

**UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE TECNOLOGIA
CARRERA ELECTRONICA Y TELECOMUNICACIONES**



**NIVEL LICENCIATURA
EXAMEN DE GRADO
PROYECTO DE APLICACIÓN**

**“ANEMOMETRO POR EL PUERTO USB - CLASE CDC
EN BASE AL PIC18F4550”**

POSTULANTE: OSCAR CHAMBI CHAMBI

DICIEMBRE 2012

LA PAZ – BOLIVIA

DEDICATORIA

A mis padres, por su cariño y apoyo incondicional.

AGRADECIMIENTOS

A la Carrera de Electrónica y Telecomunicaciones, por haberme permitido hallar conocimiento y amistad en sus aulas.

ÍNDICE

	Pág.
1. RESUMEN DEL TRABAJO	1
2. PLANTEAMIENTO DEL PROBLEMA.....	2
3. JUSTIFICACIÓN DEL TRABAJO.....	2
4. OBJETIVOS.....	3
4.1 Objetivo general.....	3
4.2 Objetivos específicos.....	3
5. FUNDAMENTACIÓN TEÓRICA.....	4
5.1 Viento.....	4
5.1.1 Velocidad y dirección del viento.....	4
5.2 Pic 18F4550.....	5
5.2.1 Organización de la memoria.....	7
5.2.2 Sistema de interrupciones.....	8
5.2.3 Timer0.....	9
5.2.4.1 Modo contador.....	10
5.2.4 Timer1.....	11
5.2.5.1 Interrupción por desbordamiento.....	12
5.2.5 Conversor ADC.....	13
5.2.5.1 Selección del canal de conversión.....	15
5.2.5.2 Señal de reloj de conversión.....	16
5.2.5.3 Tiempo de adquisición.....	16
5.2.6 Módulo USB.....	17
5.2.6.1 USB Clase CDC.....	18
5.3 Optoacoplador	19

5.4 Potenciómetro 360°.....	21
6. DESARROLLO DEL TRABAJO.....	22
6.1 Algoritmo del programa.....	22
6.2 Programación del Pic.....	23
6.2.1 Programación del Timer1.....	23
6.2.2 Programación del Timer0.....	24
6.2.3 Programación del ADC.....	25
6.2.4 Programación del periférico USB.....	26
6.2.5 Programa completo en CCS.....	27
6.3 Programa en la PC con Visual Basic 6.0.....	29
6.4 Circuito del anemómetro.....	30
6.5 Aspecto físico.....	31
6.6 Funcionamiento del anemómetro.....	32
6.6 Costos del proyecto.....	34
7. CONCLUSIONES.....	35
8. BIBLIOGRAFÍA.....	36
ANEXOS.....	37

ÍNDICE DE FIGURAS

Figura	Pág.
Figura 1. Diagrama en bloques del PIC 18f4550.....	6
Figura 2. Diagrama de pines PIC 18F4550.....	7
Figura 3. Organización de la memoria del PIC 18F4550.....	8
Figura 4. Diagrama en bloques del Timer0 en modo 8 bits.....	10
Figura 5. Diagrama en bloques del Timer1.....	12
Figura 6. Esquema eléctrico de optoacoplador en U.....	20
Figura 7. Forma externa de optoacoplador en U.....	21
Figura 8. Aspecto externo y datos del potenciómetro de 360°.....	21
Figura 9. Programa diseñado en Visual Basic 6.0.....	30
Figura 10. Circuito elaborado en Proteus.....	30
Figura 11. Piñón usado para el anemómetro.....	31
Figura 12. Piñón dentro el anemómetro.....	31
Figura 13. Potenciómetro encargado de medir el ángulo.....	32
Figura 14. Vista exterior del anemómetro.....	32
Figura 15. Reconocimiento del Pic en la PC.....	33
Figura 16. Puerto Com7 en Visual Basic.....	33
Figura 17. Visual Basic recibiendo datos del anemómetro.....	34

1. RESUMEN DEL TRABAJO

El presente trabajo de aplicación es el diseño y la implementación de un anemómetro cuya función principal es la medición del viento tanto en velocidad como en dirección. Para este fin se utilizó el pic18f4550, ya que este microcontrolador posee el módulo USB.

La medición de la velocidad del viento se la realizó utilizando el optoacoplador con un ciclo de trabajo de 50% el cual nos brindará información sobre el número de vueltas en un determinado intervalo de tiempo. Se utilizó el Timer1 del pic18f4550 para generar tiempos de desbordamiento de 1 segundo y el Timer0 como contador de flancos de bajada que genera el optoacoplador, el contador cuenta las vueltas y se pone en cero cuando existe un desbordamiento en el Timer1. Para que los datos sean lo más exacto posibles se recurrió a la fórmula de la medición del viento que involucra el número de vueltas, el tiempo y el radio del las aletas del anemómetro.

La medición de la dirección del viento se la realizó mediante un potenciómetro sin fin el cual nos da información de los grados de dirección en comparación a la resistencia que tenga, el potenciómetro sube hasta su máximo valor para continuar con su mínimo valor inmediatamente, gracias al módulo de conversión ADC del pic18f4550 se utilizaron 10 bits de conversión para tomar datos más precisos.

Para el envío de la información a la PC se usó el módulo UBS del pic - clase CDC que implementa un puerto virtual COMM, la recepción de los datos en la PC fue implementado en Visual Basic 6.0 utilizando el puerto COMM y por supuesto pueden ser visualizados en un textbox. La configuración del puerto comm se lo realizó en el MScomm1 parámetros como la cantidad de baudios, bits de paridad, etc.

2. PLANTEAMIENTO DEL PROBLEMA

El viento es un fenómeno que puede presentarse en cualquier dirección y además con cualquier magnitud, la necesidad de medir este fenómeno es importante, no sólo por tema de seguridad sino también de aprovechamiento de la energía.

Entonces se necesita poder tomar datos en los 360°, a esto debemos aumentar el hecho de que existen pocos dispositivos para este fin en el mercado. La velocidad del viento en un anemómetro viene determinado por el número de vueltas que este instrumento gire en un determinado tiempo, entonces el uso de un dispositivo que pueda sensor esto también es evidente.

A esto se suma que en las PC's los puertos paralelo y serie ya no vienen incorporados y se hace necesario utilizar el puerto USB que es el que ahora tiene más demanda.

¿Cómo implementar un anemómetro por el puerto USB en base al microcontrolador 18F4550?.

3. JUSTIFICACION DEL TRABAJO

En la electrónica digital siempre ha sido necesario el censar fenómenos físicos ya sean estos de temperatura, presión, flujo, intensidad de corriente, etc., no puede ser menos importante la medición del viento, su uso siempre ha sido importante para el hombre; la necesidad de saber en qué dirección y con qué velocidad se dirige es importante en aeropuertos, en zonas geográficamente amenazadas por huracanes, etc.

Otra de las aplicaciones de un anemómetro puede ser el hecho de saber el promedio de velocidad del viento en un región determinada, quizá para su

aprovechamiento en la conversión de energía eólica en energía eléctrica por ejemplo.

Contar con estos datos de velocidad y dirección pueden ser muy útiles si son datos digitales, teniendo en cuenta que las computadoras actuales no vienen con un puerto paralelo incorporado y muy pocas con puerto serial, el uso del puerto USB para este fin es una buena alternativa.

4. OBJETIVOS

4.1 OBJETIVO GENERAL

Medir la velocidad y dirección del viento en base a un microcontrolador, y visualizar estos datos en la PC.

4.2 OBJETIVOS ESPECIFICOS

- Diseñar e implementar un anemómetro el cual se enviará datos por el puerto USB - clase cdc.
- Utilizar los recursos adecuados del pic18f4550 para la elaboración del proyecto.
- Emplear sensores ópticos y potenciómetros sin fin para determinar la velocidad y dirección del viento.
- Elaborar un programa en Visual Basic que pueda recepcionar los datos enviados por el microcontrolador.

5. FUNDAMENTACIÓN TEÓRICA

5.1 VIENTO

Es un fenómeno físico que tiene su procedencia en la incidencia del sol sobre la tierra, a la vez esta calienta al aire a su alrededor, así los vientos son producidos por la diferencia en temperatura del aire.

El viento es el flujo de gases a gran escala. En la Tierra, el viento es el movimiento en masa del aire en la atmósfera.¹

5.1.1 VELOCIDAD Y DIRECCION DEL VIENTO

La velocidad es la fuerza con que el viento se traslada, mientras que la dirección es la ruta o la procedencia y el destino al cual se dirige el viento. Un instrumento antiguo en la medición de la dirección del viento es la veleta.

La medición de la velocidad y dirección del viento se efectúa con instrumentos registradores llamados anemómetros, que disponen de dos sensores: uno para medir la velocidad y otro para medir la dirección del viento.²

La intensidad del viento se ordena según su rapidez utilizando la escala de Beaufort. Esta escala se divide en varios tramos según sus efectos y/o daños causados, desde el aire en calma hasta los huracanes de categoría 5 y los tornados.³

La velocidad del viento puede ser determinada por un anemómetro determinando el número de vueltas en un determinado tiempo.

