



INSTITUTO TECNOLÓGICO DE TUXTLA GUTIERREZ

RESIDENCIA PROFESIONAL

***SISTEMA “USB” DE ADQUISICION DE DATOS DE
UNA ESTACION AGROCLIMATICA.***

ALUMNO: OSCAR DE JESUS LEON VAZQUEZ

N° DE CONTROL: 03270382

ASESOR: DR. HECTOR RICARDO HERNANDEZ DE LEON

REVISOR: ROGER EDUARDO CASTELLANOS GALDAMEZ

TUXTLA GUTIERREZ, CHIAPAS. DICIEMBRE DE 2008

I N D I C E

1.- INTRODUCCION	1
2.-JUSTIFICACION	1
3.- OBJETIVOS	2
3.1 OBJETIVO GENERAL	2
3.2 OBJETIVOS ESPECIFICOS	2
4.-CARACTERIZACION DEL AREA EN QUE PARTICIPO	3
5.-DATOS DE LA EMPRESA	3
6.- PROBLEMAS A RESOLVER	4
7.- ALCANCES Y LIMITACIONES	4
8.- FUNDAMENTO TEORICO	6
8.1.- INTERFACES SERIE	6
8.2.- TIPOS DE COMUNICACIONES SERIALES	7
8.3.- NORMA RS-232	9
8.4.- CONECTOR DE LA COMUNICACIÓN SERIAL	9
8.5.- PUERTO SERIAL EN NULL	12
8.6.- CONEXIÓN DE UN μC Y LA PC	13
8.7.- EL CIRCUITO MAX-232	13
9.- EL BUS SERIE UNIVERSAL (USB)	14
10.- MICROCONTROLADORES	18
10.1.- FAMILIA DE MICROCONTROLADORES18F	19
10.2.- PINES DEL 18F4550	20
10.3.- ORGANIZACIÓN DE MEMORIA	20
10.4.- ARQUITECTURA HARDVARD	21
10.5.- MEMORIA RAM DE DATOS	23
10.6.- SISTEMAS DE ADQUISICIÓN DE DATOS MEDIANTE SENSORES	23
11. -REALIZACION DEL HARDWARE	25
12.- USB vs. FireWire	25
13.- QUE ES UNA ESTACION AGROCLIMATICA	28
14.-TIPOS DE ESTACIONES	28

15.- ESTADO DEL ARTE	29
15.1.- VARIABLES CLIMÁTICAS MEDIDAS	29
15.2.- APLICACIONES	30
15.3.- CRECIMIENTO DE LA RED	30
16.- SENSORES QUE SE USAN EN LAS ESTACIONES AGROCLIMATICAS Y SUS CARACTERISTICAS TECNICAS	32
16.1.- SENSOR DE HUMEDAD	32
16.1.1.- SENSOR RESISTIVO	32
16.1.2.- SENSOR CAPACITIVO HC-201	32
16.1.3.- SENSOR INTEGRADO	33
16.2.- SENSOR DE PRESION BAROMETRICA	33
16.3.- SENSOR DE PRECIPITACION	33
17.- PAISES DONDE EXISTEN ESTACIONES AGROCLIMATICAS DE ACUERDO A SU TIPO	36
17.1.- ESTACIONES WMO-GTS	36
17.2.- ESTACIONES VOLMET	37
17.3.- OTRAS ESTACIONES	38
18.- DISEÑO Y COMPONENTES DEL SISTEMA	38
18.1.- SENSOR DE TEMPERATURA LM35	38
19.- CONCLUSION Y RECOMENDACIONES	40
20.- REFERENCIAS BIBLIOGRAFICAS	41
21.- ANEXOS	42

1.-INTRODUCCION

Debido al desarrollo inmediato de las sociedades así como de la globalización, han creado como resultado la constante competitividad por parte de diferentes países así como de distintas empresas las cuales se esmeran por realizar y sofisticar cada día aparatos que faciliten las actividades del ser humano, por ello hoy en día podemos ver como a través del tiempo han venido cambiando los diferentes dispositivos capaces de guardar información dentro de ellos como el disco flexible , el CD, etc.; hasta llegar a nuestros días donde gracias al avance inmediato de la tecnología podemos ver gran variedad de dispositivos que nos son de mucha ayuda, uno de los mas importante el USB; el diseño del USB tenía en mente eliminar la necesidad de adquirir tarjetas separadas para poner en los puertos bus ISA o PCI, y mejorar las capacidades plug-and-play permitiendo a esos dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciar. Cuando se conecta un nuevo dispositivo, el servidor lo enumera y agrega el software necesario para que pueda funcionar.

2.-JUSTIFICACION

El **Universal Serial Bus** (bus universal en serie) o **Conductor Universal en Serie**, abreviado comúnmente **USB**, es un puerto que sirve para conectar periféricos a una computadora. Fue creado en 1996 por siete empresas: IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC.

El estándar incluye la transmisión de energía eléctrica al dispositivo conectado. Algunos dispositivos requieren una potencia mínima, así que se pueden conectar varios sin necesitar fuentes de alimentación extra. La gran mayoría de los concentradores incluyen fuentes de alimentación que brindan energía a los dispositivos conectados a ellos, pero algunos dispositivos consumen tanta energía que necesitan su propia fuente de alimentación. Los concentradores con fuente de alimentación pueden proporcionarle corriente eléctrica a otros dispositivos sin quitarle corriente al resto de la conexión (dentro de ciertos límites).

El estándar USB especifica tolerancias para impedancia y de especificaciones mecánicas relativamente bajas para sus conectores, intentando minimizar la incompatibilidad entre los conectores fabricados por distintas compañías. Una meta a la que se ha logrado

llegar. El estándar USB, a diferencia de otros estándares también define tamaños para el área alrededor del conector de un dispositivo, para evitar el bloqueo de un puerto adyacente por el dispositivo en cuestión.

Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

- Baja velocidad (1.0): Tasa de transferencia de hasta 1'5 Mbps (192 KB/s). Utilizado en su mayor parte por dispositivos de interfaz humana (Human interface device, en inglés) como los teclados, los ratones y los joysticks.
- Velocidad completa (1.1): Tasa de transferencia de hasta 12 Mbps (1'5 MB/s). Ésta fue la más rápida antes de la especificación USB 2.0, y muchos dispositivos fabricados en la actualidad trabajan a esta velocidad. Estos dispositivos dividen el ancho de banda de la conexión USB entre ellos, basados en un algoritmo de búferes FIFO.
- Alta velocidad (2.0): Tasa de transferencia de hasta 480 Mbps (60 MB/s).
- Súper velocidad (3.0): Actualmente en fase experimental y con tasa de transferencia de hasta 4.8 Gbps (600 MB/s).
-

3.-OBJETIVOS

3-1.-OBJETIVO GENERAL.

Diseñar e implementar un circuito digital basado en microcontroladores que permita la emisión y recepción de datos vía USB.

3-2.-OBJETIVOS ESPECIFICOS.

- 1.-Lograr la comunicación vía USB del emisor al receptor.
- 2.-Obtención de datos específicos gracias a la comunicación existente del emisor al receptor.
- 3.-Manipulación fácil de los datos obtenidos.

4.-CARACTERIZACIÓN DEL ÁREA EN QUE PARTICIPÓ.

Con la finalidad de realizar el proyecto con la calidad suficiente se me fue asignado el cubículo I-15 correspondiente a la obtención de circuitos impresos, el cual se encuentra ubicado en las instalaciones del edificio I de la carrera de ingeniería electrónica del instituto tecnológico de Tuxtla Gutiérrez, donde se me brindo el apoyo necesario para concluir con el proyecto de residencia proporcionándome el material necesario para realizar pruebas específicas de dicho proyecto.

5.- DATOS DE LA EMPRESA

La empresa donde se desarrolló el proyecto se describe a continuación:

1. **Nombre:** Instituto Tecnológico de Tuxtla Gutiérrez
2. **Ubicación:** Carretera panamericana Km. 1080 en la ciudad de Tuxtla Gutiérrez, Chiapas, México.
3. **Código postal:** 29000, apartado postal 599
4. **Telefonos:** (961)61-5-03-80 y (961) 61-5-04-61
5. **Fax:**(961) 61-5-16-87

Misión

Formar de manera integral profesionalistas de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora, respeto al medio ambiente y apego a los valores éticos.

Visión

Ser una institución de excelencia en la educación superior tecnológica del Sureste, comprometida con el desarrollo socioeconómico sustentable de la región.

Valores

1. El ser humano
2. El espíritu de servicio

3. El liderazgo
4. El trabajo en equipo
5. La calidad
6. El alto desempeño

Las carreteras con que cuenta el instituto son:

1. Ing. Química
2. Ing. Electrónica
3. Ing. Eléctrica
4. Ing. Mecánica
5. Ing. Industrial
6. Ing. Bioquímica
7. Ing. En Sistemas Computacionales
8. Ing. En Gestión Empresarial
9. Lic. En Informática

6.-PROBLEMAS A RESOLVER, PRIORIZÁNDOLOS.

A pesar de que el sistema fue realizado en distintas etapas, conforme fui avanzando en dicha realización me tope con algunos problemas, uno de los mas sobresaliente fue la de cómo pasar la información del PIC a la PC para que el microcontrolador realizara la función especificada, desplegando la información recibida en dicho PIC que obtiene sus datos por medio de una comunicación usb.

7.-ALCANCES Y LIMITACIONES.

Con el desarrollo del sistema se pretende implementar en hardware mediante un microcontrolador y la PC, algoritmos de emisión y recepción.

La comunicación entre la PC y el microcontrolador, se implementa mediante el puerto serial y el USB, la comunicación por el puerto USB se realiza con un adaptador.

Así pues se pretende alcanzar totalmente la recepción exacta transmitida por el emisor logrando así como resultado la creación de una señal indicada a través de la pc.

Dentro de las limitaciones se puede mencionar el aspecto del alcance de dicha emisión del dispositivo cuya responsabilidad recae en el dispositivo emisor.

Así como la recepción que se pretende llegue de manera correcta a la pc para indicarnos lo que pasa del lado donde se encuentra el emisor.

8.- FUNDAMENTO TEÓRICO

8.1.- INTERFACES SERIE

La conexión a través de esta interfaz es muy importante debido a su gran flexibilidad. En los ordenadores personales la interfaz serie se utiliza para conectar múltiples dispositivos como plotters, módems, ratones y también impresoras.

En la transmisión serie se van transfiriendo los bits de información de uno en uno a través de una línea de datos, pudiendo ser las transferencias síncronas o asíncronas.

Si se utilizan señales adicionales (reloj o señales de petición y reconocimiento) para indicar cuándo el bit siguiente es válido, entonces se dice que la transmisión se realiza de forma síncrona. La principal ventaja de este tipo de transferencias es que el receptor puede funcionar a varias frecuencias de reloj (siempre que no sobrepase su frecuencia máxima de funcionamiento). Simplemente bastará con retrasar el envío de la señal de reconocimiento para ralentizar el protocolo. En las transferencias asíncronas, por el contrario, tanto el receptor como el transmisor deben funcionar a la misma frecuencia. En este caso se envía también información de sincronización a través de la línea de datos, que se corresponde con un bit de comienzo (bit de start), que indica el comienzo de una unidad de datos, un bit de fin (bit de stop) indicando su finalización y, opcionalmente, un bit de paridad para controlar los posibles errores.

El bit de paridad lo generan los controladores serie de forma automática, pudiendo configurarse entre las opciones de: sin paridad, paridad par (odd), paridad impar (even), siempre un nivel alto (mark) o siempre un nivel bajo (space).

Las tasas de transferencia de datos se miden en baudios. Los baudios indican el número de veces que puede cambiar una señal en la línea de transmisión por segundo.

