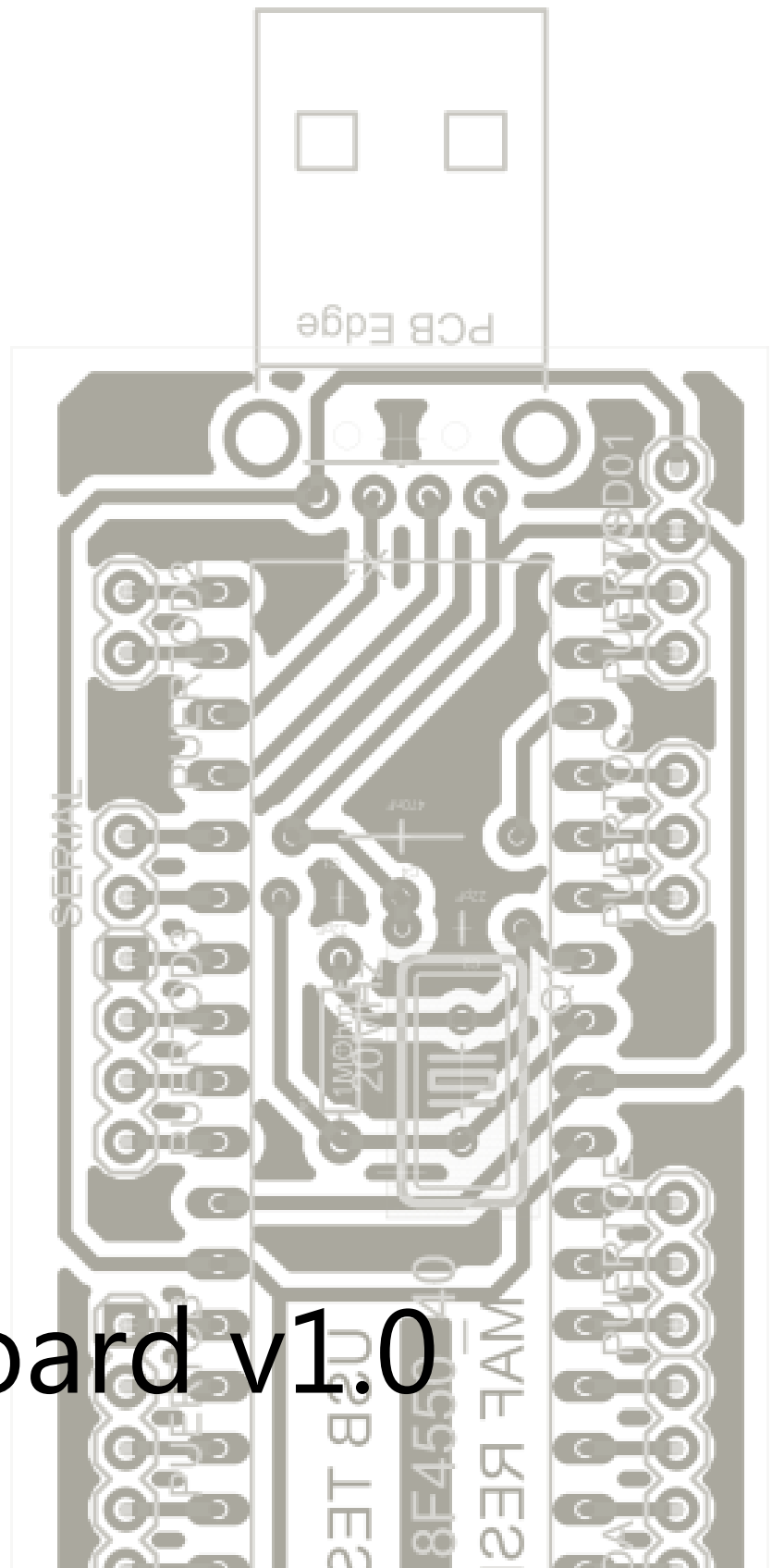


Powered by **MAF Research**





USB proBoard es un proyecto que surge después de la incursión de MAF Research en electrónica y de la generación de proyectos como "LightSense". Durante los últimos años MAF Research se ha involucrado en el desarrollo y uso de dispositivos que permitan conectar USB a otros dispositivos electrónicos. Es debido a esto MAF impulsa la creación y difusión de este proyecto como un Open-Source Hardware.