¹ Enciclopedia Wikipedia. Viento

² Enciclopedia Wikipedia. Viento

³ Enciclopedia Wikipedia. Viento

$$Vv = 2\pi rfs \quad (1)$$

Donde r es la radio del anemómetro y f_s es la frecuencia de la señal, cabe señalar que la frecuencia de la señal está determinada por:

$$f_s = \frac{n}{t} \quad (2)$$

Siendo n el número de vueltas y t el tiempo en el que se toma la muestra. Sustituyendo la ecuación (2) en la (1) tenemos la velocidad del viento en relación al número de vueltas y el tiempo:⁴

$$Vv = 2\pi r \frac{n}{t} \quad (3)$$

5.2 PIC18F4550

Este microcontrolador es de gama alta, construido en arquitectura Harvard con 40 pines. Sus características principales son las siguientes:

- Frecuencia de operación hasta 48 MHz.
- Memoria de programa 32768 Bytes, RAM de 2048 bytes y EEPROM de 256 bytes.
- 20 interrupciones, 35 líneas de E/S, 4 temporizadores.
- Módulos de comparación y captura.
- Canales de comunicación serie MSSP y EUSART.
- Canal USB
- Puerto paralelo de comunicación de datos SSP.
- 13 canales de conversión A/D con 10 bits.
- 2 comparadores analógicos.

⁴ Ríos Piña, María del Rosario. Anemómetro. p. 8

El diagrama en bloques es el siguiente:

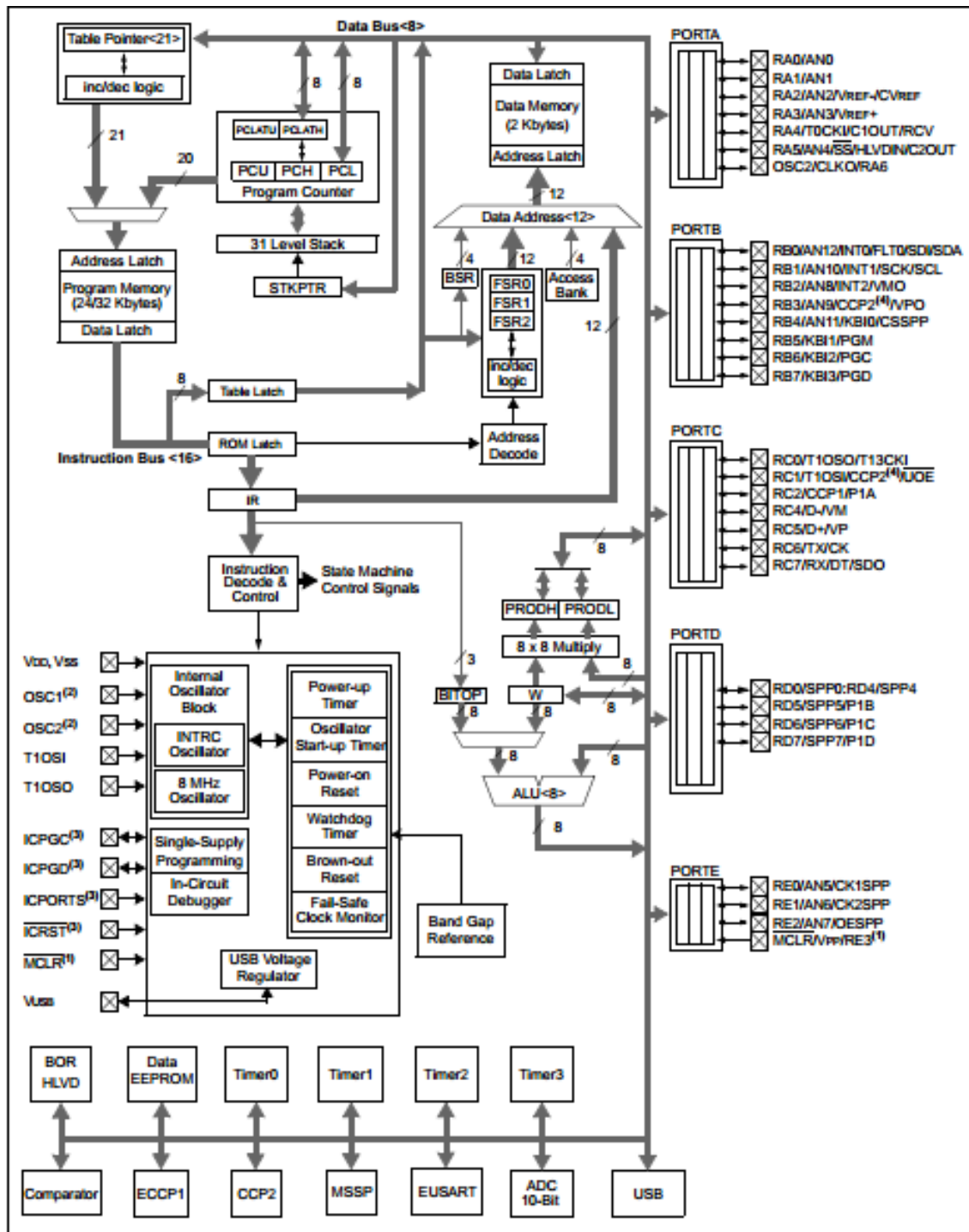


Fig. 1 Diagrama en bloques del PIC 18f4550.

La configuración de sus pines:

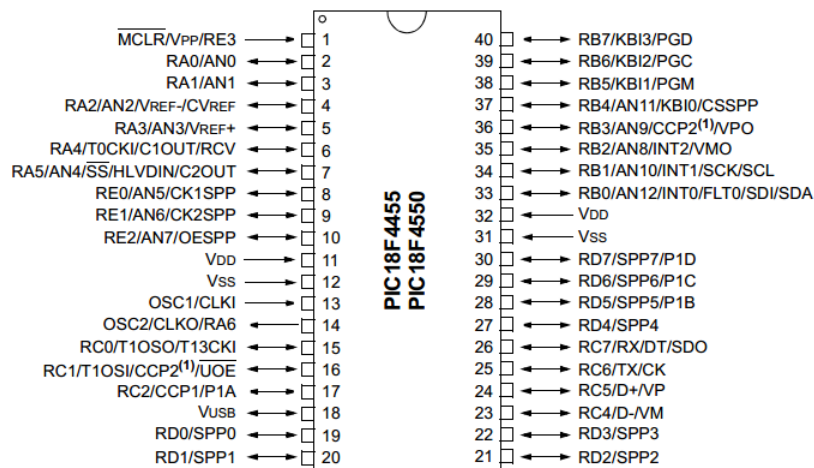


Fig.2 Diagrama de pines PIC 18F4550.

5.2.1 ORGANIZACIÓN DE LA MEMORIA

- El uC PIC18F4550 dispone una memoria de programa de 32.768 bytes de memoria de programa (0000H-7FFFH). Las instrucciones ocupan 2 bytes (excepto CALL, MOVFF, GOTO y LSFR que ocupan 4). Por lo tanto la memoria de programa pueden almacenar hasta 16.384 instrucciones.
- La operación de lectura en posición de memoria por encima de 7FFFH da '0' como resultado (equivalente a la instrucción NOP).
- Direcciones especiales de la memoria de programa
 - Vectorización del Resetes 0000H
 - Vectorización de las interrupciones de alta prioridades la 0008H.
 - Vectorización de las interrupciones de baja prioridades la 0018H.⁵

⁵ Universidad Politécnica de Valencia. Microcontrolador PIC18F4550. p. 11

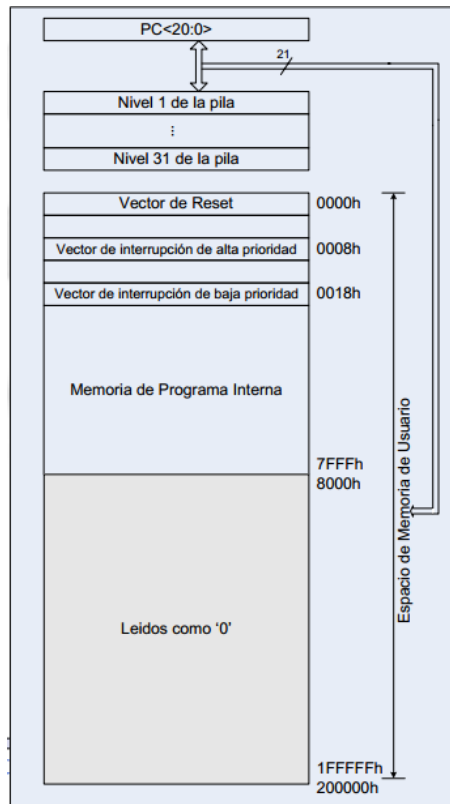


Fig. 3 Organización de la memoria del PIC 18F4550.

