_hora
btfsc  PORTD,6
goto   ff3
```

ff4

```
call    cambio_valor
movlw  .5
movwf  dir_LCD
movff  temp_0,temp_4
movff  temp_0,dato_LCD
call   Escribir_LDC
btfsc  PORTD,7
goto   salir_fecha_hora
btfsc  PORTD,6
goto   ff4
```

ff5

```
call    cambio_valor
movlw  .7
movwf  dir_LCD
movff  temp_0,temp_5
movff  temp_0,dato_LCD
call   Escribir_LDC
btsc   PORTD,7
goto   salir_fecha_hora
btsc   PORTD,6
goto   ff5
```

ff6

```
call    cambio_valor
movlw  .8
movwf  dir_LCD
movff  temp_0,temp_6
movff  temp_0,dato_LCD
call   Escribir_LDC
btsc   PORTD,7
goto   salir_fecha_hora
btsc   PORTD,6
goto   ff6
```

ff7

```
call    cambio_valor
movlw  0x0A
movwf  dir_LCD
movff  temp_0,temp_7
movff  temp_0,dato_LCD
call   Escribir_LDC
btsc   PORTD,7
goto   salir_fecha_hora
```

```
    btfsc PORTD,6
    goto  ff7
```

ff8

```
    call  cambio_valor
    movlw 0x0B
    movwf dir_LCD
    movff temp_0,temp_8
    movff temp_0,dato_LCD
    call  Escribir_LDC
    btfsc PORTD,7
    goto  salir_fecha_hora
    btfsc PORTD,6
    goto  ff8
```

ff9

```
    call  cambio_valor
    movlw 0x0D
    movwf dir_LCD
    movff temp_0,temp_9
    movff temp_0,dato_LCD
    call  Escribir_LDC
    btfsc PORTD,7
    goto  salir_fecha_hora
    btfsc PORTD,6
    goto  ff9
```

ff10

```
    call  cambio_valor
    movlw 0x0E
    movwf dir_LCD
    movff temp_0,temp_10
    movff temp_0,dato_LCD
    call  Escribir_LDC
```

```
btfscl PORTD,7
goto salir_fecha_hora
btfscl PORTD,6
goto ff10
```

;Actualización de datos modificados.

```
movff temp_1,dia_2
movff temp_2,dia_1
movff temp_3,mes_2
movff temp_4,mes_1
movff temp_5,anno_2
movff temp_6,anno_1
movff temp_7,hora_2
movff temp_8,hora_1
movff temp_9,min_2
movff temp_10,min_1
```

```
movlw .10
mulwf temp_1
movf PRODL,0
addwf temp_2,0
movwf dia
movlw .10
mulwf temp_3
movf PRODL,0
addwf temp_4,0
movwf mes
movlw .10
mulwf temp_5
movf PRODL,0
addwf temp_6,0
movwf anno
movlw .10
```

```

mulwf temp_7
movf  PRODL,0
addwf temp_8,0
movwf hora
movlw .10
mulwf temp_9
movf  PRODL,0
addwf temp_10,0
movwf minuto

```

salir\_fecha\_hora

```

return

```

```

;-----
;-----
;Rutina de cambio de constantes de proporcionalidad.

```

cambio\_constante

```

call  Constante

```

cc1

```

call  cambio_valor
movlw .8
movwf dir_LCD
movff temp_0,temp_1
movff temp_0,dato_LCD
call  Escribir_LDC
btsc  PORTD,7
goto  salir_constante
btsc  PORTD,6
goto  cc1

```

cc2