Objetivo

Este proyecto busca desarrollar prototipos que sirvan a la inducción de estudiantes y/o profesionales a la creación, desarrollo, construcción y programación de dispositivos USB. A la vez se busca mantener el diseño libre y abierto para más desarrolladores.

Introducción

Actualmente podemos desarrollar y/o consultar herramientas que nos permiten acceder a información como humedad, temperatura, luminosidad (variables que se encuentran a nuestro alrededor) y/o controlar dispositivos eléctricos como focos, robots, cerraduras, etc. Todo esto se logra gracias al uso de dispositivos como los PIC's (Programmable Integrated Circuit) desarrollados por Microchip.

Estos dispositivos cuentan con características como conversor Analógico-Digital, conversor Digital-Analógico, CPU, puertos seriales de comunicación, salidas digitales, puerto de comunicación USB, memoria ROM y RAM para cargar/ejecutar la programación y puertos de comunicación con otros circuitos integrados.

Gracias a estas características podemos realizar proyectos como los mencionados anteriormente. A continuación, en este documento, se abordará el desarrollo de circuitos que nos permitan medir variables físicas con USB proBoard.

¿Qué es una variable física?

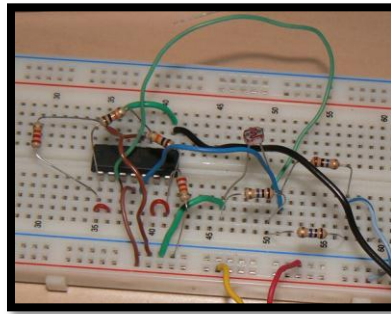
Una variable física es una unidad que se puede medir como temperatura, humedad, presión, intensidad de luz, etc. En este caso vamos a medir la temperatura que se presente en el ambiente. La temperatura se mide en grados Celsius, Kelvin o Fahrenheit pero en este caso, para fines prácticos, vamos a medir en grados Celsius.

¿Qué componentes me ayudaran a realizar este proyecto?

En el mundo de los componentes electrónicos hay mucho que elegir pero en este caso haremos uso de un circuito con un PIC para poder desarrollar la aplicación deseada. En este caso vamos a dividir el circuito en 2 partes:

- Circuito Sensor
- Circuito acondicionador y de comunicación

Circuito Sensor

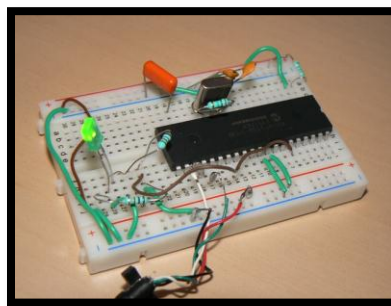


Este circuito es el más importante debido a que mide la unidad física deseada y nos entrega un voltaje de acuerdo a la cantidad de la unidad medida. Este voltaje será transmitido a nuestro circuito acondicionador para que realice sus funciones correspondientes.

Algo importante de este circuito es que es adaptativo. El hecho de que sea adaptativo significa que cambiando un solo componente nos puede medir gran cantidad de variables físicas y con esto desarrollar más este circuito que es una base para aplicaciones más complejas.

En el caso de este circuito se utilizara un LM35 que entrega 1°C por cada 10mVolts, es resistente al agua, maneja un rango de -55°C a 150°C y opera entre 4 y 30 Volts.

Circuito acondicionador y de comunicación (USB proBoard)



Este circuito recibe un voltaje variable analógico del circuito sensor y lo convierte en una variable digital. Esta variable digital será procesada y entregada a la computadora vía puerto USB. En este caso el circuito, aunque es USB, emulara un puerto RS-232 y por este puerto leeremos la cantidad medida en la computadora.