En una interfaz serie, las señales cambian siempre a la misma frecuencia y se realiza una codificación binaria de la información de forma que cuando se quiere enviar un '1' se pone la línea a nivel alto y cuando se quiere enviar un '0' se pone la línea a nivel bajo. En este caso

los baudios coinciden con el número de bits por segundo transferidos si se incluyen también los bits de comienzo, de fin y de paridad.

Existen dos formas de intercambiar información binaria: la paralela y la serial. La comunicación paralela transmite todos los bits de un dato de manera simultánea, por lo tanto la velocidad de transferencia es rápida, sin embargo tiene la desventaja de utilizar una gran cantidad de líneas, por lo tanto se vuelve mas costoso y tiene las desventaja de atenuarse a grandes distancias, por la capacitancia entre conductores así como sus parámetros distribuidos.

La comunicación serial consiste en transmitir los bits de datos uno por uno, y no en bloques de 8 como en la comunicación paralela. La transmisión de datos en serie es una de las más comunes para aquellas aplicaciones en las que la velocidad no es demasiado importante, o no es posible conseguirla. Además se utilizan para enviar datos a través de largas distancias, ya que las comunicaciones en paralelo exigen demasiado cableado para ser operativas.

El puerto serial de las computadoras es conocido como puerto RS-232, la ventaja de este puerto es que todas las computadoras traen al menos un puerto serial, este permite las comunicaciones entre otros dispositivos tales como otra computadora, el mouse, impresora y para nuestro caso con los microcontroladores.

8.2.- TIPOS DE COMUNICACIONES SERIALES

Existen dos tipos de comunicaciones seriales: la síncrona y asíncrona. En la comunicación serial síncrona, además de una línea sobre la cual se transmitirán los datos se necesita de una línea la cual contendrá los pulsos de reloj que indicaran cuando un dato es valido.

Ejemplos de este tipo de comunicación son:

- **I2C**
- **ONE WIRE**
- **SPI**

En la comunicación serial asíncrona, no son necesarios los pulsos de reloj, La duración de cada bit esta determinada por la velocidad con la

cual se realiza la transferencia de datos. La siguiente figura muestra la estructura de un carácter que se trasmite en forma serial asíncrona..

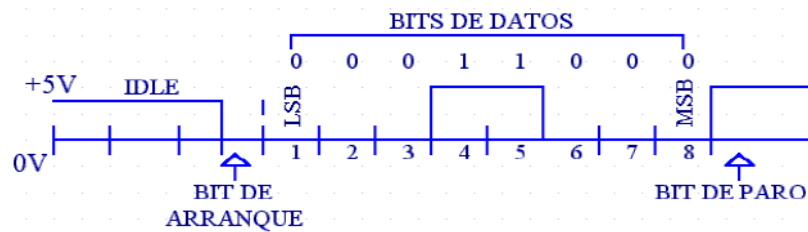


Figura No.1 “Transmisión de un carácter”

Normalmente cuando no se realiza ninguna transferencia de datos, la línea del transmisor se encuentra en estado de (idle) esto quiere decir en estado alto.

Para iniciar la transmisión de datos, el transmisor coloca esta línea en bajo durante determinado tiempo, lo cual se le conoce como bit de arranque (start bit) y a continuación empieza a transmitir con un intervalo de tiempo los bits correspondientes al dato, empezando siempre por el BIT menos significativo (LSB), y terminando con el BIT mas significativo.

Si el receptor no esta sincronizado con el transmisor, este desconoce cuando se van a recibir los datos. Por lo tanto el transmisor y el receptor deberán tener los mismos parámetros de velocidad, paridad, número de bits del dato transmitido y de BIT de parada.

En los circuitos digitales, cuyas distancias son relativamente cortas, se pueden manejar transmisiones en niveles lógicos TTL (0-5V), pero cuando las distancias aumentan, estas señales tienden a distorsionarse debido al efecto capacitivo de los conductores y su resistencia eléctrica. El efecto se incrementa a medida que se incrementa la velocidad de la transmisión. Todo esto origina que los datos recibidos nos sean igual a los datos transmitidos, por lo que no se puede permitir la transferencia de datos.

Una de las soluciones mas l3gica es aumentar los m3rgenes de voltaje con que se transmiten los datos, de tal manera que las perturbaciones a causa de la l3nea se pueda corregir.

8.3.- NORMA RS-232

Ante la gran variedad de equipos, sistemas y protocolos que existen surgi3 la necesidad de un acuerdo que permitiera a los equipos de varios fabricantes comunicarse entre si. La **EIA (Electronics Industry Association)** elaboro la norma RS-232, la cual define la interfase mec3nica, los pines, las se3ales y los protocolos que debe cumplir la comunicaci3n serial. Todas las normas RS-232 cumplen con los siguientes niveles de voltaje:

- Un "1" l3gico es un voltaje comprendido entre -5v y -15v en el transmisor y entre -3v y -25v en el receptor.
- Un "0" l3gico es un voltaje comprendido entre +5v y +15 v en el trasmisor y entre +3v y +25 v en el receptor.

El env3o de niveles l3gicos (bits) a trav3s de cables o l3neas de transmisi3n necesita la conversi3n a voltajes apropiados. En los microcontroladores para representar un 0 l3gico se trabaja con voltajes inferiores a 0.8v, y para un 1 l3gico con voltajes mayores a 2.0 V. En general cuando se trabaja con familias TTL y CMOS se asume que un "0" l3gico es igual a cero Volts y un "1" l3gico es igual a cinco Volts.

La importancia de conocer esta norma, radica en los niveles de voltaje que maneja el puerto serial del ordenador, ya que son diferentes a los que utilizan los microcontroladores y los dem3s circuitos integrados. Por lo tanto se necesita de una interfase que haga posible la conversi3n de los niveles de voltaje a los est3ndares manejados por los CI TTL.

8.4.- CONECTOR DE LA COMUNICACI3N SERIAL

La norma especifica dos tipos de conectores: DB-25 de 25 pines, que es similar al del puerto paralelo, y el DB-9 de 9 pines, que es m3s barato y m3s utilizado. En cualquier caso, no se suele emplear m3s de 9 pines en el conector DB-25. Ambos conectores con machos en la parte trasera de la computadora, as3 que para conectar alg3n dispositivo, se necesitar3 un conector DB-25 o DB-9 hembra. A continuaci3n se

muestra una figura con la distribución de los pines en el DB-9. El DB-25 es similar al del puerto paralelo

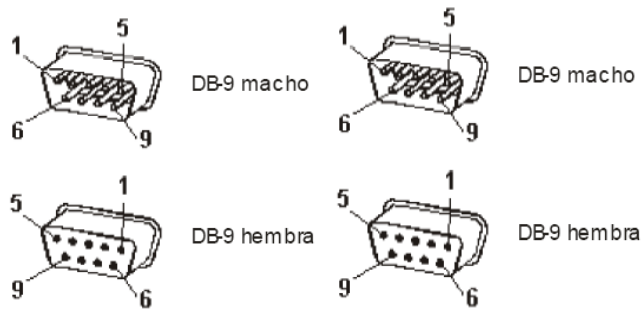


Figura No. 2 “Conector serial: macho y hembra”

La asignación de cada pin, tanto en el DB-25 como en el DB-9 se muestra en la siguiente tabla:

NÚMERO DE PIN		SEÑAL	DESCRIPCIÓN	E/S
DB-25	DB-9			
1	1	-	Chasis (Protección eléctrica)	-
2	3	TxD	Transmit Data	S
3	2	RxD	Receive Data	E
4	7	RTS	Request To Send	S
5	8	CTS	Clear To Send	E
6	6	DSR	Data Set Ready	E
7	5	SG	Signal Ground	-
8	1	CD/DCD	(Data) Carrier Detect	E
20	4	DTR	Data Terminal Ready	S
22	9	RI	Ring Indicator	E

Tabla 1. “Descripción de pines del conector serial DB-25 y DB9”

El término de entrada y salida es con respecto a al DTE (la computadora).

La función de cada pin se muestra en la siguiente tabla:

SEÑAL	DESCRIPCIÓN	FUNCIÓN
TxD	Transmit Data	Salida de datos seriales
RxD	Receive Data	Recepción de datos seriales
CTS	Clear To Send	Indica que el MODEM esta listo para intercambiar datos
DCD	Data Carrier Detect	Detección de portadora
DSR	Data Set Ready	Indica al DTE que el MODEM esta listo para comunicarse
DTR	Data Terminal Ready	Indica al MODEM que el DTE esta listo para comunicarse
RTS	Request To Send	Indica al MODEM que el DTE esta listo para intercambiar datos
RI	Ring Indicador	Detección del sonido de llamada

Tabla No. 2 y 3 “Función de pines del conector serial DB-25 y DB9”

Tanto el dispositivo a conectar como la computadora (o el programa terminal) tienen que usar el mismo protocolo serie para comunicarse entre si. Puesto que el estándar RS-232 no permite indicar en que modo se esta trabajando, es el usuario quien tiene que decidirlo y configurar ambas partes. Los parámetros que hay que configurar son:

- Número de bits de datos a transmitir
- Bit de paridad
- Numero de bits de stop
- Velocidad de transmisión

Los pines que portan los datos son RXD y TXD. Las demás se encargan de otros trabajos: DTR indica que la computadora está encendida, DSR que el aparato conectado a dicho puerto esta encendido, RTS que la computadora puede recibir datos (porque no esta ocupado), CTS que el aparato conectado puede recibir datos, y DCD detecta que existe una comunicación, presencia de datos.

8.5.- PUERTO SERIAL EN NULL

El modo NULL del puerto serial es utilizado para conectar dos DTEs juntos. Esta es una forma muy común, fácil y barata de transferir datos entre dos computadoras o entre una computadora y un microcontrolador con interfaz de comunicación serial.

En el siguiente diagrama se muestra el cableado de pines en modo NULL:

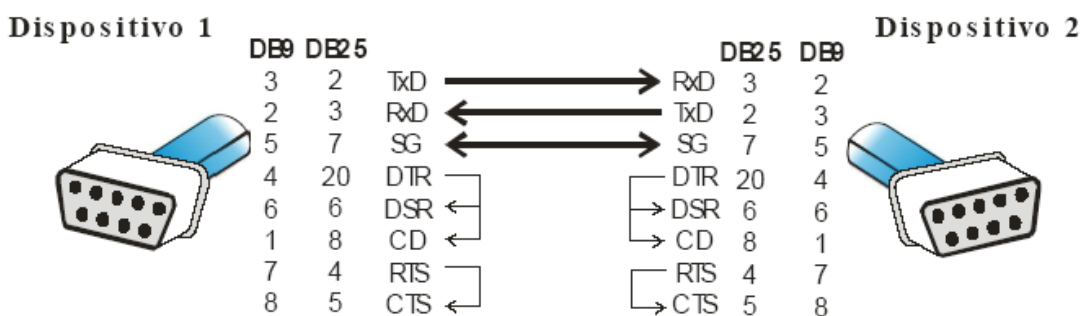


Figura No. 3 “Cableado de pines en modo Null”

Se puede observar que solo se necesitan 3 líneas para interconectarse: TxD, RxD y SG. El resto de los pines se encuentran interconectados entre sí como se observa en el diagrama. La teoría de funcionamiento de esta conexión es muy fácil de entender: los datos transmitidos por el dispositivo 1, deben ser recibidos por el dispositivo 2, entonces conectamos TxD del dispositivo 1 con RxD del dispositivo 2. Los datos transmitidos por el dispositivos 2 deben ser recibidos por el dispositivo 1, entonces conectamos TxD del dispositivo 2 con RxD del dispositivo 1. La 12 terminal de tierra debe ser común para los dos.