```

call  cambio_valor
movlw .9
movwf dir_LCD
movff temp_0,temp_2

```

```

movff temp_0,dato_LCD
call Escribir_LDC
btfsc PORTD,7
goto salir_constante
btfsc PORTD,6
goto cc2
cc3
call cambio_valor
movlw 0x0B
movwf dir_LCD
movff temp_0,temp_3
movff temp_0,dato_LCD
call Escribir_LDC
btfsc PORTD,7
goto salir_constante
btfsc PORTD,6
goto cc3
cc4
call cambio_valor
movlw .48
movwf dir_LCD
movff temp_0,temp_4
movff temp_0,dato_LCD
call Escribir_LDC
btfsc PORTD,7
goto salir_constante
btfsc PORTD,6
goto cc4
cc5
call cambio_valor
movlw .49
movwf dir_LCD
movff temp_0,temp_5
movff temp_0,dato_LCD

```

```

    call    Escribir_LDC
    btfsc  PORTD,7
    goto   salir_constante
    btfsc  PORTD,6
    goto   cc5
cc6
    call    cambio_valor
    movlw  0x4B
    movwf  dir_LCD
    movff  temp_0,temp_6
    movff  temp_0,dato_LCD
    call    Escribir_LDC
    btfsc  PORTD,7
    goto   salir_constante
    btfsc  PORTD,6
    goto   cc6
;Establecemos los valores de la constante kt y ki
    movff  temp_1,ki_2
    movff  temp_2,ki_1
    movff  temp_3,ki_0
    movff  temp_4,kt_2
    movff  temp_5,kt_1
    movff  temp_6,kt_0

    movff  temp_3,ki_dec
    movff  temp_6,kt_dec

    movlw  .10
    mulwf  temp_1
    movf   PRODL,0
    addwf  temp_2,0
    movwf  ki
    movlw  .10
    mulwf  temp_4

```

```

movf  PRODL,0
addwf temp_5,0
movwf kt

```

;Almacenamos el valor de las constantes kt & ki en la EEPROM del PIC, con el  
;propósito en caso de reinicio del dispositivo o falla en la alimentación,  
;se mantendrán sus últimos parámetros de calibración en la EEPROM del PIC.

```

movlw 0x45
movwf Dir_EEPROM
movff ki,Dato_EEPROM
call  Escribir_EEPROM_PIC
movlw 0x46
movwf Dir_EEPROM
movff ki_dec,Dato_EEPROM
call  Escribir_EEPROM_PIC
movlw 0x47
movwf Dir_EEPROM
movff kt,Dato_EEPROM
call  Escribir_EEPROM_PIC
movlw 0x48
movwf Dir_EEPROM
movff kt_dec,Dato_EEPROM
call  Escribir_EEPROM_PIC

```

```

salir_constante
return

```

```

;-----
;-----

```

```

;Rutina de cambio de período de muestreo
cambio_período
call  Periodo

```

pp1

```
call    cambio_valor
movlw  .4
movwf  dir_LCD
movff  temp_0,temp_1
movff  temp_0,dato_LCD
call   Escribir_LDC
btfsc  PORTD,7
goto   salir_período
btfsc  PORTD,6
goto   pp1
```

pp2

```
call    cambio_valor
movlw  .5
movwf  dir_LCD
movff  temp_0,temp_2
movff  temp_0,dato_LCD
call   Escribir_LDC
btfsc  PORTD,7
goto   salir_período
btfsc  PORTD,6
goto   pp2
```

pp3

```
call    cambio_valor
movlw  .6
movwf  dir_LCD
movff  temp_0,temp_3
movff  temp_0,dato_LCD
call   Escribir_LDC
btfsc  PORTD,7
goto   salir_período
btfsc  PORTD,6
goto   pp3
```

;establecemos el valor del período en el programa.

```
movff temp_1,T_3
movff temp_2,T_2
movff temp_3,T_1
```

```
movlw .100
mulwf temp_1
movf PRODL,0
addwf temp_3,1
movlw .10
mulwf temp_2
movf PRODL,0
addwf temp_3,0
movwf T
```

;Almacenamos el período de muestreo en la memoria EEPROM del PIC

```
movlw 0x49
movwf Dir_EEPROM
movff T,Dato_EEPROM
call Escribir_EEPROM_PIC
```

```
salir_periodo
return
```

```
;-----
;-----
```

;Escritura y lectura en memoria EEPROM serial

configuracion\_serial

;inicialización de variables