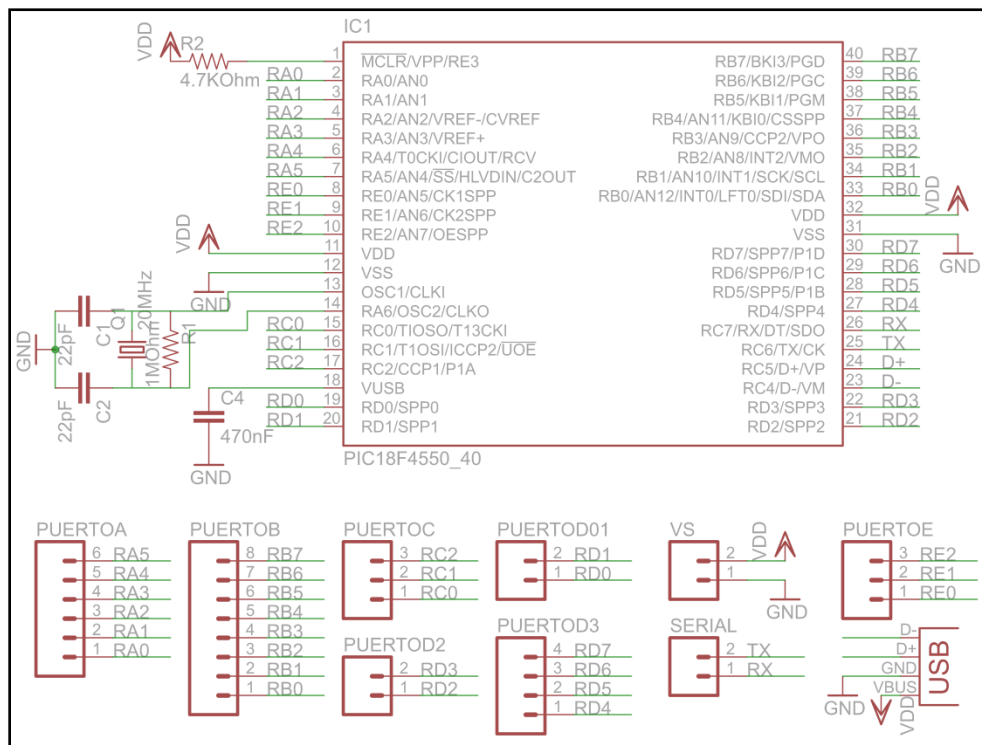
Para que todo esto sea posible se hará el uso de un PIC18F4550. Este PIC se programara para que procese nuestra variable analógica, la digitalice, la procese, se conecte a la computadora y nos muestre el formato final de la variable.

Desarrollo

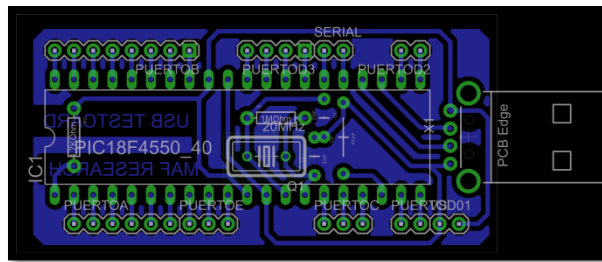
Para el desarrollo de la USB proBoard (Circuito acondicionador y de comunicación) se necesitaran los siguientes componentes:

- Un PIC18F4550 en configuración DIP (Dual In-line Package).
- Resistencias de 1 MOhm y 4.7KOhm.
- Cristal resonador de 20MHz.
- Capacitores cerámicos de 22pF y de poliéster de 470nF.
- Un conector USB tipo "A" macho.
- Base para C.I. de 40 pines.
- 1 Tira de conectores hembra.
- Placa fenólica de 10cmX10cm
- Hojas tamaño carta de papel couche de 100gm o menor gramaje.
- Un bote de cloruro férrico.

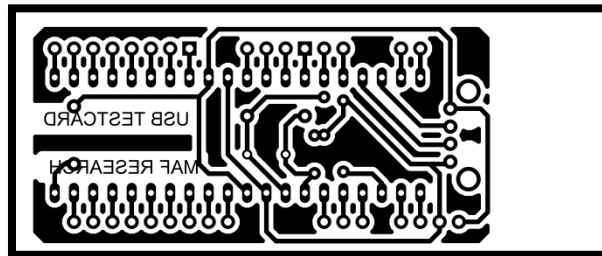
Estos no servirán para crear el circuito que se muestra en el siguiente esquemático:



En este esquemático se muestran los componentes y conexiones que se hacen con el PIC para poder desarrollar este circuito. Este esquema nos ayuda a realizar el diseño de nuestro circuito impreso y por ende la forma en la que quedara en nuestro PCB, a continuación se muestra la imagen del diseño PCB con componente y el diseño para transferirse a placa:



Diseño con componentes



Diseño para transferirse a placa

Para transferir este diseño se deberá imprimir este en una hoja de papel couche de 100gm o de menor gramaje. Después de la impresión pondremos la cara que tiene el tóner del diseño encima de nuestra placa de cobre de 10cmX10cm y presionarlo con una Plancha encendida a la más alta temperatura durante 5 minutos. Dejaremos reposar durante 5 minutos la placa y la pondremos en un recipiente con agua para que se desprenda la hoja de papel couche. Habiendo terminado lo anterior deberemos sumergir la placa en Cloruro de Férrico para quitar el cobre que no cubre el tóner (excedente), perforar la placa, colocar y soldar nuestros componentes.

