

Desarrollo de un adquisidor de datos mediante puerto USB con fines didácticos.

Mariano H. Méndez⁽¹⁾, Alfonso Pons⁽¹⁾, Laura Angelone^(1,2)

(1) Facultad de Ciencias Exactas, Ingeniería y Agrimensura – UNR- Rosario, Argentina

(2) CIFASIS-Conicet, Rosario, Argentina

Correo-e: {mmendez, alfonso, langelon}@fceia.unr.edu.ar

Resumen. En el presente trabajo se expone el desarrollo de un sistema basado en microcontrolador para proporcionar una interfaz para la enseñanza de la adquisición de datos en el primer año de Ingeniería Electrónica e Ingeniería Eléctrica de la FCEIA. Haciendo uso del puerto USB los estudiantes pueden interactuar a través de este hardware con sensores y actuadores reales. De esta manera se presenta a los mismos un enfoque de la Informática en tareas muy comunes en el ambiente industrial y/o donde se desempeñarán como futuros ingenieros. Esto permite que los estudiantes interioricen los conocimientos adquiridos de Algoritmia y Programación en el primer curso de Informática, y aprendan, desde el inicio de sus estudios, el uso y aplicación de microcontroladores y software libre.

1 Introducción

1.1 Motivación

Las asignaturas de Informática en los primeros años de las carreras de Ingeniería de la Facultad de Ciencias Exactas, Ingeniería y Agrimensura (FCEIA) están planteadas en base a la resolución de problemas utilizando algoritmos y diferentes lenguajes de programación. Siempre en situaciones que se resuelven dentro de la misma computadora, es decir, sin interacción con el exterior. Esto restringe la visión que los estudiantes pueden armarse en cuanto al uso de computadoras en ámbitos industriales, de control o de procesamiento de datos externos a la misma. El planteo es la introducción al manejo de puertos para el intercambio de información hacia y desde la computadora con el fin de adquirir datos, procesarlos y enviar respuestas; actividades muy comunes en el ambiente industrial en que se desenvolverán como futuros ingenieros.

Debido a que los estudiantes de los primeros años aún no poseen conocimientos suficientes para el diseño de hardware y, además, este diseño no es objetivo de las asignaturas de informática, se pretende desarrollar un hardware didáctico de adquisición de datos con el que los estudiantes puedan interactuar mediante la escritura de programas para lograr la comunicación, el envío y recepción de datos desde y hacia el mismo.

Se propone el uso del lenguaje C por varias razones. Primero por ser un lenguaje que combina las características de alto y bajo nivel, por su potencia y robustez a la hora de manejar librerías, y por la facilidad que presenta en las tareas de desarrollo,

reutilización de código, depuración y mantenimiento. Segundo por tener compiladores para cualquier tipo de arquitectura y sistemas operativos, sobre todos los de distribución libre. Tercero por ser el lenguaje que se utiliza en las asignaturas de Informática en el primer año de las carreras de Ingeniería Electrónica y Eléctrica.

En cuanto al puerto de comunicaciones, se utiliza el Universal Serial Bus (USB) que nació a finales de 1995 con el objetivo de ofrecer comunicaciones confiables entre una PC y periféricos externos a mayor velocidad de lo que se venía haciendo hasta ese momento y, al mismo tiempo, unificar los conectores entre la PC y estos periféricos. Desde su concepción hasta hoy en día, las especificaciones USB tuvieron modificaciones para aprovechar el actual potencial de las computadoras y mejorar las transmisiones. Actualmente está disponible la versión 3.0 de USB, sin embargo la versión 2.0 es la que aún tiene mayor difusión y con la que casi todos los periféricos actuales soportan.

1.2 Objetivos

El objetivo de este trabajo es desarrollar un sistema adquisidor de datos a través del puerto USB para mejorar la formación del estudiante de los primeros años de Ingeniería Electrónica e Ingeniería Eléctrica. Aportando a la experimentación que necesita el proceso educativo de los mismos, ampliando su visión de las aplicaciones en la práctica ingenieril real desde el inicio de sus estudios, y afianzando el uso de la Algoritmia y la Programación para resolver situaciones reales, permitiendo a los estudiantes que armen el hardware y prueben su funcionamiento.