5.2.2 SISTEMA DE INTERRUPTIONES

Las interrupciones en el PIC 18F4550 son variadas y con configuraciones similares:

- Se dispone de dos niveles de prioridad:
 - Nivel alto vectorizado en la dirección 0008H
 - Nivel bajo, vectorizado en la dirección 0018H
- Todas las interrupciones pueden ser programadas con cualquiera de las dos prioridades, salvo la interrupción externa 0 (que siempre tiene alta prioridad).
- Todas las interrupciones disponen de 3 bits de configuración (excepto la interrupción externa 0 que tiene dos):

- Bit de habilitación de interrupción: permite habilitar a nivel individual la interrupción.
- Flag de interrupción: se pone a '1' cuando se produce la condición de interrupción independientemente de si la interrupción está habilitada o no. Este flag debe ponerse '0' por software cuando se procesa la interrupción.
- Bit de prioridad de interrupción: establece si la interrupción es de alta o de baja prioridad (este bit no está disponible para la interrupción externa 0).⁶

5.2.3 TIMER0

El Timer0 es un contador/temporizador de 8/16 bits de registro, es un contador cuando la cuenta es externa y lo hace por RA4, si la cuenta es interna es un temporizador. Cuenta con un Pre escalador de 8 bits e interrupción por desbordamiento. Para su configuración se utiliza el registro T0CON.



- TMR0ON: Bit de puesta en marcha del Temporizador 0
- T08BIT: Bit de configuración del modo 8-bit/16-bit:
 - T08BIT='0': Modo 16-bit
 - T08BIT='1': Modo 8-bit
- T0CS: Bit de configuración del modo contador/temporizador
 - T0CS='0': Modo temporizador (TMR0 se incrementa en cada ciclo de instrucción FOSC/4)
 - T0CS='1': Modo contador (TMR0 se incrementa en cada transición de la línea RA4/T0CKI)

⁶ Universidad Politécnica de Valencia. Microcontrolador PIC18F4550. p. 44

- T0SE: Bit de selección de flanco en modo contador
 - T0SE='0': TMR0 se incrementa en los flancos de subida de la línea RA4/T0CKI
 - T0SE='1': TMR0 se incrementa en los flancos de bajada de la línea RA4/T0CKI
- PSA: Bit de activación del pre-escalador
 - PSA='0': Pre-escalador activado
 - PSA='1': Pre-escalador no activado
- T0PS2..T0PS0: Bits de selección del pre-escalador del Temporizador 0:⁷

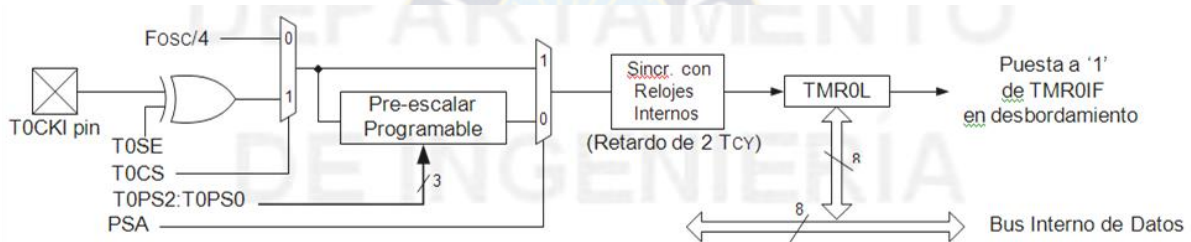


Fig.4 Diagrama en bloques del Timer0 en modo 8 bits

5.2.3.1 MODO CONTADOR

- T08BIT (T0CON)='1'/T0CS (T0CON)='1'
- Dos opciones:
 - Sin pre-escalador PSA (T0CON)='0': el registro TMR0L se incrementa con cada flanco de subida (T0SE='0') o de bajada (T0SE='1') de la línea T0CKI (RA4).
 - Con pre-escalador PSA (T0CON)='1': el contador de pre-escalador se incrementa con cada flanco de subida (T0SE='0') o de bajada (T0SE='1') de la línea T0CKI (RA4). Cuando el pre-escalador alcanza el valor programado en los bits PS2..PS0 (T0CON) se produce un pulso que incrementa el registro TMR0L.

⁷ Universidad Politécnica de Valencia. Microcontrolador PIC18F4550. p. 84

- Cuando se produce desbordamiento de TMR0L se pone a '1' el flag TMR0IF.⁸

5.2.4 TIMER1

El Timer1 es también un contador/temporizador con las siguientes características:

- Configurable como temporizador/contador de 16 bits
- Dispone de un oscilador propio que puede funcionar como:
 - Señal de reloj del temporizador 1
 - Señal de reloj del uC en modos de bajo consumo
- Pre-escalar de 3 bits programable
- Interrupción por desbordamiento



- RD16: Bit selección de modo de lectura/escritura de TMR1
 - RD16='0': Lectura/escritura deTMR1 en dos operaciones independientes de 8 bit.
 - RD16='1': Lectura/escritura deTMR1 en modo de 16-bit.
- T1RUN: Bit de monitorización de la fuente del reloj principal del sistema (solo lectura):
 - T1RUN='0': Reloj principal no derivado del oscilador del Temporizador 1.
 - T1RUN='1': Reloj principal derivado del oscilador del Temporizador 1.
- T1CKPS1..T1CKPS0: Bits de selección del pre-escalar del Temporizador 1.
- T1OSCEN: Bit de habilitación del oscilador del Temporizador 1.
 - T1OSCEN='0': Oscilador del Temporizador 1 desactivado
 - T1OSCEN='1': Oscilador del Temporizador 1 activado

⁸ Universidad Politécnica de Valencia. Microcontrolador PIC18F4550. p. 86

- T1SYNC: Bit de sincronización de la señal de reloj (solo aplica cuando TMR1CS='1').
 - T1SYNC='0': Se sincroniza la entrada de reloj externa RC0/T1OSO/T13CKI
 - T1SYNC='1': No se sincroniza la entrada de reloj externa RC0/T1OSO/T13CKI
- TMR1CS: Bits de selección de la fuente de incremento del Temporizador 1.
- TMR1CS='0': Pre-escalar del Temp. 1 se incrementa con cada ciclo de instrucción $F_{osc}/4$.
- TMR1CS='1': Pre-escalar del Temp. 1 se incrementa con cada flanco de subida de la línea RC0/T1OSO/T13CKI.
- TMR1ON: Bit de puesta en marcha del Temporizador 1.⁹

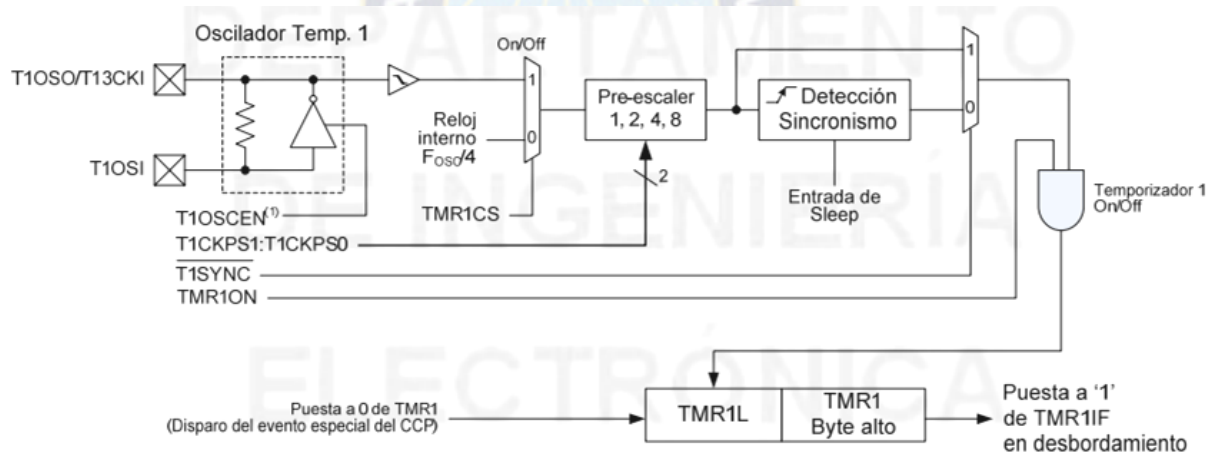


Fig. 5 Diagrama en bloques del Timer1.

5.2.4.1 INTERRUPCION POR DESBORDAMIENTO

- Flag de interrupción: bit TMR1IF (PIR1)
- Bit de habilitación: bit TMR1IE (PIE1)
- Prioridad: bit TMR1IP (IPR1): '0' -> pr. baja / '1' -> pr. alta

⁹ Universidad Politécnica de Valencia. Microcontrolador PIC18F4550. p. 94

Si se produce el desbordamiento del Temporizador 1 se pone a '1' el flag TMR1IF. Si el bit de habilitación TMR1IE está a '1' y las interrupciones de periféricos están habilitadas a nivel global se genera una interrupción y el uC pasa a ejecutar el código situado a partir de la posición 0008H o 0018H (según el nivel de prioridad establecido).¹⁰

El tiempo de desbordamiento del Timer1 se calcula mediante la siguiente ecuación:¹¹

$$T = T_{CM} \times Prescaler \times (65536 - CargaTimer1) \quad (4)$$

Donde el ciclo máquina puede obtenerse de la siguiente forma:

$$T_{CM} = \frac{4}{F_{OSC}} \quad (5)$$

Reemplazando la ecuación (5) en la (4), tenemos:

$$T = \frac{4}{F_{OSC}} \times Prescaler \times (65536 - CargaTimer1) \quad (6)$$

5.2.6 CONVERTOR ADC

Los microcontroladores PIC tienen un módulo de conversión analógico digital que usan un sistema de muestreo y retención. Además este módulo emplea las aproximaciones sucesivas para su cometido. Sus características principales son las siguientes:

- 10 bits de resolución
- 13 canales multiplexados
- Señal de reloj de conversión configurable

¹⁰ Universidad Politécnica de Valencia. Microcontrolador PIC18F4550. p. 100

¹¹ García Breijo, Eduardo. Compilador CCS y simulador Proteus para microcontroladores PIC. p. 101

- Tiempo de adquisición programable (0 a 20TAD)
- Posibilidad de establecer el rango de tensiones de conversión mediante tensiones de referencia externas. ¹²

Los registros involucrados en la conversión analógico digital son ADCON0, ADCON1, ADCON2.