Ya con esto tendremos nuestro circuito finalizado pero antes de realizar cualquier prueba en la computadora se deberá que programar el PIC para que la computadora lo pueda reconocer. Para esto vamos usar un programa llamado "PIC C Compiler". Este programa nos ayuda a programar el PIC en lenguaje C y transformarlo en .HEX para que pueda comprender el PIC lo que se le ordena.

La programación del PIC deberá comprender la funcionalidad de un conversor Analógico-Digital, la conexión vía USB con la computadora y el envío de los datos adquiridos por el AD por el puerto USB.

Iniciamos el programa con la inclusión de las librerías para el uso de las funciones del PIC y se definen variables que se usaran dentro de la funcionalidad del PIC:

```
#include <18F4550.h> //Libreria del PIC18F4550
#define device adc=10 //Usamos un ADC de 10 Bits.
#define fuses HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,VREGEN //Se habilitan registros para el uso del PIC
#define use delay(clock=48000000) //Se usa un reloj de 48MHz para la conexión USB
#define define USB_CON_SENSE_PIN PIN_B2 //Se habilita para usar un reloj de 20MHz físico
#include <usb_cdc.h> //Libreria con funciones de conexión USB
```

Después declaramos la rutina principal del programa:

```
void main()
{
    int16 q,q1; //Variables que almacenan los datos del ADC
    float p; //Variable temporal que almacena datos convertidos del ADC
    q1=0; //Inicializamos q1 a cero para evitar basura en el programa

    setup_adc_ports(AN0|VSS_VDD); //Habilitamos el Puerto AN0 como ADC y con Vres maximo de VSS
    setup_adc(ADC_CLOCK_INTERNAL); //Usamos el reloj interno para el ADC
    set_adc_channel(0); //Usamos el canal 0 para el ADC

    usb_cdc_init(); //Inicializamos la conexión USB
    usb_init();
}
```

```

do
{
    usb_task();
    if (usb_enumerated())
    {
        delay_ms(500);
        q = read_adc();
        if (q!=q1)
        {
            p = 5.0 * q / 1024.0;
            printf(usb_cdc_putc, "\r Voltage=%01.2fv", p);
        }
        q1=q;
    }
} while (TRUE);
}

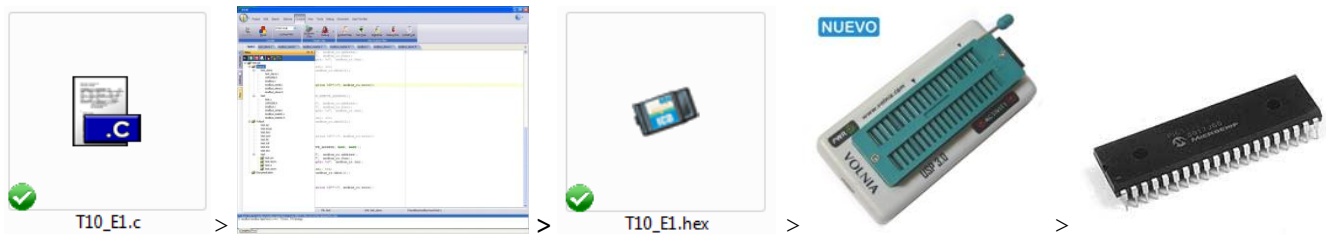
```

```

//Siempre se realizan estas acciones
//Enumeramos el dispositivo
//Si el dispositivo esta enumerado
//Nos esperamos 500 milisegundos
//Leemos el Puerto ADC
//Si q es diferente de q1
//P adquiere el valor del voltaje en el ADC convertido a decimal
//Introducimos en el Puerto USB la cadena 'Voltaje = 5v'
//q1 adquiere el valor de q

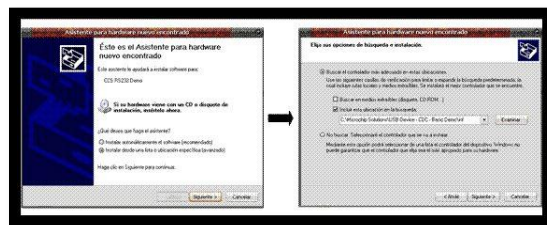
```

Compilamos este programa dentro de "PIC C Compiler", este nos generara un archivo ".hex" y con ayuda de un programador de PIC's podemos cargar nuestro programa al PIC.