Utilizando el modo NULL se puede realizar un arreglo llamado “conexión de retroalimentación”, el cual consiste en conectar el pin transmisor del puerto con el pin receptor del mismo puerto, tal como se observa a continuación:

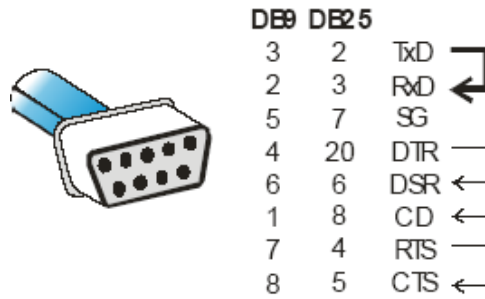


Figura No. 4 “Conexión de retroalimentación”

8.6.- CONEXIÓN DE UN MICROCONTROLADOR Y LA PC

Para conectar la PC a un microcontrolador por el puerto serie en modo Null, se utilizan las señales Tx, Rx y GND. La PC utiliza la norma RS232, por lo que los niveles de tensión de los pines están comprendidos entre +15 y -15 voltios. Los microcontroladores normalmente trabajan con niveles TTL (0-5v). Es necesario por tanto intercalar un circuito que adapte los niveles:

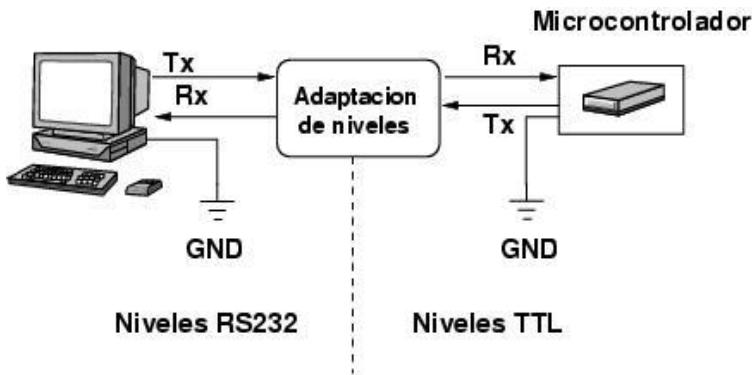


Figura No. 5 “Acomodamiento de voltajes entre microcontrolador y la PC”

8.7.- EL CIRCUITO MAX-232

Este circuito soluciona los problemas de niveles de voltaje cuando se requiere enviar señales digitales sobre una línea RS-232. Este chip se utiliza en aquellas aplicaciones donde no se dispone de fuentes dobles de

+12 y -12 Volts. El MAX-232 necesita solamente una fuente de +5V para su operación, internamente tiene un elevador de voltaje que convierte el voltaje de +5V al de doble polaridad de +12V y -12V. Cabe mencionar que existen una gran variedad de CI que cumplen con la norma RS-232 como lo son: MAX220, DS14C232, MAX233, LT1180A.

9.- EL BUS SERIE UNIVERSAL (USB).

El USB es un estándar (1995) que define un bus utilizado para conectar periféricos al ordenador. La principal característica que tiene es que la conexión es muy sencilla, ya que utiliza un único conector para conectar a través de un bus serie todos los dispositivos. En él se definen los conectores y los cables, una topología especial tipo estrella para conectar hasta 127 dispositivos y protocolos que permiten la detección y configuración automática de los dispositivos conectados. USB 1.0 soporta dos tasas de transferencia diferentes, una baja de 1,5 Mbps para la conexión de dispositivos lentos de bajo coste (joysticks, ratones) y otra alta de hasta 12 Mbps para la conexión de dispositivos que requieren un mayor ancho de banda (discos o CDROMS).

La especificación de este estándar ha sido respaldada por las empresas líderes mundiales en el campo de la informática: Intel, IBM, DEC, Microsoft, Compac, NEC

y Northern Telecom, empresas que garantizan su continuidad y utilización.

A mediados del año 2000 aparece la versión 2.0, que fue creada por el conjunto de compañías arriba mencionadas, a las cuales se unieron Hewlett Packard, Lucent y Philips. USB 2.0 multiplica la velocidad del bus por un factor de 30 o 40, llegando a

alcanzar una velocidad de 480 Mbps, con una diferencia de coste casi inapreciable. Es compatible con la versión anterior y utiliza los mismos cables y conectores, únicamente se necesitan nuevos hubs que soporten la versión 2.0. Estos hubs son algo más complejos que los anteriores, ya que tienen que manejar el tráfico de datos de tres velocidades distintas sin ser excluyentes entre ellas.

Cabe también destacar que USB 2.0 nunca llegará a reemplazar completamente a USB 1.0, ya que existen algunos tipos de dispositivos, como los HID (teclados,

ratones,...), que no requieren las altas velocidades que alcanza esta nueva versión y que únicamente encarecerían el dispositivo.

Anteriormente los periféricos se conectaban mapeados directamente en direcciones de E/S, se les asignaba una dirección específica y en algunos casos un canal DMA. Esta situación conducía a tener conflictos en la asignación de estos recursos, puesto que siempre han estado bastante limitados en el ordenador. Además cada dispositivo tenía su propio puerto de conexión y utilizaba sus cables específicos,

lo que daba lugar a un incremento de los costes. Debido a que a cada dispositivo se le tenían que asignar unos recursos específicos la detección del mismo debía hacerse a la hora de arrancar el sistema y nunca se podía incorporar un nuevo dispositivo cuando el sistema estaba en marcha.

Los dos aspectos fundamentales que motivaron la realización de este estándar fueron la necesidad de configurar de forma sencilla los periféricos conectados al ordenador y la necesidad de aumentar el número de puertos disponibles.

Este estándar define una topología de conexión en estrella, tal como se muestra en la figura 2.1, por medio de la incorporación de varios concentradores.

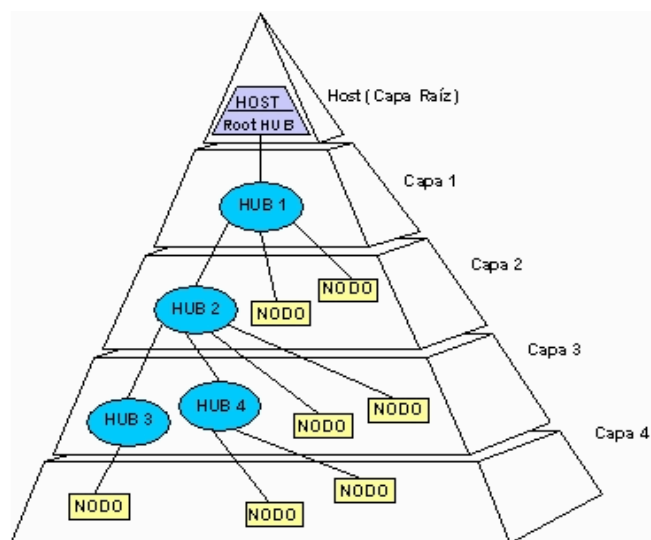


Figura 6: Topología de una conexión USB

Cada concentrador se conecta por un lado al ordenador, que contiene una o dos interfaces de este tipo en la placa base, o a otro concentrador y, por otro lado, se conecta a varios dispositivos o incluso a otro concentrador. De este modo pueden existir periféricos que vengan ya preparados con nuevos conectores USB para incorporar nuevos dispositivos, hasta un total de 127, todos ellos funcionando simultáneamente. Los hubs tienen la misión de ampliar el número de dispositivos que se pueden conectar al bus. Son concentradores cableados que permiten la conexión simultánea de múltiples dispositivos y lo más importante es que se pueden concatenar entre sí ampliando la cantidad de puertos disponibles para los periféricos. El concentrador detecta cuándo un periférico es conectado o desconectado a/de uno de sus puertos, notificándolo de inmediato al controlador de USB. También realiza funciones de acoplamiento de las velocidades de los dispositivos más lentos.

Existe una gran variedad de dispositivos USB que se conectan todos al mismo bus. La característica más importante es que todos ellos utilizan el mismo tipo de cable y de conector y se conectan de la misma forma tan sencilla. El host decide qué dispositivo puede acceder al bus, utilizando un protocolo parecido al de paso de testigo. Este protocolo se caracteriza porque entre los diferentes dispositivos se va pasando un identificador a lo largo del tiempo que permite la utilización del bus.

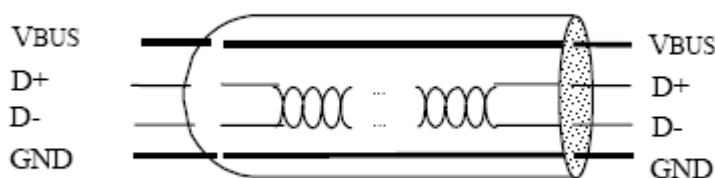


Figura 7: Formato del tipo de cable utilizado

El host USB tiene las funciones de:

- Detectar la conexión/desconexión de dispositivos y configurarlos.
- Controlar las transferencias de datos y de control que tienen lugar en el bus.

- Realización de auditorías sobre la actividad del sistema.
- Servir como fuente de alimentación a los dispositivos.

El USB define dos líneas para transmitir datos y otras dos para transmitir potencia. Los datos se transmiten de forma balanceada a velocidades entre 1,5 Mbps y 12 Mbps. La señal se transmite codificada en un código auto reloj de no retorno a cero invertido (NRZI) para poder incluir junto con los datos información de sincronización. Las líneas de alimentación (Vbus y GND) evitan la necesidad de utilizar fuentes de alimentación externas. Tiene una tensión de 5 V y la corriente se limita a un máximo de 3 a 5 amperios por razones de seguridad, siendo el consumo y la configuración eléctrica totalmente transparente al usuario. La distancia entre dos periféricos conectados al mismo cable no debe ser superior a 5 metros para evitar problemas de caídas de tensión.

El computador identifica automáticamente el dispositivo que se conecta mientras opera y lo configura sin tener que instalar drivers específicos del fabricante. Al comienzo se detectan los dispositivos conectados midiendo los niveles de voltaje de las líneas. Si un dispositivo está conectado, entonces el dispositivo envía información sobre el tipo o la clase a la que pertenece, qué modo de transferencia utilizará y cuáles son sus necesidades de ancho de banda. El host reconocerá el dispositivo buscando en la lista de drivers del sistema operativo y teniendo en cuenta los demás dispositivos conectados le asignará un ancho de banda determinado. De la misma forma también se pueden desconectar los dispositivos del sistema. El controlador USB del host asigna un número diferente de dispositivo a cada uno de los periféricos que se conectan a este bus. Para empezar la transferencia, éste envía un paquete que identifica al dispositivo objeto de la transferencia. El protocolo soporta cuatro tipos de transferencias:

- **Control.** Son transferencias que se utilizan para leer información de los descriptores en los registros de los dispositivos (llamados *endpoints*), interpretarla y poder configurarlos.
- **Interrupción.** Usadas en los periféricos del tipo de los controladores de juegos, teclados y ratones, cuya comunicación es unidireccional y poco frecuente.
- **Masiva.** Son transferencias no periódicas que precisan de todo el ancho de banda disponible. Utilizadas por las impresoras y los scanners.

- **Isócrona.** Dedicadas a las transferencias de telecomunicaciones, como voz o vídeo, que garantiza unas tasas de transferencia constantes. Se caracterizan porque el número de pulsos de reloj que transcurren entre la transmisión de dos caracteres es constante, por lo tanto, se está enviando información constantemente entre el host y el dispositivo.

10.- MICROCONTROLADORES

Un controlador es un dispositivo que se emplea en el gobierno de uno o varios procesos. Aunque el concepto de controlador ha continuado inalterable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta; posteriormente se utilizaron los microprocesadores, que se rodeaban con chips de memoria y E/S sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip que recibe el nombre de microcontrolador.