2 Hardware

En la Fig.1 se muestra el esquema básico del sistema, donde la interfaz acondiciona los valores de los sensores para enviar a la PC por el puerto USB.

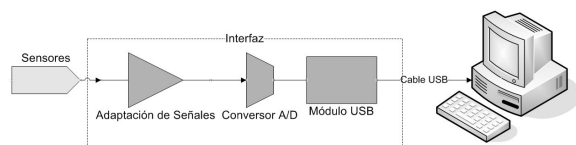


Figura N°1: Esquema del sistema

La placa de adquisición de datos puede tener sensores analógicos y digitales, mientras que las salidas serán solo digitales.

2.1 Interfaz

La placa adquisidora de datos es la responsable de comunicar a los sensores y salidas con la PC. Para lograr esto es necesario un dispositivo capaz de traducir los datos que entregan los sensores para ser comprendidos por la PC y, al mismo tiempo, codificar las instrucciones de la PC a las salidas de la placa. Todo esto puede ser resuelto mediante el uso de un microcontrolador.

Existen distintos fabricantes de microcontroladores y cada uno de ellos ofrece diferentes modelos con diversos periféricos y prestaciones. El modelo elegido para la interfaz es el PIC18F2550 de Microchip. Este modelo de microcontrolador se adapta a los requerimientos del sistema, ya que ofrece: Conversor Analógico/Digital de 10 bits incorporado, módulo de comunicaciones USB 2.0 y operan con tensión de alimentación de 5V, lo que permite alimentar al microcontrolador directamente con la tensión del bus USB.

2.2 Sensores y Señales de Entrada

Con el fin de proporcionar en la misma placa los sensores y tratar que éstos sean lo más próximos a situaciones reales, se utilizan potenciómetros y un sensor de temperatura analógico integrado. De esta forma se logra asemejarse a sensores industriales que, por lo general, entregan una tensión que varía en relación directa con la magnitud que están midiendo. También el sistema cuenta con entradas digitales hechas con pulsadores tipo Tac-Switch.

2.3 Salidas Digitales

Las salidas digitales están implementadas con LEDs y relees. Por un lado con 8 LEDs que se iluminan indicando un estado digital alto (la cantidad de LEDs es para visualizar valores de variables de 1 Byte) y, por el otro, por 2 relees simples inversores que dan la capacidad al sistema de accionar cargas que requieran un gran consumo de corriente como ser lámparas incandescentes, motores, etc.

2.4 Circuito Eléctrico

En el circuito de la Fig.2 pueden verse todos los elementos que conforman el sistema y la forma en que estos están conectados.

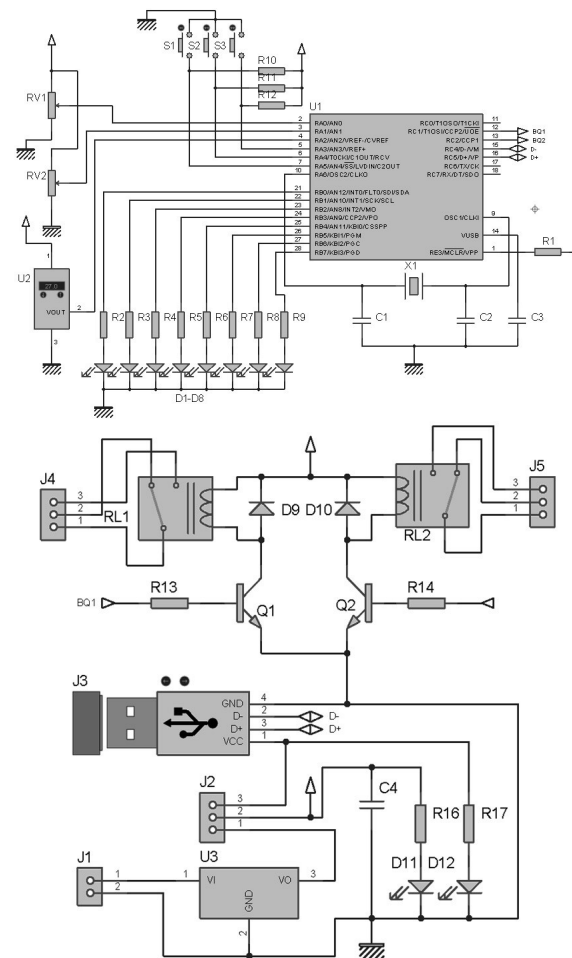


Figura 2: Circuito Eléctrico

En la Tabla 1 se listan los componentes utilizados con sus valores comerciales y funciones en el sistema de adquisición de datos.