Cabe mencionar que este programa entrega el Voltaje recibido en el ADC y para efectos d hacer mediciones de variables fisica se debe hacer la conversión pertinente de voltaje a medida fisica entregada por el sensor usado.

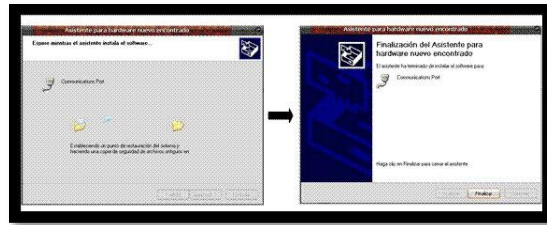
Al momento de conectar el PIC a la maquina deberemos de instalar el controlador de este para que se pueda usar (dar clic aquí para bajar controlador), los pasos son los siguientes en Windows XP (este circuito ofrece soporte a cualquier sistema operativo Windows superior a XP).



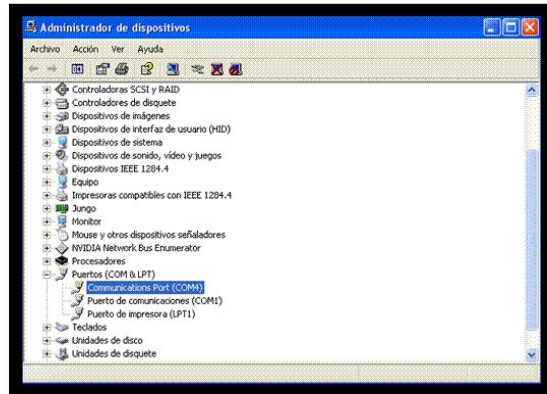
Detección del dispositivo y petición del driver.



Identificación y confirmación de controladores.



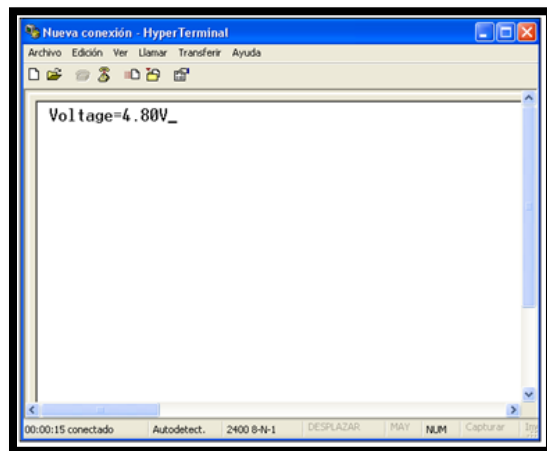
Instalación del dispositivo.



Detección del puerto RS-232 virtual (en este caso es el puerto 4).

Después de instalar el dispositivo notaremos que instalo un puerto serial en vez de algún dispositivo USB. Esto es debido a que el PIC emula un puerto serial mediante una conexión USB. Al momento de crear nuestras interfaces de software con el PIC deberemos que adquirir y enviar datos vía puerto serial.

En caso de pruebas podemos recurrir a la Hyperterminal de Windows XP o "Putty" para Windows Vista o 7 para saber que datos nos esta enviando el PIC seleccionando Puerto COM que esta usando nuestro dispositivo.

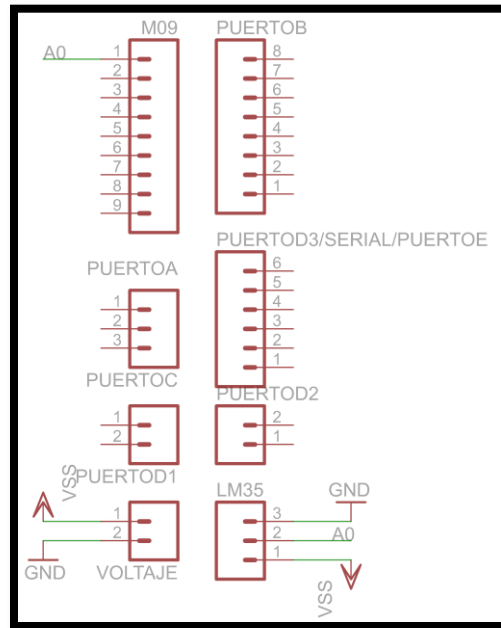


Circuito Sensor

Gracias a la modularidad y el diseño de USB proBoard se puede generar circuitos independientes y modulares que nos permitan extender su uso. Para poner en práctica nuestra USB proBoard crearemos un circuito que nos permita la adquisición de una variable física y que nos entregue vía cantidad de voltaje la cantidad de esta variable. Para este caso haremos un sensor de temperatura que utilizara un LM35 y requerirá del siguiente material:

- Un LM35 (sensor de temperatura).
- Placa fenólica de 10cmX10cm.
- Hojas tamaño carta de papel cuche de 100gm o de menor gramaje.
- 2 líneas de conectores macho.

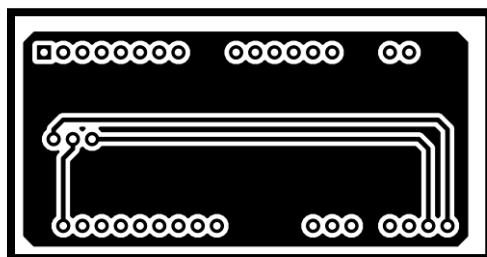
Estos no servirán para crear el circuito que se muestra en el siguiente esquemático:



En este esquemático se muestran los componentes y conexiones que se hacen con el PIC para poder desarrollar este circuito. Este esquema nos ayuda a realizar el diseño de nuestro circuito impreso y por ende la forma en la que quedara en nuestro PCB, a continuación se muestra la imagen del diseño PCB con componente y el diseño para transferirse a placa:



Diseño con componentes



Diseño para transferirse a placa

Para transferir este diseño se deberá imprimir este en una hoja de papel couche de 100gm o de menor gramaje. Después de la impresión pondremos la cara que tiene el tóner del diseño encima de nuestra placa de cobre de 10cmX10cm y presionarlo con una Plancha encendida a la más alta temperatura durante 5 minutos. Dejaremos reposar durante 5 minutos la placa y la pondremos en un recipiente con agua para que se desprenda la hoja de papel couche. Habiendo terminado lo anterior deberemos sumergir la placa en Cloruro de Férrico para quitar el cobre que no cubre el tóner (excedente), perforar la placa, colocar y soldar nuestros componentes.

Ya terminado este proceso podemos acoplar nuestra placa del sensor al circuito del USB proBoard y empezar a hacer pruebas en el PC. Cabe mencionar que en la programación del USB proBoard deberemos que cambiar algunas líneas del programa en C y recompilarlo para poder obtener el formato de temperatura y no de voltaje añ conectarlo al PC. Las líneas que se deben que cambiar son:

```
p = 5.0 * q / 1024.0; //P adquiere el valor del voltaje en el ADC convertido a decimal
printf(usb_cdc_putc, "\r Voltage=%01.2fV", p);
```

Se deben cambiar por:

```
p = (5.0 * q / 1024.0) * 100; //P adquiere el valor del voltaje en el ADC convertido a decimal
//Se multiplica por 100 para el caso de un LM35 ya que entrega
//1mV por Grado Centígrado
printf(usb_cdc_putc, "\r Temperatura = %01.2fV", p);
```

Después de cambiar las líneas de código, recompilar, generar un nuevo ".hex" y cargarlo en el PIC deberemos que obtener en nuestra maquina la cantidad de temperatura que esta obteniendo el sensor.

Conclusiones

USB proBoard es un proyecto que maximiza el uso del PIC18F4550 y nos permite generar un numero sin fin de aplicaciones gracias a su modularidad. El uso de puertos USB y de dispositivos electrónicos para adquirir información y/o controlar dispositivos es algo que va en aumento en la industria y las aplicaciones comerciales. El objetivo de este proyecto se ve alcanzado al acercar a profesionales, estudiantes y profesores al desarrollo de dispositivos USB gracias a su diseño y simplicidad. En un futuro este dispositivo ira mejorando para poder innovar y simplificar su uso.

Bibliografía

Compilador C CSS y simulador Proteus para microcontroladores PIC por Eduardo Garcia. Editorial AlfaOmega.

Referencias

LightSense, presentación y documentación

<http://mafmiicrosoft.wordpress.com/2008/12/16/%E2%80%9CLightsense%E2%80%9D-presentacion-y-documentacion/>