REGISTRO ADCON0

	-0	-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
ADCON0	-	-	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON

- CHS3..CHS0: Bits selección del canal de conversión A/D (13 canales)
- GO/DONE: Bit de inicio y de monitorización del estado de la conversión A/D:
 - GO/DONE='0': Proceso de conversión parado
 - GO/DONE='1': Proceso de conversión en marcha
- ADON: Bit de habilitación del convertidor A/D
 - ADON='0': Convertidor A/D desactivado
 - ADON='1': Convertidor A/D activado

REGISTRO ADCON1

	-0	-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
ADCON1	-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0

- VCFG1: Bit de configuración de la tensión de referencia VREF-:
 - VCFG1='0': VREF-se conecta a VSS
 - VCFG1='1': VREF-se conecta a la línea física RA2
- VCFG0: Bit de configuración de la tensión de referencia VREF+:
 - VCFG1='0': VREF+ se conecta a VDD

¹² Universidad Politécnica de Valencia. Microcontrolador PIC18F4550. p. 107

- VCFG1='1': VREF+ se conecta a la línea física RA3
- PCFG3..PCFG0: Bits configuración de los puertos de conversión A/D. Mediante estos bits se establecen las líneas físicas (RA5..RA0, RB4..RB0, RE1 y RE0) que van a trabajar como entradas del convertidor A/D.



- ADFM: Bit de configuración del tipo de almacenamiento del resultado de la conversión en los registros ADRESH y ADRESL:
 - ADFM='0': El resultado de la conversión se almacena con justificación a izquierdas
 - ADFM='1': El resultado de la conversión se almacena con justificación a derechas
- ACQT2..ACQT0 : Bits de configuración del tiempo de adquisición
- ADCS2..ADCS0: Bits selección de la señal de reloj del convertidor A/D. ¹³

5.2.6.1 SELECCIÓN DEL CANAL DE CONVERSION

Para que uno de los 13 canales pueda ser seleccionado, previamente debe haber sido configurado como entrada analógica mediante los bits PCFG3..PCFG0 del registro ADCON1 (A: analógico / D: digital).

Una vez configurado como línea de entrada analógica, un canal puede ser seleccionado mediante los bits CHS3. CHS0 del registro ADCON0.¹⁴

¹³ Universidad Politécnica de Valencia. Microcontrolador PIC18F4550. p. 110

¹⁴ Universidad Politécnica de Valencia. Microcontrolador PIC18F4550. p. 112

5.2.6.2 SEÑAL DE RELOJ DE CONVERSION

Se define T_{AD} como el tiempo de conversión de 1 bit. Una operación completa de conversión requiere un total de 11 T_{AD} para 10 bits.

La señal de reloj que genera las temporizaciones T_{AD} puede ser establecida mediante los bits ADCS2..ADCS0 del registro ADCON2. Existen dos fuentes para dicha señal de reloj:

- El oscilador principal
- Una red RC interna que incorpora el propio convertidor A/D. Esta red puede utilizarse cuando se deseen realizar conversiones en modos de bajo consumo. Esta red RC permite que se puedan llevar a cabo conversiones con el oscilador principal desactivado.

El valor de T_{AD} debe ser lo menor posible pero siempre superior al T_{AD} mínimo indicado en las hojas de datos de PIC18F4550: 0,7 us. Cuando se elige el oscilador RC interno el T_{AD} viene prefijado internamente y siempre cumple con la condición del valor mínimo.¹⁵

5.2.6.3 TIEMPO DE ADQUISICIÓN

La circuitería interna del convertidor A/D incorpora un condensador de muestreo. Antes de realizar una conversión debemos asegurarnos de que dicho condensador ha sido totalmente cargado a la tensión del canal seleccionado.

Cuando realizamos un cambio en la selección de canal debemos esperar un tiempo que dependerá de la impedancia de entrada del convertidor A/D y de la impedancia de salida del circuito sobre el que se está haciendo la conversión.

¹⁵ Universidad Politécnica de Valencia. Microcontrolador PIC18F4550. p. 115

Existen dos opciones para generar este retardo antes de comenzar la conversión:

- Por programa: se implementa un retardo software entre la selección del nuevo canal y el inicio de la conversión.
- Estableciendo un tiempo de adquisición automático: se programa un tiempo de adquisición que se establecerá de forma automática entre la orden de inicio de conversión y el muestreo de la señal para iniciar la conversión. Dicho tiempo puede ser programado mediante los bits ACQT2..ACQT0 del registro ADCON2. Este tiempo puede tener unos valores que oscilan entre $2 \cdot T_{AD}$ y $20 \cdot T_{AD}$.¹⁶

5.2.6 MODULO USB

El Bus Serie Universal fue creado en los años 90 por una asociación de empresas con la idea, entre otras, de mejorar las técnicas Plug and Play, es decir, permitir a los dispositivos conectarse y desconectarse sin necesidad de reiniciación, configurándose automáticamente al ser conectados, además se le dotó de transición de energía eléctrica para los dispositivos conectados.

Este bus tiene una estructura de árbol y se pueden ir conectando dispositivos en cadena, pudiéndose conectar hasta 127 dispositivos permitiendo la transferencia síncrona y asíncrona.

Se puede clasificar según su velocidad de transferencia de datos (desde kilobits hasta megabits): baja velocidad (1.0) utilizado para los dispositivos de interface humana (HID) como ratones, etc.; velocidad completa (1.1) y alta velocidad (2.0) para conexiones a Internet, etc.

¹⁶ Universidad Politécnica de Valencia. Microcontrolador PIC18F4550. p. 116

Físicamente, los datos del USB se transmiten por un par trenzado (D+ y D-) además de la masa y alimentación (+5V). Los conectores están sujetos al estándar (tipo A, tipo B).

5.2.6.1 USB CLASE CDC

Una clase USB es una agrupación de dispositivos de características comunes, es decir, utilizan una misma forma de comunicarse con el entorno. La clase de dispositivo permite conocer la forma en que la interfaz se comunica con el sistema, el cual puede localizar el driver que puede controlar la conectividad entre la interfaz y el sistema.

USB sólo permite al driver comunicarse con el periférico a través de las tuberías establecidas entre el sistema USB y los endpoints del periférico. Los tipos de transferencia a través de las pipes dependen del endpoint y pueden ser: bulk, control, interrupt e Isochronous. Una tubería es un enlace virtual entre el host y el dispositivo USB, donde se configura el ancho de banda, el tipo de transferencia, la dirección del flujo de datos y el tamaño del paquete de datos.

Estos enlaces se definen y crean durante la inicialización del USB. Un endpoint es un buffer dentro del dispositivo o periférico donde se almacenan paquetes de información, todos los dispositivos deben admitir el endpoint 0, el cual recibe el control y las peticiones de estado durante la enumeración del dispositivo. Cuando se conecta un dispositivo al host se produce la enumeración en la cual el host interroga al dispositivo sobre sus características principales, asignándole una dirección y permitiendo la transferencia de datos.

La especificación Clase de Dispositivos de Comunicación (CDC) define algunos modelos de comunicación, incluyendo la comunicación serie. Windows suministra el driver usber.sys para la especificación. Para la especificación CDC se necesitan dos interfaces USB, primero la interfaz Communication Class usando un IN

Interruptendpoint de interrupción y el segundo es la interfaz Data Class usando OUT bulkendpoint y un IN bulkendpoint. Esta interfaz es utilizada para transferir los datos que normalmente deberían ser transferidos a través de la interfaz RS-232.

Desde el punto de vista de sistema USB, el dispositivo puede tener distintas configuraciones, para cada una de las cuales puede funcionar de forma distinta. Los dispositivos suministran la información necesaria al sistema USB a través de los descriptores; estos contienen unos campos que permiten al sistema clasificar al dispositivo y asignarle un driver. La primera información que necesita es la del fabricante y producto (USB vendor ID – VID y el Product ID – PID). El VID es un número de 16 bits asignado por el USB implementers Forum (USB-IF) y debe ser obtenido por el fabricante del dispositivo USB, cada VID puede contener 65536 PID diferentes al ser también un número de 16 bits. Microchip suministra su VID y los PID para cada familia de PIC con USB.