En definitiva, un microcontrolador es un circuito integrado programable que contiene todos los componentes de un computador. Se utiliza para controlar el funcionamiento de una tarea determinada y, debido a su reducida medida, suele ir incorporado en el propio dispositivo que gobierna. Esta última característica es la que le confiere la denominación de controlador incrustado (*embedded controller*).

El microcontrolador es un computador dedicado. En su memoria solamente reside un programa destinado a gobernar una aplicación determinada; sus líneas de entrada/salida soportan la conexión de los sensores y actuadores del dispositivo a controlar y todos los recursos complementarios disponibles tienen como única finalidad atender sus requerimientos. Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea asignada.

Un microcontrolador posee todos los componentes de un computador pero con unas características fijas que no pueden alterarse. Todos disponen de los bloques esenciales: procesador, memoria de datos y de instrucciones, módulos de E/S, oscilador de reloj y módulos controladores de periféricos. Además de estos elementos,

existen una serie de recursos especiales que los fabricantes pueden ofertar, algunos amplían las capacidades de las memorias, otros incorporan nuevo recursos y hay quienes reducen las prestaciones al mínimo para aplicaciones muy simples. Depende del programador el encontrar el modelo mínimo que se ajuste a sus requerimientos y así minimizar el coste, el hardware y el software. Algunos de los principales recursos específicos que incorporan los microcontroladores son:

Temporizadores (*Timers*).

Perro guardián (*Watchdog*).

Protección frente a fallo de alimentación (*Brown-out*).

Estado de bajo consumo.

Conversores AD y DA.

Modulador de anchura de pulsos PWM.

Comparadores analógicos.

Puertos de E/S digital.

Puertos de comunicación: serie, CAN, USB, I2C,...

10.1.- FAMILIA DE MICROCONTROLADORES18F

Características fundamentales:

- Arquitectura RISC avanzada Harvard: 16- bit con 8- bit de datos.
- 77 instrucciones
- Desde 18 a 80 pines
- Hasta 64K bytes de programa (hasta 2 Mbytes en ROMless)
- Multiplicador Hardware 8x8
- Hasta 3968 bytes de RAM y 1KBytes de EEPROM
- Frecuencia máxima de reloj 40Mhz. Hasta 10 MIPS.
- Pila de 32 niveles.
- Múltiples fuentes de interrupción
- Periféricos de comunicación avanzados (CAN y USB)

10.2.- PINES DEL 18F4550

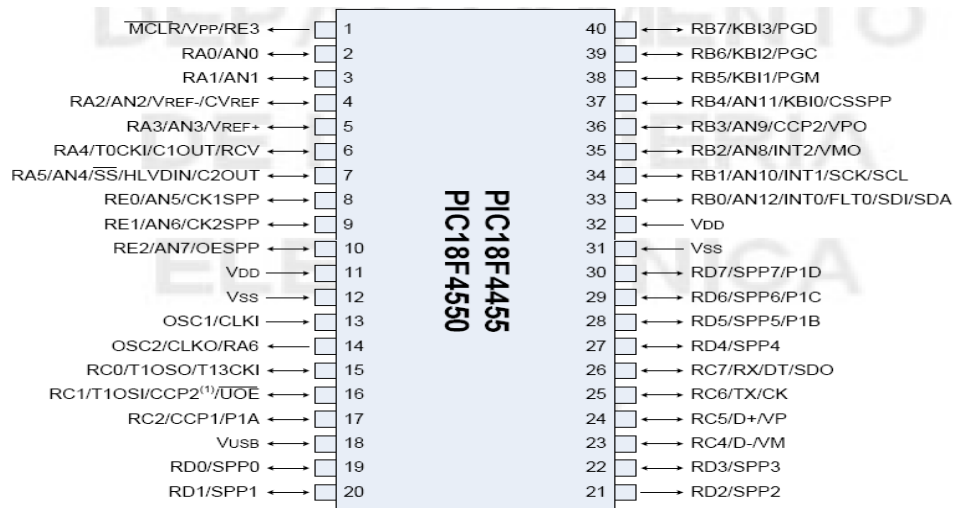


Figura No.8 “Pines del microcontrolador 18F4550”

10.3.- ORGANIZACIÓN DE MEMORIA

El uC PIC18F4550 dispone de las siguientes memorias:

- Memoria de programa: memoria flash interna de 32.768 bytes
 - Almacena instrucciones y constantes/datos
 - Puede ser escrita/leída mediante un programador externo o durante la ejecución programa mediante unos punteros.
- Memoria RAM de datos: memoria SRAM interna de 2048 bytes en la que están incluidos los registros de función especial.
 - Almacena datos de forma temporal durante la ejecución del programa
 - Puede ser escrita/leída en tiempo de ejecución mediante diversas instrucciones
- Memoria EEPROM de datos: memoria no volátil de 256 bytes.
 - Almacena datos que se deben conservar aun en ausencia de tensión de alimentación

- Puede ser escrita/leída en tiempo de ejecución a través de registros
- Pila: bloque de 31 palabras de 21 bits
 - Almacena la dirección de la instrucción que debe ser ejecutada después de una interrupción o subrutina
- Memoria de configuración: memoria en la que se incluyen los bits de configuración (12 bytes de memoria flash) y los registros de identificación (2 bytes de memoria de solo lectura).

10.4.- ARQUITECTURA HARDVARD

Los microcontroladores pueden clasificarse según su arquitectura, que puede ser Von Neumann o Harvard. La **arquitectura Von Neumann** se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único.

Hay aspectos positivos en esta configuración como los accesos a tablas almacenadas en memoria ROM y un set de instrucciones más ortogonal. El bus de direcciones es usado para identificar qué posición de memoria esta siendo accedida, mientras que el bus de datos es utilizado para trasladar información entre la CPU y alguna dirección de memoria o viceversa. Con un único sistema de buses, la arquitectura Von Neumann es usada secuencialmente para acceder a instrucciones de la memoria de programa y ejecutarlas regresando desde/hacia la memoria de datos. Esto significa que el ciclo de instrucción no puede solaparse con ningún acceso a la memoria de datos.

Una desventaja de esta arquitectura podría ser que el contador de programa o algún otro registro se corrompieran y apuntaran a la memoria de datos y se tomara ésta momentáneamente como memoria de programa. Consecuentemente se ejecutaría una instrucción no deseada o un error en la decodificación de la instrucción.

La **Arquitectura Harvard** se caracteriza por disponer de dos memorias independientes, una que contiene sólo instrucciones y otra con sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias. Una de las ventajas de esta

arquitectura es que la operación del microcontrolador puede ser controlada más fácilmente si se presentara una anomalía en el contador de programa.

Existe otra arquitectura que permite accesos a tablas de datos desde la memoria de programa. Esta arquitectura es la llamada arquitectura Harvard modificada.

Esta última arquitectura es la dominante en los microcontroladores actuales ya que la memoria de programa es usualmente ROM, OTP, EPROM o FLASH mientras que la memoria de datos es usualmente RAM. Consecuentemente, las tablas de datos pueden estar en la memoria de programa sin que sean perdidas cada vez que el sistema es apagado. Otra ventaja importante en la arquitectura Harvard modificada es que las transferencias de datos pueden ser solapadas con los ciclos de decodificación de instrucciones. Esto quiere decir que la siguiente instrucción puede ser cargada de la memoria de programa mientras se está ejecutando una instrucción que accede a la memoria de datos. La desventaja de la arquitectura Harvard modificada podría ser que se requieren instrucciones especiales para acceder a valores en memoria RAM y ROM haciendo la programación un poco complicada.

Las principales ventajas que se pueden encontrar en el uso de microcontroladores son:

Gestión eficiente de procesos.

Aumento de la fiabilidad.

Reducción del tamaño, consumo y coste.

Mayor flexibilidad (únicamente se requiere la reprogramación).

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16 o 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 4 y 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado y los de 4 bits se resisten a desaparecer. La razón de esta tendencia es que los microcontroladores de 4 y 8 bits son apropiados para la gran mayoría de las aplicaciones, lo que hace innecesario emplear microcontroladores más potentes y consecuentemente más caros.

En cuanto a las técnicas de fabricación, cabe decir que prácticamente la totalidad de los microcontroladores actuales se fabrican con tecnología CMOS 4 (*Complementary Metal Oxide Semiconductor*). Esta tecnología supera a las técnicas

anteriores por su bajo consumo y alta inmunidad al ruido.

- El uC PIC18F4550 dispone buses diferentes para el acceso a memoria de programa y memoria de datos (arquitectura Harvard):
- Bus de la memoria de programa:
 - 21 líneas de dirección
 - 16/8 líneas de datos (16 líneas para instrucciones/8 líneas para datos)
- Bus de la memoria de datos:
 - 12 líneas de dirección
 - 8 líneas de datos

10.5.-MEMORIA RAM DE DATOS

- El uC PIC18F4550 dispone una memoria RAM de datos 2.048 bytes (8 bancos de 256 bytes). Además dispone de 160 bytes dedicados a los registros de función especial (SFR's) situados en la parte alta del banco 15.
- Para acceder a un byte de la memoria RAM de datos primero debe seleccionarse el banco al que pertenece el byte mediante el registro de selección de banco (BSR) y a continuación direccionar el byte dentro del banco. Además existe una modalidad de acceso rápido a las 96 posiciones de la parte baja del banco 0 y a los 160 bytes de SFR's (banco de acceso rápido).

10.6.-SISTEMAS DE ADQUISICIÓN DE DATOS MEDIANTE SENSORES.

Adquirir datos es el proceso de medir los parámetros físicos que los sensores transforman en señales eléctricas e introducirlos en el sistema. Mientras se adquiere, se producen procesos que involucran esos datos adquiridos: alarmas, escalado de datos, a veces control, guardar los datos, etc. Posteriormente a la adquisición de datos se realizan análisis para extraer información útil a partir de la cual se reportan los datos, se

visualizan resultados y se comparte la información. Estas fases representan la funcionalidad de los sistemas modernos de adquisición de datos basada en PC.

La función de adquisición es una de las componentes más críticas. En un sistema basado en PC la adquisición se realiza mediante un hardware especializado de medición que puede desglosarse en sensores, conectividad de la señal o de los sensores, acondicionamiento de la señal, y conversión analógica-digital.

Una amplia variedad de sensores se usan para convertir magnitudes físicas en señales eléctricas. Los hay de todos los tipos: de temperatura como termopares y termistores, transductores de presión, galgas extensiométricas y acelerómetros,...

Después de instalar los sensores, hay que conectarlos al sistema de adquisición.

La conectividad de la señal describe los componentes del hardware con los cuales se conectan los sensores al equipo. Hay muchos tipos de conectores y es un factor a decidir a la hora de configurar el equipo.

El acondicionamiento de la señal es uno de los componentes más importantes de un equipo de adquisición basado en PC. Muchas señales requieren algún tipo de preparación antes de ser digitalizadas. Por ejemplo, los termopares producen señales de muy bajo nivel que requieren amplificación, filtrado y linealización. Otros sensores, como termistores, galgas y acelerómetros, requieren alimentación además de amplificación y filtrado, mientras que otras señales pueden requerir aislamiento para proteger el sistema de alto voltaje. Idealmente un equipo debería permitir todo tipo de acondicionamiento incluido la combinación de algunos procesos, si bien éste ha de adaptarse a las posibles necesidades del usuario.

Después de que los parámetros físicos se han convertido en señales eléctricas y se acondicionan correctamente, las señales eléctricas analógicas se convierten en valores digitales y se pasan estos valores al computador. La conversión analógica-digital se realiza a través de una tarjeta de adquisición de datos o en un sistema integrado con Acondicionamiento y conectividad.

La combinación de sensores, conectividad de la señal, acondicionamiento y conversión analógica-digital constituye el hardware de medición de un equipo de adquisición basado en PC. Este hardware se configura y se controla a través de software construyendo aplicaciones a la medida de la aplicación deseada.