Tabla 1: Listado de Componentes

Denominación	Valor	Función
C1, C2	22 pF	Circuito de Reloj
C3	470 nF	Desacople de V_{USB}
D1 a D8	LED Verde	Salidas Digitales
D9, D10	1N4007	Protección de Q1 y Q2
D11	LED Rojo	Indicador de alimentación
D12	LED Amarillo	Indicador de Conexión USB
J1	Conector DC	Alimentación externa (12V)

Denominación	Valor	Función
J2	Jumper	Selección de alimentación
J3	Conector USB-B	Comunicación con PC
J4, J5	Bornera 3 pines	Salidas de releos
Q1, Q2	BC537	Drivers de releos
R1, R10-R12	10 K Ω	Pull-Up
R2-R8, R13-R17	560 Ω	Resistencias Limitadoras
RV1, RV2	5 K Ω	Entradas analógicas
S1-S3	Tac-Switch	Entradas Digitales
U1	PIC18F2550	Microcontrolador
U2	LM35DZ	Sensor de Temperatura
U3	7805	Regulador de Tensión

2.5 Circuito Impreso

En la Fig.3 se muestra el circuito impreso para el adquirente de datos del lado cobre, mientras que en la Fig.4 puede verse el lado de los componentes con la identificación de los mismos.

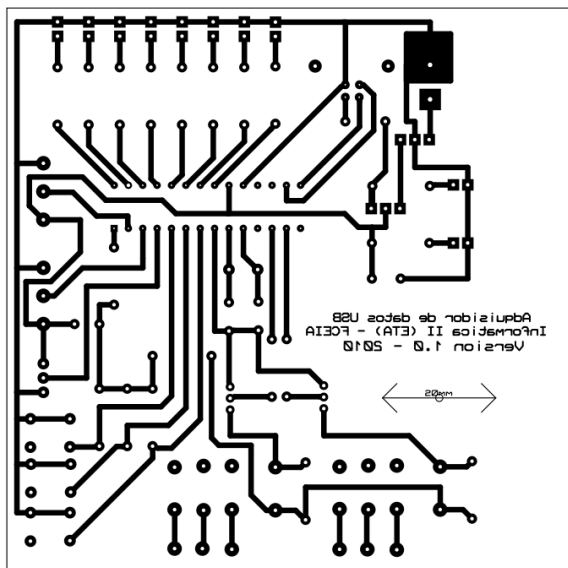


Figura N° 3: PCB lado cobre

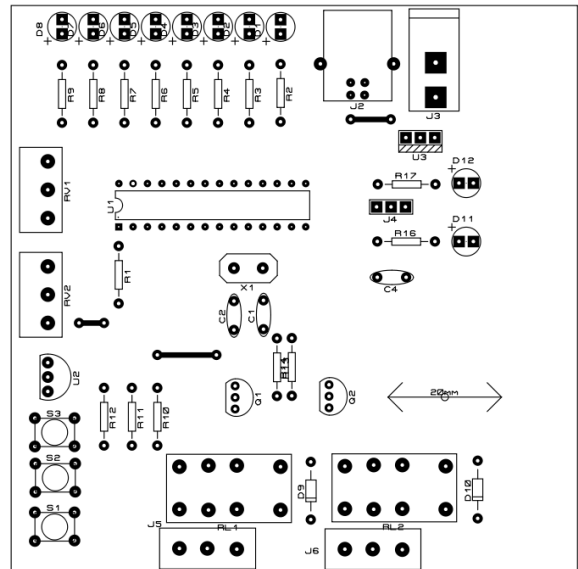


Figura N°4: PCB lado componentes

3 Firmware

El firmware, conjunto de instrucciones que comanda el funcionamiento del microcontrolador, asume tres tareas:

1. convierte los datos que entregan las señales de entrada;
2. comanda las salidas según los datos recibidos; y
3. controla el envío y recepción de datos por medio del puerto USB. Estas tareas son implementadas en simultáneo y según el dato recibido por el bus USB se realiza una u otra conversión.