Microsoft Windows (2000 o XP) no tiene un fichero *.inf estándar para el driver CDC, así que es necesario suministrar este fichero cuando se conecta un dispositivo USB por primera vez al sistema. Microchip suministra el fichero mchpcdc.inf necesario para sus dispositivos PIC.¹⁷

5.3 OPTOACOPLADOR

Los Optoacopladores u Optoaisladores son dispositivos que podemos encontrar en múltiples aplicaciones dentro de un equipo electrónico, cuando una señal debe ser transmitida desde un circuito específico a otro, sin que exista conexión eléctrica entre ambos.

A pesar de que un optoacoplador o “acoplador de señales eléctricas mediante un enlace óptico” puede tomar formatos físicos muy variados, su arquitectura es

¹⁷ García Breijo, Eduardo. Compilador CCS y simulador Proteus para microcontroladores PIC. p. 252

siempre reiterada en el concepto fundamental. Por un lado, se utiliza para transmitir la información un diodo LED infrarrojo, construido a base de un compuesto de Arseniuro de Galio, que se enlaza en forma óptica con un detector encargado de capturar esa información luminosa y transformarla en una señal eléctrica idéntica, en su composición de niveles, a la que el LED emite. Luego, la naturaleza de este detector nos brindará una respuesta acorde al tipo de señal aplicada al LED y a la función específica para la que fue construido ese detector que trae consigo el optoacoplador.

Por ejemplo, si el elemento receptor (o detector) es un fototransistor, podremos utilizar el dispositivo para transferir señales analógicas como puede ser audio o video. Si en cambio es otro fotodiodo, o un foto-SCR, nos será útil como “rectificador controlado y aislado eléctricamente”. De este modo, los detectores se multiplican en formatos y tipo de aplicación, como puede ser un Triac (para trabajar con corrientes alternas) y hasta podemos encontrar puertas lógicas, como detectores dentro de un optoacoplador. Lo que siempre conservará su naturaleza es el elemento transmisor o emisor; siempre será un diodo (o un conjunto de ellos) LED infrarrojo.

Otro tipo de acoplador óptico muy popular es el que trae una forma de “U” y su activación se basa, ya no en un reflejo, sino en una interrupción de un haz detectado en forma permanente (en el tercer video hacemos un ensayo con uno).¹⁸

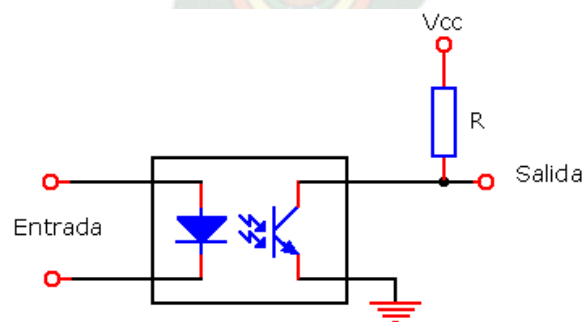


Fig.6 Esquema eléctrico de optoacoplador en U.

¹⁸ Sacca, Mario. Optoacopladores



Fig. 7 Forma externa de optoacoplador en U

5.4 POTENCIOMETRO 360°

Son potenciómetros que nos dan la posibilidad de girar 360° y cambiar su resistencia de forma lineal, cuando se encuentra en su máximo valor vuelve a su mínima resistencia si el giro continúa.

POTENCIOMETROS 360°							
Resolucion infinita - Larga vida - Alta calidad - Robustos							
CODIGO	RESISTENCIA (Ohms)	TOLERANCIA	POTENCIA (W)	MATERIAL	ANGULO	LARGO DEL EJE (mm)	DIMENSIONES (MM)
6639S102	1 K	±15%	1.0	CERMET	340°+8°	15	22x11x35
6639S202	2 K						
6639S502	5 K						
6639S103	10 K						
6639S203	20 K						

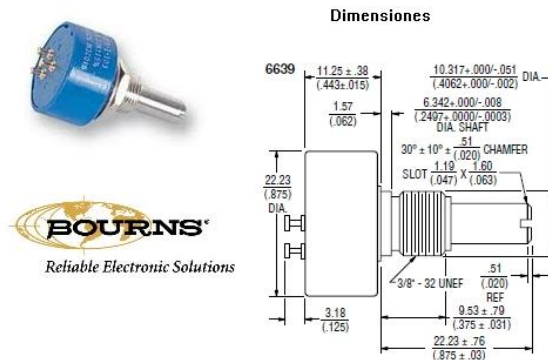
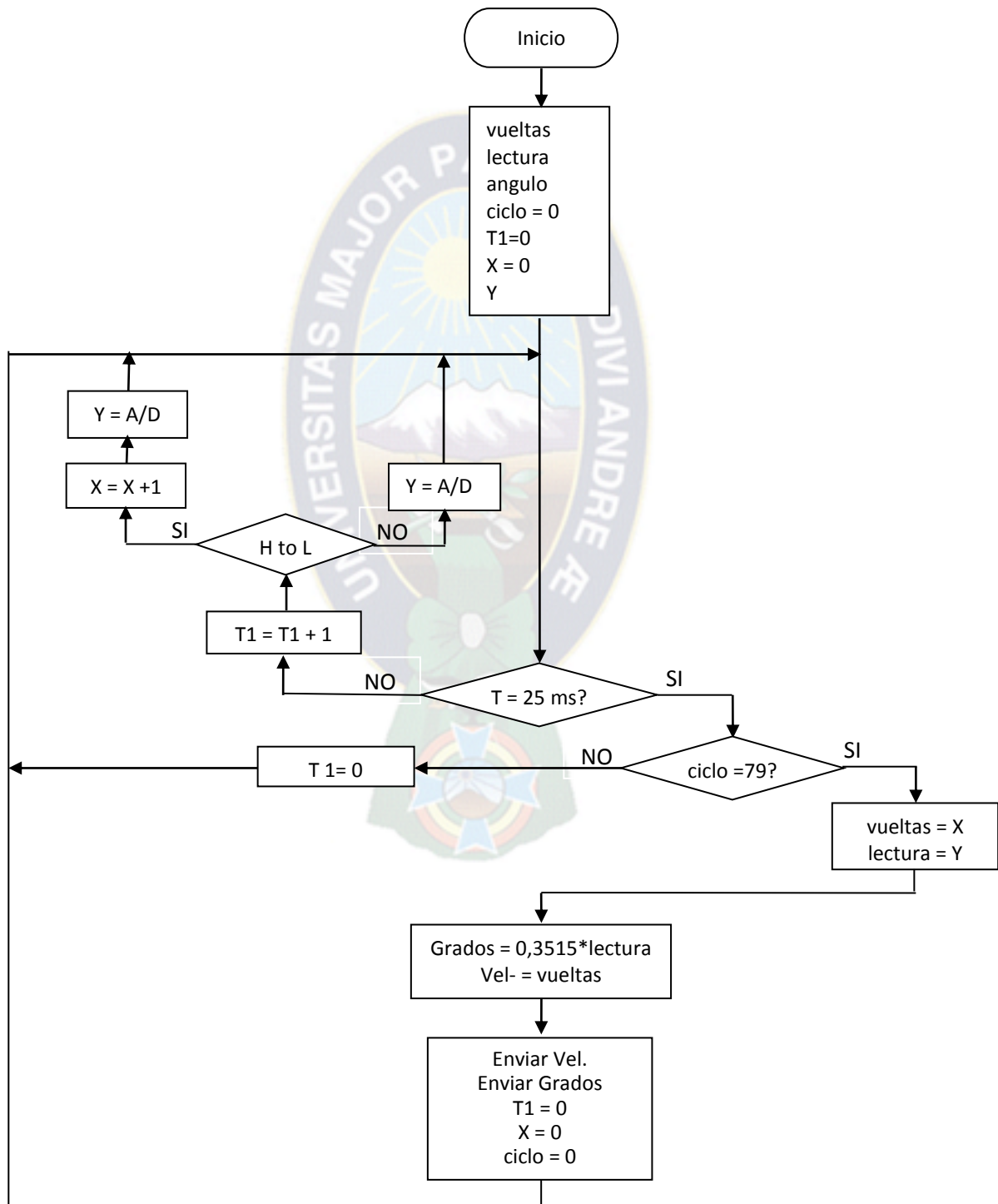


Fig. 8 Aspecto externo y datos del potenciómetro de 360°.

6. DESARROLLO DEL TRABAJO

6.1 ALGORITMO DEL PROGRAMA

El diagrama de flujos del programa:



A continuación indicaremos los pasos que seguimos para la elaboración del programa:

- 1er Paso. Configurar un temporizador para periodos de 2 segundos.
- 2do Paso. Utilizar un contador capaz de detectar flancos de bajada.
- 3er Paso. Conversor A/D detectando niveles de voltaje.
- 3er Paso. Al finalizar los 2 segundos obtener el número de vueltas y el voltaje.
- 4to Paso. Calcular la velocidad y la dirección del viento.
- 5to. Paso. Enviar los datos.
- 6to. Paso. Contador puesto a cero volver al primer paso.

6.2 PROGRAMACIÓN DEL PIC

El compilador CCS convierte el lenguaje de alto nivel en instrucciones de código máquina, es el compilador con el que realizamos el programa para el PIC dado que cuenta con librerías que facilitan el uso de las funciones con los que cuenta el PIC 18f4550.