11.- REALIZACION DEL HARDWARE

El hardware necesario para la realización del proyecto consiste por una parte, en la plataforma hardware sobre la que utilizar el software de desarrollo, por otra, en los elementos necesarios para el montaje de los componentes y testeo de la placa. En cuanto al material para el montaje de los componentes sobre la placa se utilizará un soldador de montaje superficial y para el testeo de la misma será necesario utilizar un multímetro.

12.-USB vs. FireWire

Una comparación entre ambos tipos de enlace.

Tanto FireWire como USB son tecnologías que persiguen un nuevo método de conectar múltiples periféricos a un ordenador, permitiendo que estos sean añadidos o desconectados sin la necesidad de reiniciar. Ambos usan cables ligeros y flexibles con conectores duraderos, pero aquí terminan los parecidos. Aunque los cables de FireWire y USB pueden parecer a la vista los mismo, la cantidad de datos que por ellos transcurre es bastante diferente. Como muestra la siguiente tabla, la velocidad y la capacidad de transferencia marcan la principal distinción entre estas dos tecnologías:

	FireWire	USB
Número máximo de dispositivos	63	127
Cambio en caliente (agregar o quitar dispositivos sin tener que reiniciar el ordenador).	Sí	Sí
Velocidad de transferencia		
De Datos.	200 Mbps	1,5 Mbps
	400 Mbps	12 Mbps
	800 Mbps	480 Mbps

Como puede verse, FireWire ofrece velocidades de transferencia de datos superiores a las ofrecidas por USB pero esto no significa que FireWire gane la "guerra" de interfaces. Los conectores FireWire y USB pueden coexistir pacíficamente. USB 1.0 queda reservado para periféricos con un pequeño ancho de banda (ratones, teclados, módems, etc.), mientras que FireWire, gracias a su mayor ancho de banda, es más adecuado para aplicaciones de vídeo digital (DV), audio profesional, cámaras fotográficas digitales de alto nivel y aparatos de ocio domésticos.

Según el datasheet del microcontrolador [PIC] los valores de los condensadores que acompañan al cristal deben ser de 27 pF. Estos valores son orientativos. Una capacidad mayor produce mayor estabilidad del cristal pero también un tiempo de arranque mayor. La resistencia R_s puede ser necesaria para no sobre alimentar al cristal.

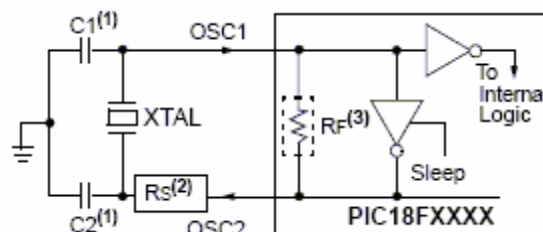


Figura 9: Oscilador cristal o cerámico (configuración XT, HS o HSPLL)

La detección de un periférico USB por parte de un ordenador se realiza conectando en el dispositivo, la línea D+ o D- a 3,3 V a través de una resistencia de pull-up. El hub tiene una resistencia de pull-down de 15 K Ω en cada una de las dos líneas de datos del puerto USB, D+ y D-. El dispositivo tiene una resistencia de pullup de 1,5 K Ω en la línea D+ (para velocidad completa) o D- (para baja velocidad) conectada a una tensión de 3,3 V. Cuando el periférico se conecta al puerto, la línea que tiene la resistencia de pull-up conectada se pone a nivel alto y cuando el hub detecta este nivel alto en una de las líneas, asume que un dispositivo está conectado y determina su velocidad en función de la línea que sea.

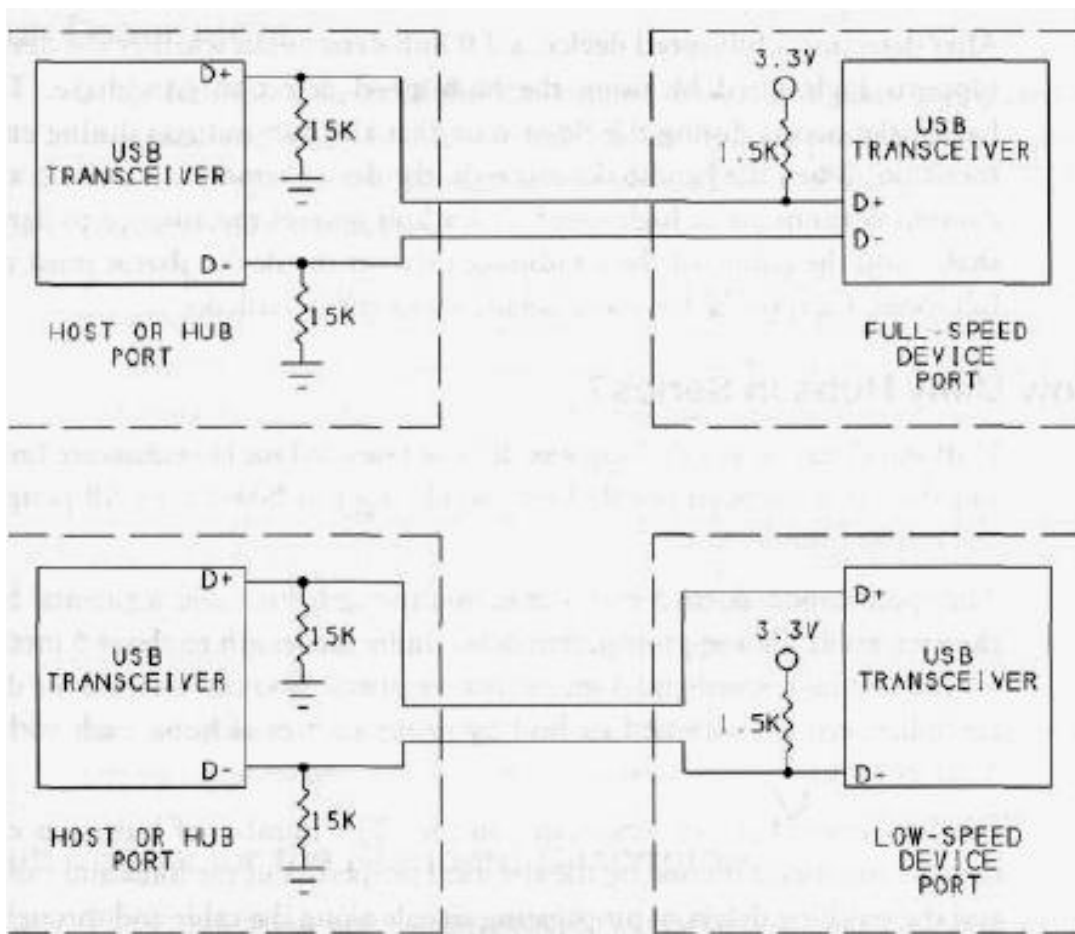


Figura 10: detección de un periférico USB

13.- QUE ES UNA ESTACION AGROCLIMATICA.

Una Estación Agroclimática es un conjunto de dispositivos que pueden realizar diversas medidas meteorológicas, como Temperatura, Humedad del aire, Dirección y Velocidad del viento, etc.

Se utilizan estas estaciones para diversos fines. En el sector agrario, por ejemplo, se utilizan para conocer el clima de la zona y más en concreto las necesidades de agua de los cultivos.

Para la instalación de una estación agroclimática hay que elegir correctamente la ubicación, los instrumentos, así como la adecuada disposición de los mismos (protección, orientación, altura). Además, hay que prever la existencia de personal debidamente formado para el mantenimiento, lectura y envío de la información al centro zonal correspondiente.

Actualmente existen numerosas redes de estaciones por todo el territorio español, promovidas por las comunidades autónomas, organismos públicos o por iniciativa privada, para el control de diferentes fenómenos, como alerta de plagas, de información agroclimática, prevención de incendios, etc.

14.- TIPOS DE ESTACIONES

Podemos definir globalmente 2 tipos de estaciones agroclimáticas:

- Estaciones Manuales
- Estaciones Automáticas

Las Estaciones Manuales son aquellas que están formadas por 'instrumentos' y no 'sensores' como las Automáticas. La diferencia estriba en que los 'instrumentos' son dispositivos analógicos y los 'sensores' son dispositivos electrónicos o digitales que traducen una señal analógica en una señal digital fácilmente procesable.

Según el Manual de Sistema Mundial de Observación, las estaciones automáticas son aquellas en las que 'los instrumentos efectúan y transmiten o registran automáticamente las observaciones, realizando, en caso necesario, directamente la conversión a la clave correspondiente o bien realizándose esa conversión en una estación transcriptoras'.

En estas estaciones como ya dijimos anteriormente, se sustituyen los instrumentos por sensores.

Podemos decir que las estaciones automáticas surgieron de la necesidad de obtener información en ubicaciones con dificultad de acceso, en lugares inhóspitos o, en las estaciones dotadas de personal, para efectuar observaciones fuera del horario normal de trabajo de los observadores.

15.- ESTADO DEL ARTE

15.1.- VARIABLES CLIMÁTICAS MEDIDAS

- Las estaciones de la Red están provistas de sensores para el registro de la temperatura del aire a 2 metros sobre el suelo, precipitación, humedad relativa, presión barométrica, humedad del follaje, punto de rocío, radiación solar, dirección y velocidad del viento.



Figura 11: estación agroclimática

15.2.- APLICACIONES

Las estaciones agroclimáticas tienen diferentes funciones de acuerdo al lugar y por supuesto al clima que se da en este por lo que la Red de estaciones brinda información para apoyar los procesos de producción agropecuaria y así poder saber en que podemos usar las estaciones agroclimáticas, por mencionar algunas:

- Detección temprana de heladas
- Aplicación del riego agrícola
- Control de plagas y enfermedades
- Reducción en el uso de agroquímicos
- Caracterización regional del clima
- Predicción de la cosecha en función de la tendencia climática
- Identificación de áreas con potencial climático para el establecimiento de cultivos alternativos
- Monitoreo meteorológico en tiempo real
- Generación de registros climáticos en forma sistematizada
- Acceso a los datos en forma continua a través de Internet.

15.3.- CRECIMIENTO DE LA RED

Existen diferentes países que poco a poco han venido implementando el uso de las estaciones agroclimáticas por lo cual es necesario saber donde podemos encontrar algunas de estas estaciones; por ejemplo en México La red está constituida por 762 estaciones distribuidas en 29 estados (Aguascalientes, Baja California Sur, Campeche, Chiapas, Chihuahua, Coahuila, Colima, Durango, Estado de México, Guanajuato, Guerrero, Hidalgo, Jalisco, Michoacán, Morelos, Nayarit, Nuevo León, Oaxaca, Puebla, Quintana Roo, San Luis Potosí, Sonora, Sinaloa, Tabasco, Tamaulipas, Tlaxcala, Veracruz, Yucatán y Zacatecas) de la República Mexicana.

La red crecerá hacia las áreas cañeras del país, donde se establecerán 86 estaciones en las 13 regiones cañeras del país.

Son usuarios de este servicio, las oficinas de meteorología de cada país, los centros de investigación relacionados con el estudio del clima, los medios de información pública (prensa, radio, TV), entre otros.

Los boletines meteorológicos proporcionados por cada una de las estaciones fijas pertenecientes a este servicio, se realizan en base a la investigación y la observación de los fenómenos climáticos que se originan en la región en donde se hallan situadas. Dicha información es luego transmitida respetando los códigos y metodologías determinadas por la WMO, a fin de asegurar la rápida distribución de los mensajes en favor de los usuarios del servicio. La mayoría de las estaciones están activas durante las 24 horas y sus reportes se actualizan cada hora.