```
movlw 0x28
movwf SSPCON1
clrf SSPCON2
clrf SSPSTAT
movlw 0x18
```

```

movwf SSPADD
bcf    PIR1,SSPIF
bcf    SSPCON1,SSPOV
bcf    SSPCON2,ACKEN

bcf    SSPCON2,6
bsf    SSPCON2,0    ;señal de inicio

ss1
btfss  PIR1,3
goto   ss1
bcf    PIR1,3
movff  mserial,temp_0
bcf    STATUS,C
rlncf  temp_0,1
movf   temp_0,0
addlw  b'10100000' ;byte de control
movwf  SSPBUF

ss2
btfss  SSPCON2,6    ;reconocimiento
goto   ss2
bcf    SSPCON2,6

btfss  PIR1,3
goto   ss3
bcf    PIR1,3
movff  DHEEPROM,SSPBUF ;Hbyte de dirección

ss3
btfss  SSPCON2,6
goto   ss3
bcf    SSPCON2,6

goto   ss3
bcf    PIR1,3
movff  DLEEPROM,SSPBUF ;Lbyte de dirección

```

ss4

```
btfss SSPCON2,6
goto ss4
bcf SSPCON2,6
```

return

;------

escritura\_serial

;Rutina para la escritura en la memoria EEPROM serial

```
call configuración_serial
goto ss3
bcf PIR1,3
movff D_SEEPROM,SSPBUF;Dato
```

ss5

```
btfss SSPCON2,6
goto ss5
bcf SSPCON2,6
```

```
bsf SSPCON2,4 ;condición de parada
bsf SSPCON2,2
```

return

;------

;Rutina para lectura de EEPROM

lectura\_serial

```
call configuracion_serial
bsf SSPCON2,1
```

ss6

```
btfss PIR1,3
goto ss6
```

```

    bcf    PIR1,3
    movff  mserial,temp_0
    bcf    STATUS,C
    rlnof  temp_0,1
    movf   temp_0,0
    addlw  b'10100001'    ;byte de control
ss7
    btfss  SSPCON2,6    ;reconocimiento
    goto   ss7
    bcf    SSPCON2,6

    bsf    SSPCON2,3    ;lectura de memoria
ss8
    btfss  SSPCON2,6
    goto   ss8
    bcf    SSPCON2,6

    movff  SSPBUF,D_SEEPROM;dato recuperado

    bsf    SSPCON2,4    ;condición de parada
    bsf    SSPCON2,2

    return
;-----
localidad    ;aumenta la localidad de la memoria EEPROM serial
    incf   DLEEPROM,1
    btfsc  STATUS,C
    incf   DHEEPROM
    movlw  0xFA
    cpfslt DHEEPROM
    clrf   DHEEPROM
    movf   DHEEPROM,1
    btfsc  STATUS,Z
    incf   mserial,1

```

return

;-  
;-

;Se inician las rutinas para el manejo de la interrupciones por muestreo  
;y por transferencia de datos

interrupciones

```
    btfsc  INTCON,TMR0IF
    call   control_tiempo_muestreo
    btfsc  PIR1,6
    call   transmision_PC
    retfie
```

;-

;Esta rutina se encarga del control del tiempo

control\_tiempo\_muestreo

```
    bcf    INTCON,TMR0IF
    movff  minuto,temp_10
    call   fecha_hora
    movf   temp_10,0
    cpfseq minuto
    call   muestreo
    return
```

;-

;Muestreo de datos

muestreo