6.2.1 PROGRAMACIÓN DEL TIMER1

Lo primero que se hizo es el cálculo del Timer1 en modo de temporizador, para este fin utilizamos la ecuación (6) de desbordamiento.

$$T = \frac{4}{F_{osc}} \times Prescaler \times (65536 - CargaTimer1)$$

El cálculo se lo realiza para un intervalo de 25 ms, con un pre escalador de 8, F_{osc} de 20 MHz.

$$0,025 = \frac{4}{48 Mhz} \times 8 \times (65536 - Ct1)$$

$$CT1 = 65536 - \frac{0,025 \times 48 \text{ MHz}}{4 \times 8} = 28036$$

Para que los datos mostrados no sean rápidos, el tiempo de 25 ms de lo multiplicó por 80 para tener intervalos de 2 segundos.

$$T_1 = 0,025ms \times 80 = 2 \text{ seg}$$

Las instrucciones utilizadas en compilador CCS, son las siguientes:

```
setup_timer_1(t1_internal | t1_div_by_8); // configuración del Timer1.
enable_interrupts(int_timer1); // habilitación del Timer1.
```

Y la rutina de desbordamiento del Timer1:

```
void timer1_isr(void)
{
  if (ciclo==79)
  {
    // En esta parte viene los cálculos y el envío de los datos.
  }
  ciclo ++;
  set_timer1 (28036); // recarga del Timer 1.
}
```

6.2.2 PROGRAMACIÓN DEL TIMER0

Timer0 se lo configuro en modo contador por RA4, pasado los 2 segundos, o sea 80 desbordamientos del Timer1, se saca la cuenta y se lo vuelve a poner a 0.

Es importante encontrar el valor de la constante que multiplicará a N, número de vueltas, para tener la velocidad del viento en Km/h. De la ecuación (3) se tiene:

$$Vv = 2\pi r \frac{n}{t}$$

El radio del anemómetro es de 8,7 cm y el intervalo de tiempo en el que se toma cada muestra es de $t = 2(\text{seg})$.

$$Vv_{\left(\frac{Km}{h}\right)} = 2\pi \times 8.7cm \times \frac{N}{2\text{seg}} \times \frac{1\text{ m}}{100\text{ cm}} \times \frac{1\text{ Km}}{1000\text{ m}} \times \frac{3600\text{ seg}}{1\text{ h}} = 0,98N \cong N$$

Las instrucciones realizadas en el compilador CCS son las siguientes:

```
setup_timer_0(rtcc_ext_h_to_1 | rtcc_div_2 | rtcc_8_bit); // configuración en
                                                    flanco de bajada.
set_timer0(0); //colocamos el contador en 0.
enable_interrupts(int_timer0); //habilitación del timer0.
```

En la rutina de desbordamiento del Timer1 se realiza el cálculo y envío de los datos, con las siguientes instrucciones:

```
vueltas=get_timer0();// tomamos el número de vueltas N.
printf(usb_cdc_putc,"VEL=%lu km/h_", vueltas); // si está conectado USB
                                                    envió de la velocidad
```

6.2.3 PROGRAMACIÓN DEL ADC

Para el cálculo de la dirección se utilizó el conversor analógico digital, cuya resolución depende de los 10 bits, el cálculo de la constante por la que se multiplicará la lectura del conversor es el siguiente:

$$Resolución = \frac{360}{2^{10}} = 0,3515$$

Las instrucciones para la configuración del conversor ADC en CCS se la hicieron de la siguiente manera:

```
setup_adc_ports(AN0|VSS_VDD), // configuración del conversor..
setup_adc(ADC_CLOCK_INTERNAL); // elección del reloj interno
set_adc_channel(0); // selección del canal analógico
```

Los cálculos para determinar el ángulo de incidencia del viento se lo realizo de la siguiente manera:

```
lectura=read_adc(); // tomamos el valor digital.
angulo=(35.15*lectura)/100; // calculamos el ángulo.
printf(usb_cdc_putc,"DIR=%3.0f Grados",angulo); // envió de datos
```

6.2.4 PROGRAMACIÓN DEL PERIFERICO USB

Para utilizar el puerto USB en CCS, debemos incluir la librería `usb_cdc.h`, que contiene el driver necesario para una conexión clase CDC. Cabe mencionar que dentro de esta librería se deben hacer los cambios del ID vendor a `0XD8,0x04` y el ID Product a `0X0A,0X00`; que son los números suministrados por los fabricantes del pic, la empresa Microchip.

Las instrucciones necesarias para manejar el puerto Usb son las siguientes:

```
#include <usb_cdc.h> // incluir librería usb_cdc.h
#define USB_CON_SENSE_PIN PIN_A2 // Si UBS está conectado
usb_init(); // iniciar USB
usb_cdc_init(); // 7 iniciar clase cdc.
while(!usb_cdc_connected()); // mientras USB esté conectado.
usb_task(); // pregunta periódica de USB conectado.
if (usb_enumerated()) // si USB ha sido enumerado para envió de datos.
```


6.2.5 PROGRAMA COMPLETO EN CCS

```
#include <18F4550.h>
#device adc=10
#use delay(clock=48000000)
#fuses
HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,VRE
GEN
#define USB_CON_SENSE_PIN PIN_A2
#include <usb_cdc.h>

int16 vueltas;
int8 ciclo=0;
int16 lectura;
float angulo;

#int_timer1
void timer1_isr(void)
{
if (ciclo==79)
{
    vueltas=get_timer0();
    lectura=read_adc();
    angulo=(35.15*lectura)/100;
    usb_task();
    set_timer1(28036);
    set_timer0(0);
    if (usb_enumerated())
    {
        printf(usb_cdc_putc,"VEL=%lu km/h_", vueltas);
        printf(usb_cdc_putc,"DIR=%3.0f Grados",angulo);
    }
}
}
```



```
    }
    ciclo=0;
}
ciclo++;
set_timer1 (28036);
}

void main()
{
    setup_timer_0(rtcc_ext_h_to_l | rtcc_div_2 | rtcc_8_bit);
    setup_timer_1(t1_internal | t1_div_by_8);
    setup_adc_ports(AN0|VSS_VDD),
    setup_adc(ADC_CLOCK_INTERNAL);
    set_adc_channel(0);
    usb_init();
    usb_cdc_init();
    while(!usb_cdc_connected());

    set_timer0(0);
    set_timer1(28036);
    enable_interrupts(int_timer0);
    enable_interrupts(int_timer1);
    enable_interrupts(global);
while(1)
    {
    }
}
```



6.3 PROGRAMA EN LA PC CON VISUAL BASIC 6.0

En el programa Visual Basic se lo configuro de modo que reciba los datos de velocidad y dirección que mande el PIC a un textbox, cabe mencionar que se uso MSComm que emula una comunicación serie.

```
Private Sub Command2_Click()
```

```
Text2.Text = ""
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
MSComm1.PortOpen = True
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
MSComm1.PortOpen = False
```

```
End Sub
```

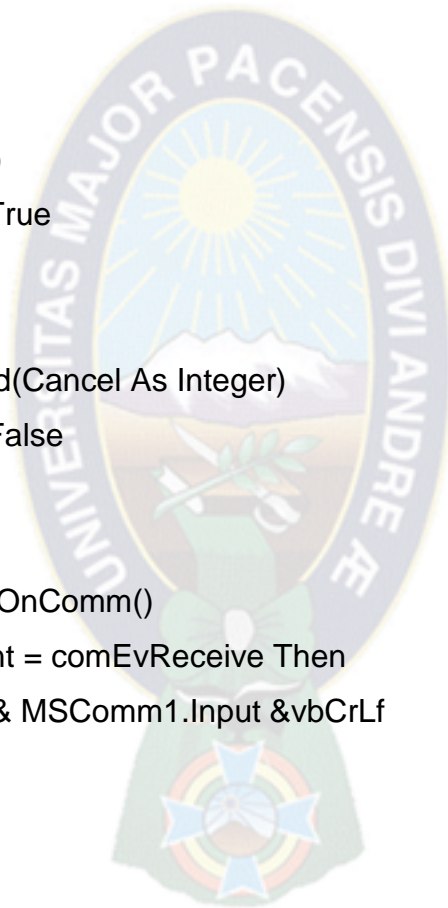
```
Private Sub MSComm1_OnComm()
```

```
If MSComm1.CommEvent = comEvReceive Then
```

```
Text2.Text = Text2.Text & MSComm1.Input &vbCrLf
```

```
EndIf
```

```
End Sub
```