Por lo general las transmisiones en HF se efectúan recurriendo al uso del FAX y el RTTY, lo cual hace posible la recepción de mapas, fotografías, cuadros y demás datos de interés que brindan un completo panorama de la situación climática al momento de recibir cada informe. Los códigos habitualmente utilizados para transmitir los boletines son el PILOT, el SHIP, el SINOP y el TEMP.

A continuación citamos aquí a las estaciones más importantes:

- Auckland METEO (NZL):
- Beijing METEO (CHN):
- Bracknell METEO (G):
- Buenos Aires METEO (ARG):
- Delhi METEO (IND):
- Hamburgo METEO (D):
- Madrid METEO (E):
- Melbourne METEO (AUS):
- Moscú METEO (RUS):
- Roma METEO (I):
- Tokio METEO (J):

Asimismo, existen algunas pocas estaciones que emplean también la telegrafía (CW) y la voz (USB) para la distribución de los partes meteorológicos. Entre ellas podemos citar las siguientes:

- Bangkok METEO (THA):
- Taipei METEO (TWN):
- Tokio METEO (J):

inglés Global Telecommunications Systems. Este posibilita la recepción de todo tipo de información meteorológica suministrada en determinados «formatos especiales» para facilitar su aprovechamiento por parte de

terceros, a quienes esa información les permite estar al tanto de las condiciones climatológicas existentes en cualquier parte del mundo.

16.- SENSORES QUE SE USAN EN LAS ESTACIONES AGROCLIMATICAS Y SUS CARACTERISTICAS TECNICAS.

16.1.- SENSOR DE HUMEDAD

Podemos definir la precisión de un sensor como la desviación con respecto a un patrón de laboratorio. Esta característica es afectada por los siguientes factores:

- Temperatura y humedad a la que fue calibrado el sensor
- Dependencia de la calibración con la humedad y la temperatura, muchos sensores son no-lineales y casi todos varían con la temperatura
- Como afecta al sensor el envejecimiento y la velocidad de envejecimiento
- Que tan sensitivo es el sensor a los contaminantes

16.1.1.- sensores resistivos

Los sensores de humedad resistivos están hechos sobre una delgada tableta de un polímero capaz de absorber agua, sobre la cual se han impreso dos contactos entrelazados de material conductor metálico o de carbón.

En la imagen se ve un ejemplo, fabricado por General Eastern. Tiene una longitud de unos 10 mm. Es un componente que se vende independientemente, sin la electrónica necesaria para procesar la medición.

El parámetro que se mide es la resistencia eléctrica a través del polímero, que cambia con el contenido de agua.

16.1.2.- Sensor capacitivo HC201

El HC201 es un sensor capacitivo pensado para uso en aplicaciones de gran escala y efectividad de costo en el control climático de interiores.

En el rango de humedad relativa de 20–90% es posible realizar una aproximación lineal, manteniendo el error en valores menores a $\pm 2\%$ de la humedad relativa medida.

La versión con encapsulado plástico, HC201/H, facilita su montaje en placas de circuito impreso.

16.1.3.- sensor integrado

El SHT11 de Sensirion es un sensor integrado de humedad, calibrado en fábrica y con salida digital. La comunicación se establece a través de un bus serie síncrono, usando un protocolo propio. El dispositivo posee además en su interior un sensor de temperatura para compensar la medición de humedad con respecto a la temperatura, de ser necesario. Cuenta también con un calefactor interno que evita la condensación en el interior de la cápsula de medición en condiciones de niebla o cuando existe condensación.

16.2.- SENSOR DE PRESIÓN BAROMÉTRICA

El transductor de presión de silicio resonante RPT410F utiliza una avanzada tecnología en sensores, suministrando medidas de presión barométrica altamente precisas y

estables.

El sensor proporciona una salida de señal en frecuencia proporcional a la presión. La precisión se mantiene en un amplio rango de temperatura, y el consumo puede optimizarse

mediante una señal de control externa, controlada por el datalogger.

16.3.- SENSORES DE PRECIPITACIÓN

La lluvia, colectada por una boca con un aro de captación de diámetro calibrado, es conducida por medio de un embudo de una sola pieza al receptor interno que descarga sobre un cangilón basculante

Al volcar este un contacto magnético (reed-switch) sin vinculación mecánica con el cangilón, emite una señal de pulso digital.

El ángulo y profundidad del embudo impide rebotes a altas intensidades y mediante un sistema de filtros de malla se impide el pasaje al cangilón de materiales en suspensión e insectos.

Un adecuado diseño del cangilón mantiene la calibración del instrumento en un amplio rango de intensidades.

Construido solo con materiales inoxidables como bronce, acero inoxidable, aluminio, PVC, hacen del instrumento un equipo confiable e inalterable aún en condiciones ambientales severas.

16.4.- SENSOR DE TEMPERATURA DEL AIRE

El sensor de la temperatura del aire de admisión está situado en el sistema de admisión del motor y mide la temperatura del aire. Los valores que mide se utilizan como valores de corrección para la gestión del motor. Los fallos del sensor de la temperatura del aire causan dificultades en el arranque y un consumo de combustible mayor. Las causas de tales fallos son la suciedad, vibraciones o cortocircuitos internos.

Los sensores de temperatura están contruidos con termo resistor lineal de alta exactitud y estabilidad a largo plazo. El envainado en material inoxidable les confiere durabilidad aún en condiciones ambientales adversas.

SENSOR	MODELO	CARACTERISITICAS TECNICAS
DE HUMEDAD	HC-201	<p>Capacidad nominal (a 20 °C): 200 +/-20 pF Sensibilidad: 0,6pF / %RH Humedad, Rango de trabajo:10 .. 95% RH Temperatura, Rango de trabajo: -40 .. 110 °C Error de linealidad (20 .. 90% RH): < +/- 2% RH</p>
DE PRESION BAROMETRICA	RPT410F	<p>Rango de medida: 600hPa a 1100hPa Rango temperatura: -40°C a +60°C Rango Humedad: sin condensación Precisión: -10°C a +50°C ±0.5hPa Consumo: <8mA ON, <10µA OFF Frecuencia de salida: 600Hz a 1100Hz ON 1 a24V DC</p>
DE PRECIPITACION	TS-221	<p>SSensibilidad: 0.10; 0.20; 0.25 o 0.50 mm (0,25 mm estándar) Boca: Aro de captación biselado de 200 mm de diámetro Exactitud: 1 % a 50 mm/h Temperatura de Operación: -20 a + 60°C Detector de vuelco: Reed switch activado por un</p>
TEMPERATURA DEL AIRE	Modelo TS 245/246/247	<p>RangoS-30° a +50°C Exactitud+0.3°C en todo el rango AlimentaciónN<=4000 V estabilizada Consumo<1 Am Impedancia de carga<1MΩ Soporte Va=4A : 716Mv a 2755 Mv Rango de salida Vout [Mv]= Va[mV] x</p>

17.- PAISES DONDE EXISTEN ESTACIONES AGROCLIMATICAS DE ACUERDO A SU TIPO.

En el mundo de las estaciones utilitarias, existen aquellas que tienen como función más importante la de suministrar información relacionada con el clima y los cambios meteorológicos.

Se denomina **Servicio de Ayuda Meteorológica** al servicio de radiocomunicación cuyo objeto consiste en la transmisión de mensajes conteniendo todo tipo de información meteorológica. Dichos mensajes son irradiados por medio de un buen número de estaciones fijas que han sido establecidas en regiones específicas del mundo, las cuales se valen del empleo de la radiotelegrafía (CW), la radiotelefonía (USB), el fax y el radio teletipo (RTTY).

Dentro del espectro que comprende la onda corta (3 a 30 MHz), podemos encontrar varias estaciones que difunden reportes meteorológicos de acuerdo con la finalidad para la cual han sido destinadas. Siguiendo este carácter es posible efectuar la siguiente clasificación:

17.1.- ESTACIONES WMO-GTS

Son todas aquellas estaciones que operan según las pautas regulatorias de la **Organización Meteorológica Mundial** (WMO - World Meteorological Organization), quien conjuntamente con la Unión Internacional de Telecomunicaciones, han establecido el sistema denominado **CTS**, del inglés Global Telecommunications Systems. Este posibilita la recepción de todo tipo de información meteorológica suministrada en determinados «formatos especiales» para facilitar su aprovechamiento por parte de terceros, a quienes esa información les permite estar al tanto de las condiciones climatológicas existentes en cualquier parte del mundo.

Son usuarios de este servicio, las oficinas de meteorología de cada país, los centros de investigación relacionados con el estudio del clima, los medios de información pública (prensa, radio, TV), entre otros.

Los boletines meteorológicos proporcionados por cada una de las estaciones fijas pertenecientes a este servicio, se realizan en base a la investigación y la observación de los fenómenos climáticos que se originan

en la región en donde se hallan situadas. Dicha información es luego transmitida respetando los códigos y metodologías determinadas por la WMO, a fin de asegurar la rápida distribución de los mensajes en favor de los usuarios del servicio. La mayoría de las estaciones están activas durante las 24 horas y sus reportes se actualizan cada hora.

Por lo general las transmisiones en HF se efectúan recurriendo al uso del FAX y el RTTY, lo cual hace posible la recepción de mapas, fotografías, cuadros y demás datos de interés que brindan un completo panorama de la situación climática al momento de recibir cada informe. Los códigos habitualmente utilizados para transmitir los boletines son el PILOT, el SHIP, el SINOP y el TEMP.

A continuación citamos aquí a las estaciones más importantes:

- Auckland METEO (NZL):
- Beijing METEO (CHN):
- Bracknell METEO (G):
- Buenos Aires METEO (ARG):
- Delhi METEO (IND):
- Hamburgo METEO (D):
- Madrid METEO (E):
- Melbourne METEO (AUS):
- Moscú METEO (RUS):
- Roma METEO (I):
- Tokio METEO (J):

Asimismo, existen algunas pocas estaciones que emplean también la telegrafía (CW) y la voz (USB) para la distribución de los partes meteorológicos. Entre ellas podemos citar las siguientes:

- Bangkok METEO (THA):
- Taipei METEO (TWN):
- Tokio METEO (J):

17.2.- ESTACIONES VOLMET:

Son las estaciones fijas del Servicio Móvil Aeronáutico destinadas para realizar observaciones e informes meteorológicos para uso en la navegación aérea internacional. Sobre estas estaciones ampliamos en la página sobre DX aeronáutico en esta sección del sitio.

17.3.- OTRAS ESTACIONES:

Por último, cabe destacar que existen ciertas emisoras que si bien no son propiamente estaciones meteorológicas, suelen irradiar mensajes con información relativa a aquellas condiciones existentes en el área desde donde transmiten.

Tal es el caso de las estaciones costeras pertenecientes al Servicio Móvil Marítimo. Estas emisoras habitualmente difunden partes meteorológicos conteniendo información referente al estado del tiempo y condiciones de navegabilidad en el área cercana a sus costas, con dirección a los navegantes que se encuentran próximos a esa zona.

Las boletines para su difusión, generalmente son proporcionados por las oficinas de meteorología del país a donde pertenece la costera, quienes se ocupan de confeccionarlos en base a las informaciones que extraen de los partes recibidos de las estaciones meteorológicas **WMO-CTS** que hemos mencionado anteriormente.

Entre las estaciones costeras que brindan este servicio podemos rescatar las siguientes:

- L2C - Buenos Aires Prefectura Radio (ARG):
- L2T - Mar del Plata Prefectura Radio (ARG):.
- PPR - Rio Radio (B):
- WOM - Miami Radio (USA):
- ZSC - CapeTown Radio (AFS):

18.- DISEÑO Y COMPONENTES DEL SISTEMA

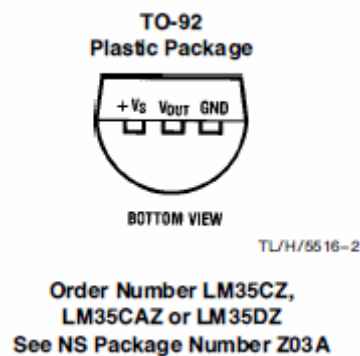
18.1.- SENSOR DE TEMPERATURA LM35.