```
    incf   n,1
    bsf    ADCON0,2
```

conversion

```
    btfss  PIR1,6
    goto   conversion
```

```

    bcf    PIR1,6
    movff  ADRESH,muestra
    movf   k,0
    mulwf  muestra,0
    movff  PRODH,temp_0
    movff  PRODL,temp_1
    clrf   temp_2
    movff  temp_1,temp_3
;Cálculo para el despliegue en pantalla
dd0
    movlw  .100
    subwfb temp_1,1
    btfs   STATUS,C
    goto   dd1
    incf   temp_2,1
    movff  temp_1,temp_3
    goto   dd0
dd1
    decf   temp_0,1
    btfs   STATUS,C
    goto   dd0

    movff  temp_2,dato_2
    clrf   temp_2
    movff  temp_3,temp_1
dd2
    movlw  .10
    subwfb temp_1,1
    btfs   STATUS,C
    goto   dd3
    incf   temp_2,1
    movff  temp_1,temp_3
    goto   dd2
dd3

```

```

movff temp_2,dato_1
movff temp_3,dato_0
call Desplegar_fecha_hora
call Desplegar_datos

```

;Rutina de almacenamiento

```

movf muestra,0
addwf m_1,1
btfsc STATUS,C
incf m_2,1
movf T,0
cpfseq n
return

```

;Almacenar en EEPROM serial

;obtener dato promedio

```

clrf temp_0
clrf temp_1
movff m_2,temp_2
movff m_1,temp_3
movf T,0
movff temp_3,temp_1

```

mm1

```

subwfb temp_3
btfsc STATUS,C
goto mm3

```

mm2

```

incf temp_0,1
movff temp_3,temp_1
goto mm1

```

mm3

```

decf temp_2,1
btfss STATUS,C
goto mm2

```

```

        movff  temp_0,muestra_m
;Inicia escritura hacia memoria serial
        call   localidad
        movlw  0x0AA
        movwf  D_SEEPROM
        call   escritura_serial

        call   localidad
        movff  anno,D_SEEPROM           ;almacenando año
        call   escritura_serial

        call   localidad
        movff  mes,D_SEEPROM           ;almacenando mes
        call   escritura_serial

        call   localidad
        movff  dia,D_SEEPROM           ;almacenando dia
        call   escritura_serial

        call   localidad
        movff  hora,D_SEEPROM         ;almacenando hora
        call   escritura_serial

        call   localidad
        movff  minuto,D_SEEPROM       ;almacenando minuto
        call   escritura_serial

        call   localidad
        movff  muestra_m,D_SEEPROM     ;almacenando dato
        call   escritura_serial

        call   localidad
        movlw  0x55

```

```
movwf D_SEEPROM
call escritura_serial
return
```

```
;-----
;Rutina para transmisión de datos seriales hacia la PC
```

```
transmitir
```

```
bcf PIR1,5
movff datum,TXREG
```

```
transmitiendo
```

```
btfss TXSTA,1
goto transmitiendo
bcf TXSTA,1
return
```

```
;-----
;Rutina para transmisión de datos seriales hacia la PC
```

```
transmision_PC
```

```
call transferencia
bcf PIR1,6
movff DHEEPROM,temp_0
movff DLEEPROM,temp_1
movff mserial,temp_2
clrf DHEEPROM
clrf DLEEPROM
```

```
continúa
```

```
movf temp_1,0
cpfsft DLEEPROM
goto proceso
movf temp_0,0
cpfsft DHEEPROM
goto proceso
```

```

        movf    temp_2,0
        cpfseq  mserial
        goto   proceso
        goto   finalizar_transmision
proceso                                     ;transferencia de datos
de PIC a PC
        call   lectura_serial
        movff  D_SEEPROM,datum
        call   transmitir
        incf   DLEEPROM,1
        btfsc STATUS,C
        incf   DHEEPROM
        movlw  0x0FA
        cpfseq DHEEPROM
        goto   continúa
        incf   mserial,1
        goto   continúa
finalizar_transmisión
        movlw  0x5A                                     ;envío de dato de parada
        movwf  datum
confirmación                               ;lectura de dato de confirmación
        btfs  PIR1,6
        goto  confirmación
        bcf   PIR1,6
        movlw 0xF0
        cpfseq RCREG
        return
;si condición es valida reiniciamos el almacenamiento en la memoria
;EEPROM serial
        clrf  DHEEPROM
        clrf  DLEEPROM
        clrf  mserial
        return

```