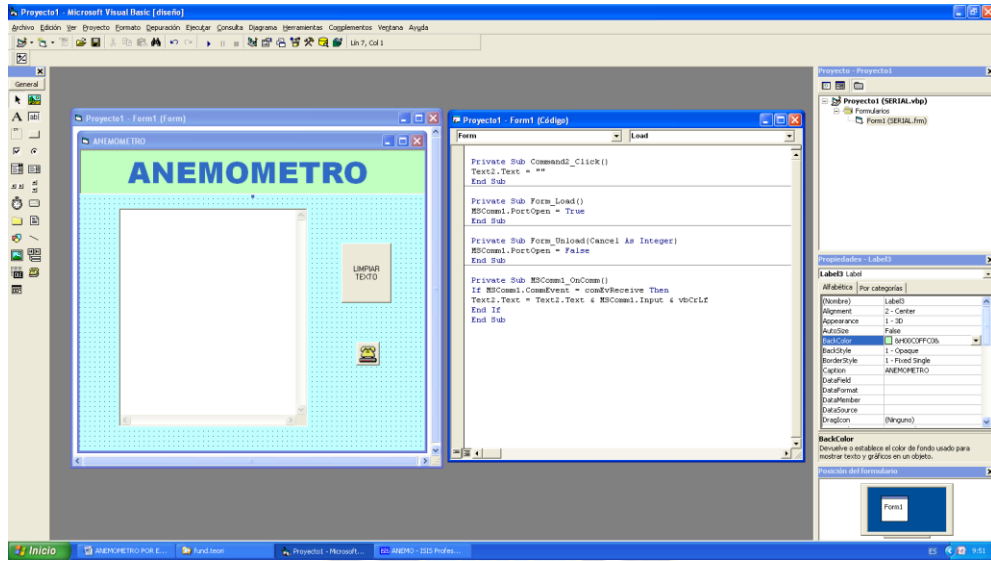


Fig. 9 Programa diseñado en Visual Basic 6.0

6.4 CIRCUITO DEL ANEMÓMETRO

Cabe mencionar que para el circuito del anemómetro se debe colocar una capacitor en el pin 18 (Vusb), para que el Plc18f4550 tome el voltaje de referencia USB.

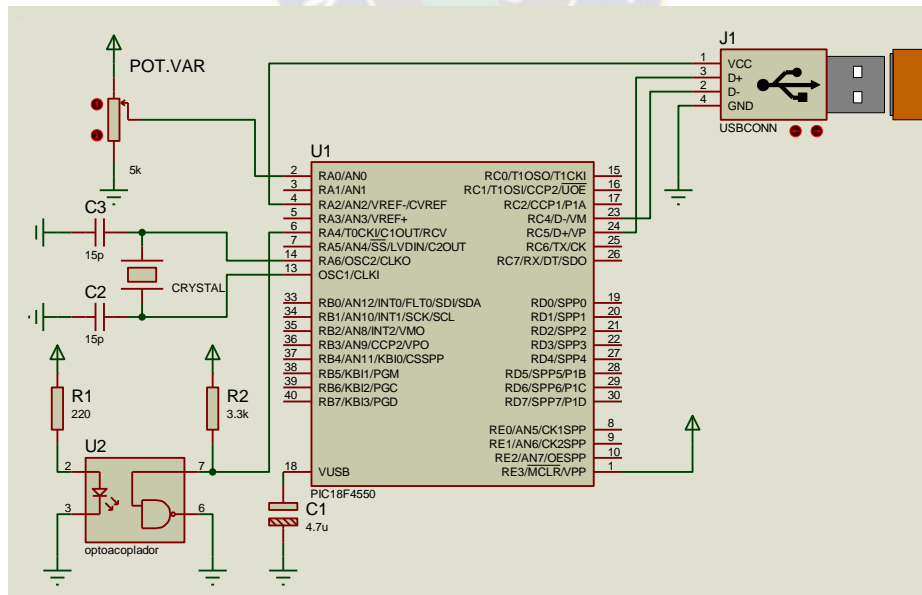


Fig. 10 Circuito elaborado en Proteus

6.5 ASPECTO FÍSICO

El elemento mecánico escogido para que el viento gire las aletas del anemómetro es un piñón móvil colocado sobre una caja de plancha. El piñón ingresa al interior de la caja, en cuyo extremo se colocó una tarjeta en forma de semicírculo. Dicha tarjeta tiene el fin de ingresar al centro del optoacoplador en forma de U, para que impida la llegada de la luz, que genera el diodo, sobre el transistor interno del acoplador. De esta manera se genera el cambio de estado a la salida del optoacoplador por cada vuelta que dé el piñón.

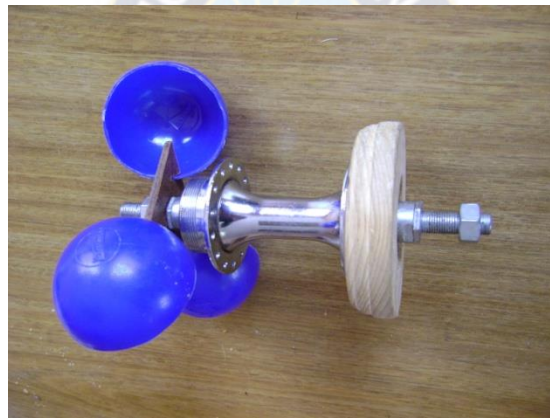


Fig. 11 Piñón usado para el anemómetro.



Fig. 12 Piñón dentro el anemómetro.

Para la elaboración de la veleta encargada de medir el ángulo de incidencia del viento se usó el mismo potenciómetro sin fin, cuyo extremo que gira se encuentra por encima de la caja metálica, así es el viento golpea contra ella generando el ángulo que necesitamos medir.

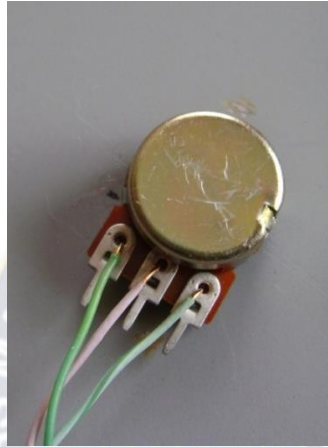


Fig. 13 Potenciómetro encargado de medir el ángulo.



Fig.14 Vista exterior del anemómetro.

6.6 FUNCIONAMIENTO DEL ANEMOMETRO

El anemómetro se conecta a la PC a través del puerto USB, una vez ocurrido esto, la PC reconocerá al Plc 18f4550 y nos pedirá direccionar la ubicación del driver;

para este fin direccionamos a la ubicación donde se encuentre el fichero mchpcdc.inf.

Una vez concluido este proceso nos dirigimos al administrador de dispositivos de la PC para verificar la conexión del Pic. Buscamos en los puertos com, ya que es una emulación de puerto serie, en nuestro caso reconoció al Pic como comm7.

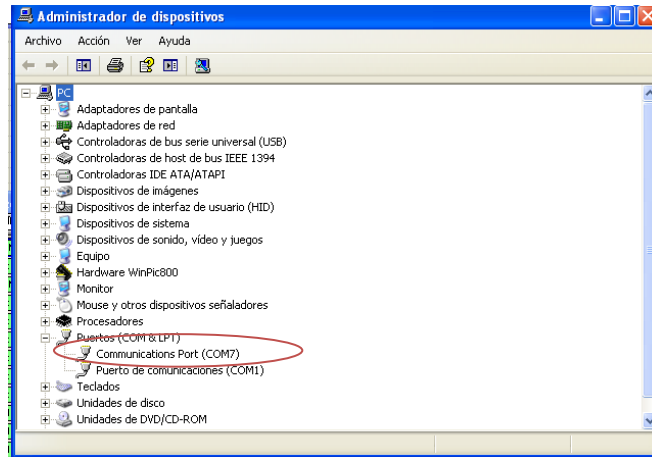


Fig.15 Reconocimiento del Pic en la PC.

Posteriormente se procede a abrir el programa de recepción de datos en Visual Basic, donde configuramos el puerto Com al mismo número con el cual reconoció al Pic la PC, en nuestro caso Cm7.

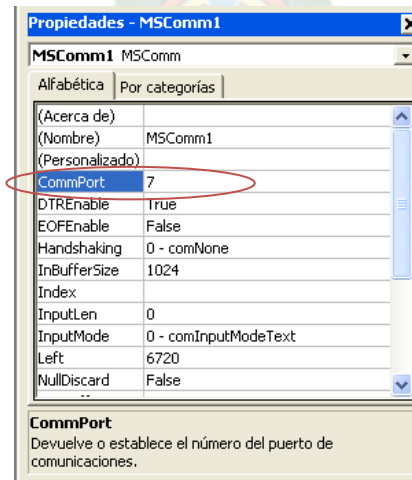


Fig. 16 Puerto Com7 en Visual Basic.

Iniciamos el programa y los datos deben ser recibidos, en el textbox que fue configurado para este fin.

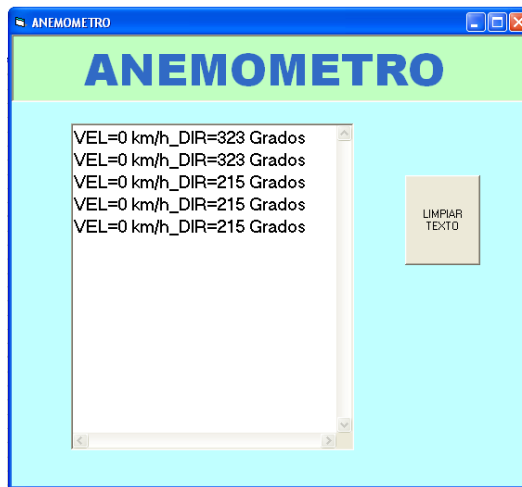


Fig.17 Visual Basic recibiendo datos del anemómetro.