3.1 Modo de Transferencia USB^{[1][4]}

El protocolo USB establece cuatro formas distintas de transferencias: de control, isócronas, de interrupción y de transferencias bulk o "paquetes". Para esta placa se utilizó el modo de transferencia bulk la cual hace uso de los endpoints USB y permite aprovechar todo el ancho de banda de las especificaciones USB 2.0.

El microcontrolador está continuamente chequeando si llegan nuevos paquetes de datos por algún endpoint USB de los 16 disponibles. En caso que esto ocurra, ejecutará las tareas correspondientes de acuerdo al valor de esos datos. Este proceso está descrito en la Fig.5 mediante un diagrama de flujo genérico que puede asociarse a cualquier entrada o salida del sistema.

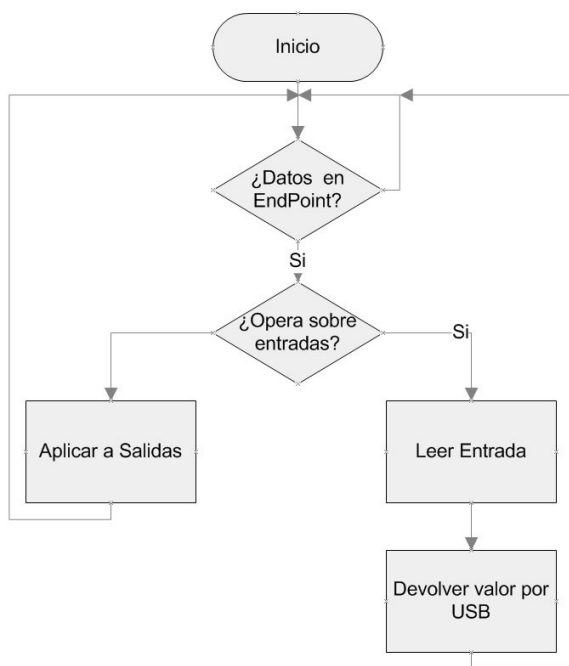


Figura N°5: Diagrama de Flujo transacciones USB

3.2 Lectura de sensores

Cuando una operación de “leer entrada” es llamada, el microcontrolador realiza la conversión analógica/digital en el caso de los potenciómetros ó el sensor de temperatura, devolviendo por el mismo endpoint que lo llamó el valor resultado de esta conversión. Para los sensores digitales (pulsadores) simplemente se devuelve el valor lógico leído en los terminales correspondientes del microcontrolador.

3.3 Escritura de Salidas

Cuando una operación del tipo “aplicar a salidas” es recibida por el microcontrolador, este chequeará cuál de las salidas deberá escribirse y con que valor.

Los 8 LEDs de salida pueden manejarse independientemente o bien en forma grupal, asociándolos a una variable del tipo Byte que represente la salida deseada.

Los relees solamente se manejan en forma individual.

4 Software para la PC

Como se mencionó en la introducción, uno de los objetivos es que los estudiantes escriban sus propios códigos para interactuar con la placa de adquisición de datos utilizando el lenguaje C. Para lograr esto se utilizan compiladores para este lenguaje que están disponibles para cualquier sistema operativo. En la FCEIA se utiliza Microsoft Windows y GNU/Linux con distribución Ubuntu, con lo cual es necesario utilizar drivers y/o librerías que sean soportadas por ellos e interactúen con el código fuente escrito en C.

Microchip ofrece los controladores necesarios para que Windows reconozca los dispositivos USB fabricados por esta empresa y puedan comunicarse con el sistema operativo. De todas formas también es posible escribir un driver propio para que se adapte a requerimientos específicos como ser nombre de clase, fabricante, etc.

En sistemas operativos que usen GNU/Linux el propio kernel ya tiene incorporado los controladores de Microchip y también es posible adaptarlos según los requerimientos ya nombrados.