```
;-----  
;  
;-----  
;  
;-----
```

;RUTINA PRINCIPAL!!!

;Configuración de los puertos y el convertidor analógico digital

Rutina\_principal

```
    movlw 0x01  
    movwf TRISA  
    movlw 0x80  
    movwf ADCON0  
    movlw 0x40  
    movwf ADCON1  
    clrf  TRISB  
    movlw 0x92  
    movwf TRISC  
    movlw 0x0F0  
    movwf TRISD
```

;Configuración del USART para recibir continuamente

```
    movlw 0xE0  
    movwf INTCON
```

;Se habilitan las interrupciones globales y periféricas

;Habilitamos la interrupción del TMR0

```
    movlw 0x20  
    movwf PIE1
```

;Habilitación de la interrupción generada cada vez que se reciba un dato

```
    movlw 0x20  
    movwf IPR1
```

;Asignamos como prioridad alta a la interrupción por dato recibido

```
    movlw 0x04
```

```

        movwf TXSTA
;Configuramos la transmisión del USART
        movlw 0x90
        movwf RCSTA
;Se habilita el módulo de recepción continua USART
        movlw 0x19
        movwf SPBRG
;Velocidad de transmisión USART 96 kbps
;Configuración de temporizador TMR0
        movlw 0xF6
        movwf T0CON
;A partir de este momento el temporizador inicia su conteo

;Configuración del LCD
;Encender la pantalla LCD
        movlw 0x0E
        movwf comando
        call comando_LCD

;Borrar pantalla
        movlw 0x01
        movwf comando
        call comando_LCD

;Modo de operación de 8 bits
        movlw 0x38
        movwf comando
        call comando_LCD

;Cursor en movimiento
        movlw 0x07
        movwf comando
        call comando_LCD

```

;Inicialización de variables

```
clrf    segundo
clrf    minuto
clrf    hora
clrf    anno
clrf    min_1
clrf    min_2
clrf    hora_1
clrf    hora_2
clrf    día_2
clrf    mes_2
clrf    anno_1
clrf    anno_2
clrf    dato_2
clrf    dato_1
clrf    dato_0
movlw  0x01
movf   día
movf   día_1
movf   mes
movf   mes_1
```

;Configuración inicial de valores a desplegar

```
call   cambio_fecha_hora    ;configurar fecha y hora
```

;Verificamos la existencia de valores previos de parámetros de calibración  
;en la memoria EEPROM, estos valores pueden ser valores de programación o  
;valores almacenados de una nueva calibración del instrumento.

```
movlw  0x40
movwf  Dir_EEPROM
call   Leer_EEPROM_PIC
```

```

    movff Dato_EEPROM,temp_0
    movf  temp_0,0
    btfss STATUS,Z
    call  cambio_constante      ;configurar constantes de proporcionalidad
    movff kt,temp_0
    movff ki,temp_1
    clrf  temp_2
dividir
    movf  temp_0,0
    subwfb temp_1,1
    btfss STATUS,C
    goto  seguir
    incf  temp_2,1
    goto  dividir
seguir
    movff temp_2,k

```

;Verificamos la existencia de un período de muestreo almacenado en la  
;memoria EEPROM, de acuerdo a cambio efectuados por el usuario

```

    movlw 0x49
    movwf Dir_EEPROM
    call  Leer_EEPROM_PIC
    movff Dato_EEPROM,temp_0
    movf  temp_0,0
    btfss STATUS,Z
    call  cambio_periodo        ;configurar período de muestreo
    call  Desplegar_fecha_hora
    call  Desplegar_datos

;ciclo principal
ciclo_principal
    call  Desplegar_fecha_hora
    btfss PORTD,4      ;Tecla +

```

```
call cambio_fecha_hora
btfss PORTD,5 ;Tecla -
call cambio_constante
btfss PORTD,6 ;Tecla Aceptar
call cambio_período
goto ciclo_principal
end
```

;------

;------

;Fin de programa de control del anemógrafo digital