6.7 COSTOS DEL PROYECTO

Los costos invertidos en el siguiente proyecto fueron en dispositivos electrónicos y en la parte de implementación del proyecto. A continuación se detalla los costos del proyecto:

CANT.	DETALLE	COSTO (Bs)
1	PIC 18F4550	60,00
1	Oscilador de cristal 20 MHz	5,00
1	Capacitor 50V - 10 uF	2,00
1	Optoacoplador	6,00
1	Potenciómetro 360° 5K	18,00
1	Resistencia 1/4 W 220 Ohm	0,20
1	Resistencia 1/4 W 3.3K	0,20
1	Caja metálica	35,00
1	Píñon	25,00
1	Cable USB	5,00
2	Capacitores 15 pF	2,00
1	Placa y quemado	30,00
TOTAL		188,40

8. CONCLUSIONES

Se logro diseñar e implementar un anemómetro por el puerto USB en base al microcontrolador PIC18F4550 cuya conectividad con la PC es la clase cdc, este anemómetro es capaz de obtener datos sobre la dirección y velocidad del viento.

Los recursos utilizados del pic18f4550 fueron el Timer0 como contador, de vueltas el Timer1 como temporizador con un tiempo de desborde de 0,025 ms en intervalos de 2 segundos, y el conversor analógico digital con una resolución de 10 bits el cual nos proporcionó 1024 distintos niveles para la conversión de la dirección del viento.

Los sensores empleados fueron debidamente configurados tanto el optoacoplador, capaz de brindarnos información sobre las vueltas que da el anemómetro; y el potenciómetro sin tope, el cual nos brindó información, en grados, de la dirección del viento.

El programa elaborado en Visual Basic 6.0 es capaz de recibir los datos enviados por el plc 18f4550.

BIBLIOGRAFÍA

- Angulo M. José; Romero, Susana; Ángulo, Ignacio (2004). Diseño práctico con microcontroladores. Thomson Editores Spain. Paraninfo S.A. España.
- Garcia Breijo, Eduardo (2008) Compilador C CSS y simulador Proteus para microcontroladores PIC. Ediciones Técnicas Marcombo. México
- G.M. Electrónica S.A.(n.d.). Potenciómetros 360°. Obtenida el 20 de Noviembre de 2012 de:
<http://www.gmelectronica.com.ar/catalogo/pag86a.html>.
- Microchip, PIC18F4550 Datasheet (n.d.). Obtenido el 30 de octubre de 2012 de: www.microchip.com.
- Sacca, Mario (n.d.). Optoacopladores. Obtenida el 12 de noviembre de 2012 de: <http://www.neoteo.com/optoacopladores-electronica-basica>
- Salmerón Gonzales, Sergio. (n.d.) La luz y la electrónica infrarrojos. Obtenida el 10 de noviembre de 2012 de:
<http://www.datasheetcatalog.org/datasheet/vishay/83751.pdf>
- Universidad Politécnica de Valencia (n.d.). Microcontrolador PIC18F4550. Obtenida el 05 de noviembre de 2012 de:
<http://www.pinguino.org.ve/descargas/Manual%20PIC%2018F4550.pdf>

ANEXOS

MICROCONTROLADORES PIC18F2455, PIC18F2550, PIC18F4455 y PIC18F4550

CARACTERISTICAS	PIC18F2455	PIC18F2450	PIC18F4455	PIC18F4450
Frecuencia de Operación	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz
Memoria de Programa (bytes)	24.576	32.768	24.576	32.768
Memoria RAM de Datos (bytes)	2.048	2.048	2.048	2.048
Memoria EEPROM Datos (bytes)	256	256	256	256
Interrupciones	19	19	20	20
Líneas de E/S	24	24	35	35
Temporizadores	4	4	4	4
Módulos de Comparación/Captura/PWM (CCP)	2	2	1	1
Módulos de Comparación/Captura/PWM mejorado (ECPP)	0	0	1	1
Canales de Comunicación Serie	MSSP,EUSART	MSSP,EUSART	MSSP,EUSART	MSSP,EUSART
Canal USB	1	1	1	1
Puerto Paralelo de Transmisión de Datos (SPP)	0	0	1	1
Canales de Conversión A/D de 10 bits	10 Canales	10 Canales	13 Canales	13 Canales
Comparadores analógicos	2	2	2	2
Juego de instrucciones	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)
Encapsulados	PDIP 28 pines SOIC 28 pines	PDIP 28 pines SOIC 28 pines	PDIP 40 pines QFN 40 pines TQFP 40 pines	PDIP 40 pines QFN 40 pines TQFP 40 pines

Características del PIC 18F4550

▪ Grupo general de interrupciones:

Interrupción del Temporizador 0
Interrupción por cambio en PORTB
Interrupción externa 0
Interrupción externa 1
Interrupción externa 2

▪ Grupo de interrupciones de periféricos

Interrupción del SPP	Interrupción de fallo del oscilador
Interrupción del A/D	Interrupción del comparador
Interrupción de recepción de la EUSART	Interrupción del USB
Interrupción de transmisión de la EUSART	Interrupción de escritura en Flash/EEPROM
Interrupción del MSSP	Interrupción de colisión de bus (MSSP)
Interrupción del CCP1	Interrupción del detección de anomalías en V _{DD}
Interrupción del Temporizador 2	Interrupción del Temporizador 3
Interrupción del Temporizador 1	Interrupción del CCP2

Sistema de interrupciones del PIC 18F4550

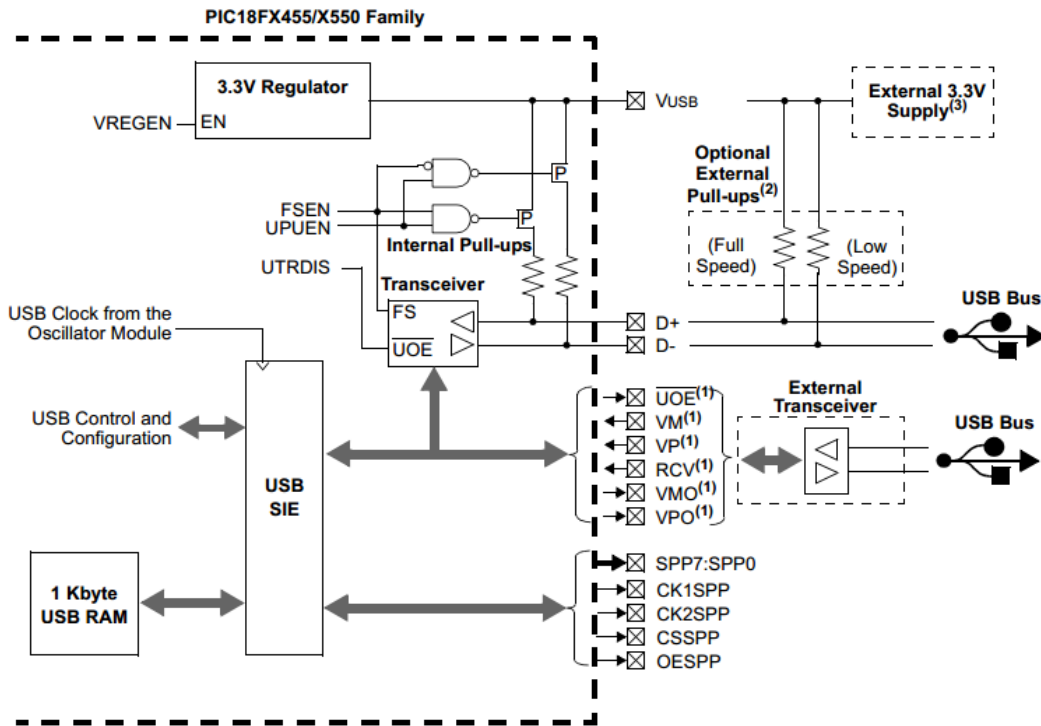
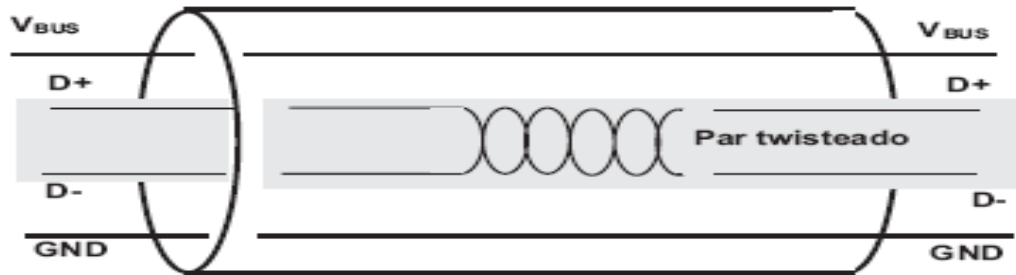


Diagrama en bloques del periférico USB PIC 18f4550



NUMERO CONTACTO	NOMBRE DE LA SENAL	COLOR IDENTIFICATIVO
1	Vbus	ROJO
2	D-	BLANCO
3	D+	VERDE
4	GND	NEGRO
PANTALLA	PANTALLA	PANTALLA

Señales físicas USB