La serie LM35 son circuitos integrados de precisión que realizan la función de censor de temperatura cuya tensión de salida es linealmente proporcional a la temperatura en la escala Celsius. El LM35 tiene pues una ventaja sobre los sensores de temperatura lineal calibrados en la escala kelvin ya que el usuario no esta obligado a restar una gran constante de voltaje de su producción para obtener la medición correcta.

El LM35 no requiere ninguna calibración externa, de bajo costo garantizado por el hecho de que no requiere calibración externa. El LM35 tiene una alta impedancia de salida, salida lineal y una calibración interna hace especialmente fácil el interfaz de lectura o de control de los circuitos.

Ello se puede utilizar con un solo suministro de energía eléctrica con más suministros.

Como se sabe este dispositivo solo requiere 60 mA y tiene muy baja auto calefacción menos de 0.1°C aun en el aire. El LM35 es apto para operar en un rango de temperatura que va desde -55°C a $+150^{\circ}\text{C}$.



El LM35 se puede aplicar fácilmente en la misma forma que otros circuitos integrados de sensores de temperatura. Puede ser pegados o cementada a una superficie y su temperatura será dentro de aproximadamente 0.01°C de la temperatura de la superficie.

Esto supone que la temperatura del aire es casi la misma que la temperatura de la superficie, si la temperatura del aire es mucho más alta o más baja que la temperatura de la superficie, la temperatura real del LM35 se mantiene en un intermedio entre la temperatura de la superficie y la temperatura del aire. Esto es cierto en el caso del dispositivo que tenemos en el esquema en el se tiene un paquete de plástico, que reduce el calor en el dispositivo.

19.- CONCLUSIONES Y RECOMENDACIONES

El funcionamiento de las estaciones agroclimáticas es en su totalidad muy importante debido a que nos mantiene, como ya se dijo anteriormente, al tanto de los diferentes cambios agroclimáticos o meteorológicos que hayan por parte de la naturaleza; es por ello que el funcionamiento de este proyecto es sin duda alguna de gran relevancia puesto que servirá de mucho como guía para generaciones venideras que poco a poco traten y logren mejorarlo.

Es importante, pues, concluir diciendo que las metas propuestas así como los objetivos previstos han sido hasta ahora alentadores puesto que, aunque nos hemos topado con una serie de obstáculos, estos no han impedido que lleguemos a lo requerido tanto en teoría como en la práctica.

Se espera seguir en la mejora de dicho proyecto para poder satisfacer las necesidades requeridas.

Se recomienda pedir asesoría e información referente al tema para poder hacer funcionar dicho proyecto puesto que, como se puede observar tiene en sí cierto grado de complejidad la cual puede llegar a alterar dicho funcionamiento y dañarlo parcial o generalmente.

20.- REFERENCIAS BIBLIOGRAFICAS

1.- Schmelkes Corina, Manual para la presentación de anteproyectos e informes de investigación, 2da Edición, Oxford, 1998.

2.- MICROCHIP, picdem fs usb demonstration board user's guide.

3.- DON ANDERSON; USB system architecture (USB 2.0) MindShare Inc, Addison-Wesley developer's press.

4.- MICROCHIP, PIC18F2455/2550/4455/4550 Data Sheet

5.- CIRCUITOS ELECTRICOS

CHARLES K. ALEXANDER.

MATTHEW N. O. SADIKU

6.- SISTEMAS DIGITALES

PRINCIPIOS Y APLICACIONES

RONALD J. TOCCI (MONROE COMMUNITY COLLEGE)

NEAL S. WIDMER (PURDUE UNIVERSITY)

PEARSON PRENTICE HALL. OCTAVA EDICION

21.- ANEXOS

PROGRAMA USB 2.0

```
//*****  
  
//  
// File Version 1.01  
  
//  
//*****  
  
#include "Definit.h"  
  
#include "VARs.h"  
  
  
//*****  
  
// The number of bytes in each report,  
  
// calculated from Report Size and Report Count in the report descriptor  
  
//*****  
  
unsigned char const HID_INPUT_REPORT_BYTES = 1;  
  
unsigned char const HID_OUTPUT_REPORT_BYTES = 1;  
  
  
unsigned char const HID_FEATURE_REPORT_BYTES = 2;  
  
//*****  
  
// Byte constants  
  
//*****  
  
unsigned char const NUM_ENDPOINTS = 2;  
  
unsigned char const ConfigDescr_wTotalLength = USB_CONFIG_DESCRIPTOR_LEN + USB_INTERF_DESCRIPTOR_LEN +  
USB_HID_DESCRIPTOR_LEN + (NUM_ENDPOINTS * USB_ENDP_DESCRIPTOR_LEN);  
  
unsigned char const HID_ReportDesc_len = 47;  
  
  
unsigned char const Low_HID_ReportDesc_len = HID_ReportDesc_len;  
  
unsigned char const High_HID_ReportDesc_len = HID_ReportDesc_len >> 8;
```

```

unsigned char const Low_HID_PACKET_SIZE    = HID_PACKET_SIZE;

unsigned char const High_HID_PACKET_SIZE   = HID_PACKET_SIZE >> 8;

//*****

// Descriptor Tables

//*****

unsigned char const DescTables[USB_DEVICE_DESCRIPTOR_ALL_LEN*2] = {

// Device Descriptor

USB_DEVICE_DESCRIPTOR_LEN, 0,    // bLength      - Length of Device descriptor (always 0x12)

USB_DEVICE_DESCRIPTOR_TYPE, 0,   // bDescriptorType  - 1 = DEVICE descriptor

0x00, 0,                        // bcdUSB          - USB revision 2.00 (low byte)

0x02, 0,                        //                (high byte)

0x00, 0,                        // bDeviceClass    - Zero means each interface operates independently (class code in the
interface descriptor)

0x00, 0,                        // bDeviceSubClass

0x00, 0,                        // bDeviceProtocol

EPO_PACKET_SIZE, 0,            // bMaxPacketSize0 - maximum size of a data packet for a control transfer over
EPO

0x34, 0,                        // idVendor        - Vendor ID (low byte)

0x12, 0,                        //                (high byte)

0x01, 0,                        // idProduct       - Product ID (low byte)

0x00, 0,                        //                (high byte)

0x01, 0,                        // bcdDevice       - ( low byte)

0x00, 0,                        //                (high byte)

0x01, 0,                        // iManufacturer   - String1

0x02, 0,                        // iProduct        - String2

```

```

0x00, 0,          // iSerialNumber    - ( None )

0x01, 0,          // bNumConfigurations - 1

// Configuration Descriptor

USB_CONFIG_DESCRIPTOR_LEN, 0,    // bLength          - Length of Configuration descriptor (always 0x09)

USB_CONFIG_DESCRIPTOR_TYPE, 0,    // bDescriptorType  - 2 = CONFIGURATION descriptor

ConfigDesc_wTotalLength, 0,    // wTotalLength     - Total length of this config. descriptor plus the interface
and endpoint descriptors that are part of the configuration.

0x00, 0,          //                ( high byte)

0x01, 0,          // bNumInterfaces   - Number of interfaces

0x01, 0,          // bConfigurationValue - Configuration Value

0x00, 0,          // iConfiguration   - String Index for this configuration ( None )

0xA0, 0,          // bmAttributes     - attributes - "Bus powered" and "Remote wakeup"

50, 0,            // MaxPower         - bus-powered draws 50*2 mA from the bus.

// Interface Descriptor

USB_INTERF_DESCRIPTOR_LEN, 0,    // bLength          - Length of Interface descriptor (always 0x09)

USB_INTERFACE_DESCRIPTOR_TYPE, 0, // bDescriptorType  - 4 = INTERFACE descriptor

0x00, 0,          // bInterfaceNumber - Number of interface, 0 based array

0x00, 0,          // bAlternateSetting - Alternate setting

NUM_ENDPOINTS, 0,          // bNumEndPoints   - Number of endpoints used in this interface

0x03, 0,          // bInterfaceClass  - assigned by the USB

0x00, 0,          // bInterfaceSubClass - Not A boot device

0x00, 0,          // bInterfaceProtocol - none

0x00, 0,          // iInterface       - Index to string descriptor that describes this interface ( None )

// HID Descriptor

USB_HID_DESCRIPTOR_LEN, 0,    // bLength          - Length of HID descriptor (always 0x09)

USB_HID_DESCRIPTOR_TYPE, 0,    // bDescriptorType  - 0x21 = HID descriptor

```

```

0x01, 0,          // HID class release number (1.01)

0x01, 0,

0x00, 0,          // Localized country code (none)

0x01, 0,          // # of HID class descriptor to follow (1)

0x22, 0,          // Report descriptor type (HID)

Low_HID_ReportDesc_len, 0,

High_HID_ReportDesc_len, 0,

// EP1_RX Descriptor

USB_ENDP_DESCRIPTOR_LEN, 0,    // bLength      - length of descriptor (always 0x07)

USB_ENDPOINT_DESCRIPTOR_TYPE, 0,    // bDescriptorType  - 5 = ENDPOINT descriptor

0x81, 0,          // bEndpointAddress  - In, EP1

USB_ENDPOINT_TYPE_INTERRUPT, 0,    // bmAttributes    - Endpoint Type - Interrupt

Low_HID_PACKET_SIZE, 0,          // wMaxPacketSize   - max packet size - low order byte

High_HID_PACKET_SIZE, 0,        //                  - max packet size - high order byte

1, 0,             // bInterval        - polling interval (1 ms)

// EP1_TX Descriptor

USB_ENDP_DESCRIPTOR_LEN, 0,    // bLength      - length of descriptor (always 0x07)

USB_ENDPOINT_DESCRIPTOR_TYPE, 0,    // bDescriptorType  - 5 = ENDPOINT descriptor

0x01, 0,          // bEndpointAddress  - Out, EP1

USB_ENDPOINT_TYPE_INTERRUPT, 0,    // bmAttributes    - Endpoint Type - Interrupt

Low_HID_PACKET_SIZE, 0,          // wMaxPacketSize   - max packet size - low order byte

High_HID_PACKET_SIZE, 0,        //                  - max packet size - high order byte

1, 0,             // bInterval        - polling interval (1 ms)

// HID_Report Descriptor

0x06, 0,          // USAGE_PAGE (Vendor Defined)

0xA0, 0,

```



```

0xFF, 0,
0x09, 0,          // USAGE ID (Vendor Usage 1)
0x01, 0,
0xA1, 0,          // COLLECTION (Application)
0x01, 0,
// The Input report
0x09, 0,          // USAGE ID - Vendor defined
0x03, 0,
0x15, 0,          // LOGICAL_MINIMUM (0)
0x00, 0,
0x26, 0,          // LOGICAL_MAXIMUM (255)
0x00, 0,
0xFF, 0,
0x75, 0,          // REPORT_SIZE (8)
0x08, 0,
0x95, 0,          // REPORT_COUNT (2)
HID_INPUT_REPORT_BYTES, 0,
0x81, 0,          // INPUT (Data,Var,Abs)
0x02, 0,
// The Output report
0x09, 0,          // USAGE ID - Vendor defined
0x04, 0,
0x15, 0,          // LOGICAL_MINIMUM (0)
0x00, 0,
0x26, 0,          // LOGICAL_MAXIMUM (255)
0x00, 0,
0xFF, 0,
0x75, 0,          // REPORT_SIZE (8)
0x08, 0,