4.1 Desarrollo de Librerías Especiales

Con el propósito de enfocar a los estudiantes solamente en la programación e interacción con la placa, se desarrollaron librerías de funciones que se encargan de ser un nexo entre el código escrito por ellos y el sistema operativo.

Las librerías desarrolladas traducen a un lenguaje más amigable para los estudiantes (tanto en el idioma como en las sintaxis de las propias funciones) otras librerías que manejan los puertos USB.

Para el caso de GNU/Linux se utiliza la librería libusb^[2] y para Windows hay dos posibilidades: libusb-win32 (adaptación de la anterior a Windows) para aplicaciones de consola o bien utilizar el servicio WinUSB^[3] para aplicaciones visuales. Según la forma de trabajo elegida se usará una u otra librería. En el Apéndice I están dichas librerías completas

Para ejemplificar las capacidades didácticas de la placa y las librerías se plantean a continuación dos ejemplos, uno con cada sistema operativo.

4.2 Ejemplo en Microsoft Windows

La FCEIA tiene un convenio con Microsoft, el cual permite a docentes y estudiantes obtener software como Visual Studio de forma gratuita. También existe la versión express de este paquete de desarrollo que es suficiente para desarrollar la aplicación aquí ejemplificada.

Haciendo uso de la suite de desarrollo Visual Studio 2008 y utilizando Visual C# como lenguaje de programación, se muestra en la Fig.6 la ventana de interacción con la placa USB en su modalidad visual.

En este caso se utilizó la librería para Visual C# **fceia_visual_c_api.cs** que hace uso del servicio de Windows WinUSB e importando desde allí las funciones de enumeración, escritura y lectura del protocolo USB.

Para interactuar con los LEDs puede enviarse un valor numérico de 0 a 255 o bien encender y apagar cada uno de ellos en forma independiente y accionar los relees utilizando los botones para tal fin.

Para la lectura de los potenciómetros deben habilitarse con las casillas de verificación correspondiente y la barra progresiva asociada a cada uno mostrará en forma porcentual la ubicación del índice del potenciómetro. También existe una casilla de verificación para habilitar la lectura del sensor de temperatura y otro para los pulsadores.

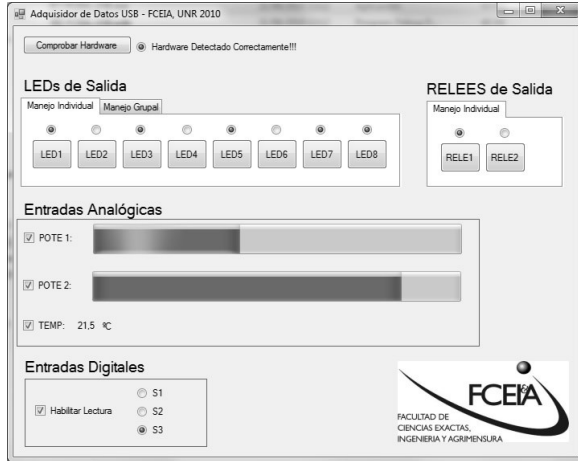
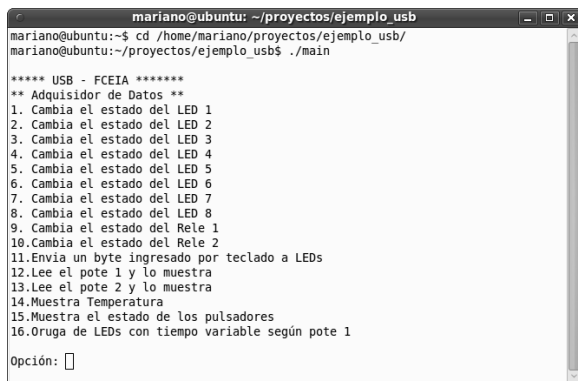


Figura N° 6: Ejemplo visual para Windows

4.3 Ejemplo en Ubuntu (GNU/Linux)

Ejemplificando una aplicación de consola, se utiliza la librería `api_fceia_usb.h` que hace uso de la librería `libusb` para este sistema operativo.