```

```

0x95, 0,          // REPORT_COUNT (2)
HID_OUTPUT_REPORT_BYTES, 0,
0x91, 0,          // OUTPUT (Data,Var,Abs)
0x02, 0,
// The Feature report
0x09, 0,          // USAGE ID - Vendor defined
0x05, 0,
0x15, 0,          // LOGICAL_MINIMUM (0)
0x00, 0,
0x26, 0,          // LOGICAL_MAXIMUM (255)
0x00, 0,
0xFF, 0,
0x75, 0,          // REPORT_SIZE (8)
0x08, 0,
0x95, 0,          // REPORT_COUNT (2)
HID_FEATURE_REPORT_BYTES, 0,
0xB1, 0,          // FEATURE (Data,Var,Abs)
0x02, 0,
// End Collection
0xC0, 0           // END_COLLECTION
};
//*****
unsigned char const LangIDDescr[8] = {
    0x04, 0,
    USB_STRING_DESCRIPTOR_TYPE, 0,
    0x09, 0,          // LangID (0x0409) - Low
    0x04, 0           //          - High
};
//*****

```

```

unsigned char const ManufacturerDescr[68] = {

    34, 0,

    USB_STRING_DESCRIPTOR_TYPE, 0,

    'm', 0, 0, 0,
    'i', 0, 0, 0,
    'k', 0, 0, 0,
    'r', 0, 0, 0,
    'o', 0, 0, 0,
    'E', 0, 0, 0,
    'l', 0, 0, 0,
    'e', 0, 0, 0,
    'k', 0, 0, 0,
    't', 0, 0, 0,
    'r', 0, 0, 0,
    'o', 0, 0, 0,
    'n', 0, 0, 0,
    'i', 0, 0, 0,
    'k', 0, 0, 0,
    'a', 0, 0, 0

};

//*****

unsigned char const ProductDescr[96] = {

    48, 0,

    USB_STRING_DESCRIPTOR_TYPE, 0,

    'm', 0, 0, 0,
    'i', 0, 0, 0,
    'k', 0, 0, 0,
    'r', 0, 0, 0,
    'o', 0, 0, 0,

```

```

'C', 0, 0, 0,
',', 0, 0, 0,
'H', 0, 0, 0,
'I', 0, 0, 0,
'D', 0, 0, 0,
',', 0, 0, 0,
'L', 0, 0, 0,
'i', 0, 0, 0,
'b', 0, 0, 0,
'r', 0, 0, 0,
'a', 0, 0, 0,
'r', 0, 0, 0,
'y', 0, 0, 0,
',', 0, 0, 0,
't', 0, 0, 0,
'e', 0, 0, 0,
's', 0, 0, 0,
't', 0, 0, 0
};

//*****

unsigned char const StrUnknownDescr[4] = {

    2, 0,

    USB_STRING_DESCRIPTOR_TYPE, 0
};

//*****

//*****

// Initialization Function

//*****

```

```

void InitUSBdsc()
{
Byte_tmp_0[0] = NUM_ENDPOINTS;
Byte_tmp_0[0] = ConfigDescr_wTotalLength;
Byte_tmp_0[0] = HID_ReportDesc_len;
Byte_tmp_0[0] = Low_HID_ReportDesc_len;
Byte_tmp_0[0] = High_HID_ReportDesc_len;
Byte_tmp_0[0] = Low_HID_PACKET_SIZE;
Byte_tmp_0[0] = High_HID_PACKET_SIZE;

DescTables;

LangIDDescr;
ManufacturerDescr;
ProductDescr;
StrUnknownDescr;

}

//*****

```

PROGRAMA BASICO DE TRANSFERENCIA

```

/*
* Project name:
    HIDtest (Testing the USB HID connection)
* Copyright:
* Test configuration:

```

```

MCU:      PIC18F87J50

Dev.Board:  LV-18FJ

Oscillator:  HS-PLL, 8.000 MHz Crystal, 48.000 MHz MCU Clock

Ext. Modules:  -

SW:      mikroC v8.1

*/

unsigned char k;

unsigned char userWR_buffer[64];

const char * text = "MIKROElektronika Compilers ER \r\n";

//*****
*

// Main Interrupt Routine

//*****
*

void interrupt()
{
    HID_InterruptProc();
}

//*****
*

//*****
*

// Initialization Routine

//*****
*

```

```

void Init_Main() {

    //-----

    // Disable all interrupts

    //-----

    INTCON = 0x00;          // Disable GIE, PEIE, TMROIE,INTOIE,RBIE

    INTCON2 = 0xF8;

    INTCON3 = 0x00;

    RCON.IPEN = 0;        // Disable Priority Levels on interrupts

    PIE1 = 0;

    PIE2 = 0;

    PIE3 = 0;

    PIR1 = 0;

    PIR2 = 0;

    PIR3 = 0;

    ANCON0 = 0x9F;        // Configure all ports with analog function as digital

    ANCON1 = 0xFC;

    //-----

    // Ports Configuration

    //-----

    TRISA = 0x3F;

    TRISB = 0xFF;

    TRISC = 0xFF;

    TRISD = 0xFF;

    TRISE = 0xFF;

    TRISF = 0xFC;

    TRISG = 0x1F;

    LATA = 0;

```

```

LATB = 0;

LATC = 0;

LATD = 0;

LATE = 0;

LATF = 0;

LATG = 0;

//-----
// Enable PLL !!!
//-----

OSCTUNE.PLLEN = 1;

Delay_mS(5);

}

//*****
*

//*****
*

// Main Program Routine

//*****
*

void main() {

    unsigned char i,k;

    Init_Main();

    HID_Enable(&userWR_buffer, &userWR_buffer);

    Delay_ms(1000);

    Delay_ms(1000);

```



```

        k = 100;

        while (k != 0)
            {
Delay_ms(1000);

                i = 0;

                while (text[i])
                    {

                        userWR_buffer[0] = text[i++];

                        while ( !HID_Write(&userWR_buffer, 1) );

                    }

                k--;

            }

Delay_ms(1000);

        HID_Disable();

    }

//*****
*
```

PROGRA MA PILOTO

```

unsigned short m, k;
unsigned short userRD_buffer[64];
unsigned short userWR_buffer[64];

void interrupt() {
    asm CALL _Hid_InterruptProc
    asm nop
} //~

void Init_Main() {
    // Disable all interrupts
    // Disable GIE, PEIE, TMROIE, INTOIE, RBIE
    INTCON = 0;
    INTCON2 = 0xF5;
    INTCON3 = 0xC0;
}
```

```

// Disable Priority Levels on interrupts
RCON.IPEN = 0;
PIE1 = 0;
PIE2 = 0;
PIR1 = 0;
PIR2 = 0;

// Configure all ports with analog function as digital
ADCON1 |= 0x0F;

// Ports Configuration
TRISA = 0;
TRISB = 0;
TRISC = 0xFF;
TRISD = 0xFF;
TRISE = 0x07;

LATA = 0;
LATB = 0;
LATC = 0;
LATD = 0;
LATE = 0;

// Clear user RAM
// Banks [00 .. 07] ( 8 x 256 = 2048 Bytes )
asm {
    LFSR    FSR0, 0x000
    MOVLW  0x08
    CLRF   POSTINC0, 0
    CPFSEQ FSR0H, 0
    BRA    $ - 2
}

// Timer 0
TOCON = 0x07;
TMR0H = (65536-156) >> 8;
TMR0L = (65536-156) & 0xFF;
INTCON.T0IE = 1;           // Enable T0IE
TOCON.TMR0ON = 1;
};//~

/** Main Program Routine */
void main() {
    Init_Main();
    Hid_Enable(&userRD_buffer, &userWR_buffer);

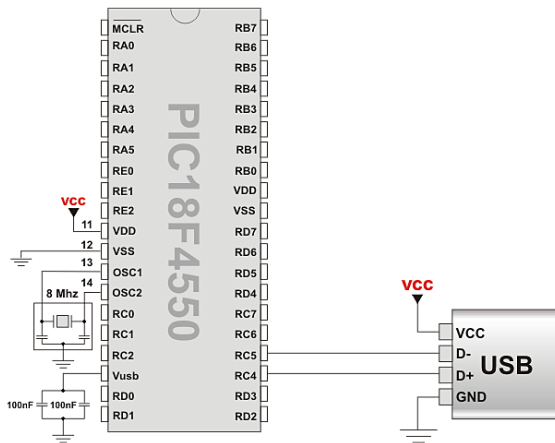
    do {
        for (k = 0; k < 255; k++) {
            // Prepare send buffer
            userWR_buffer[0] = k;

            // Send the number via usb
            Hid_Write(&userWR_buffer, 1);
        }
    } while (1);

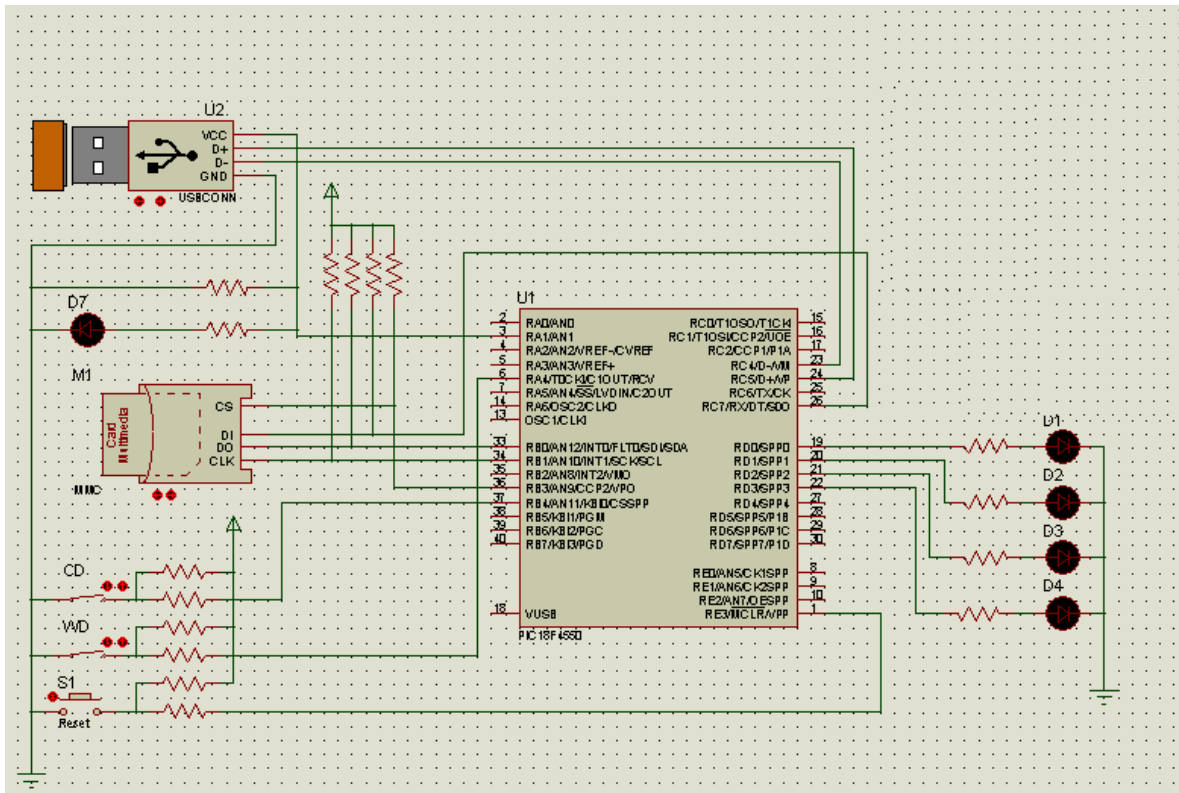
    Hid_Disable();
};//~!

```

HW CONEXION



ESQUEMA DE CONEXIÓN USB



IMÁGENES DEL CIRCUITO