En la Fig.7 se muestra el menú principal de la aplicación y una porción de código que imprime en la pantalla el valor de uno de los potenciómetros.



```

if (teclado == '12'){
    while(1){
        printf("POTE 1= %d \n", lee_pot1(placa));
    }
}
    
```

Figura N° 7: Ejemplo de consola para GNU/Linux

En la porción de código se puede visualizar la forma en que están construidas las funciones de la librería desarrollada, siendo intuitivas para los estudiantes y facilitando que elaboren sus códigos fuentes interactuando con la placa a través del puerto USB de una forma sencilla y con sintaxis simple.

5 Trabajo práctico de aplicación

Para que los estudiantes hagan uso de la placa USB y el lenguaje C, se desarrolló un trabajo práctico que consiste en controlar el proceso de producción de galletitas para una empresa panificadora que dispone de una máquina de mezcla, amasado, corte, horneado y empaquetado continuo como la de la Fig. 8.

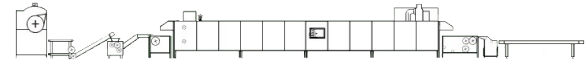


Figura N°8: Máquina continua a controlar

Se les propone a los estudiantes dividir el proceso en 7 etapas (mezcla, amasado, estirado, corte, cocción, enfriado y empaque) junto con la asignación de funciones de la Tabla 2 y un enunciado que especifica el funcionamiento del proceso de producción (tiempos de cada etapa, condiciones para pasar de una etapa a otra, valore mínimo y máximo de temperatura en el horno, etc) y además pide mostrar el valor de todas las variables medidas en pantalla.

Periférico (E/S)	Función Asociada
Pulsador S1	START: Comienzo de la producción
Pulsador S2	STOP: Fin de la producción
Pulsador S3	Parada de Emergencia (detiene el proceso y avisa con luz intermitente)
Potenciómetro 1	Peso sobre la cinta transportadora
Potenciómetro 2	Velocidad del ventilador de enfriado
Sensor Temperatura	Valor de la temperatura en el horno
Relé 1	Acciona el motor de la cinta transportadora
Relé 2	Acciona la válvula que enciende el horno
LEDs (1 a 7)	Indicación de la Etapa de producción
LED 8	Indicación luminosa para parada de emergencia

6 Conclusiones

El trabajo aquí expuesto está recientemente implementado en la FCEIA para los estudiantes de los primeros años de ingeniería eléctrica y electrónica. Desde el momento de su presentación en la clase se logra atraer la atención de los mismos con una aplicación concreta que va más allá de la resolución de situaciones problemáticas dentro de la computadora.

Esto permite que ellos aumenten su visión de la aplicación de la algoritmia y la programación mediante el manejo de un puerto actual, como lo es el USB, para el intercambio de información hacia y desde la computadora; logrando así que se interesen sobre posibles aplicaciones que les toque desarrollar en su vida profesional, tal como la que se plantea en el trabajo práctico.

También es importante destacar que muchos de los estudiantes demuestran un alto interés en el microcontrolador. La forma que este funciona, como se lo programa y en que lenguajes, etc.

Esto último aporta al conocimiento general sobre la carrera que estudian, las incumbencias de la misma y las posibilidades que esta les ofrece.

Apéndice I: Librerías Especiales

Ambas librerías, *api_fceia_visual_usb.cs* y *api_fceia_usb.h* están disponibles en forma de adjunta al presente informe. Además, en la siguiente web pueden encontrarse más detalles sobre la fabricación y utilización de la placa:

<http://toba.fceia.unr.edu.ar/mmendez>

Referencias

- [1] Axelson, J. USB Complete: The Developer's Guide. Third Edition (2005).
- [2] libusb project, <http://www.libusb.org/>
- [3] Microsoft MSDN, How to Access a USB Device by Using WinUSB Functions. <http://msdn.microsoft.com/en-us/library/ff540174.aspx>.
- [4] Posada-Gomez, R.; Enriquez-Rodriguez, J.J.; Alor-Hernandez, G.; Martinez-Sibaja, A.; USB Bulk Transfers between a PC and a PIC Microcontroller for Embedded Applications. IEEE, Electronics, Robotics and Automotive Mechanics Conference, 2008, pp. 559-564.