



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



Almacenamiento de datos de temperatura de motor BLDC para graficación y análisis en displays disponibles en tarjeta AVR BUTTERFLY y en tarjeta controladora LPCXPRESSO y presentación de mensajes de operación. (Noviembre 2012)

Pamela Puente⁽¹⁾, Antonio Delgado⁽²⁾, Carlos Valdivieso⁽³⁾
Facultad de Ingeniería en Electricidad y Computación⁽¹⁾⁽²⁾⁽³⁾
Escuela Superior Politécnica del Litoral (ESPOL)⁽¹⁾⁽²⁾⁽³⁾

Campus Gustavo Galindo, Km 30.5 Vía Perimetral, Apartado 09-01-5863. Guayaquil, Ecuador⁽¹⁾⁽²⁾⁽³⁾
pmpuente@espol.edu.ec⁽¹⁾, wadelgado@espol.edu.ec⁽²⁾, cvaldiv@fiee.espol.edu.ec⁽³⁾

Resumen

Este artículo tiene como objetivo mostrar una aplicación que hemos desarrollado basada en el uso de microcontroladores, esta será implementar un sensor de temperatura para un motor BLDC.

Este proyecto será capaz de mostrar la temperatura del motor en un display y al mismo tiempo almacenar los datos obtenidos para luego poder acceder a los últimos guardados, además se presentaran tres mensajes luminosos de acuerdo a su estado de operación.

Para la realización del proyecto se utilizan dos microcontroladores la tarjeta LPCXPRESSO 1769 de 32 bits y el Kit AVR Butterfly, que fueron programados con ayuda de dos herramientas de programación llamadas LPCXpresso 4 y AVR Studio 4.

Palabras claves: AVR Butterfly, LPCXPRESSO.

Abstract

This article will demonstrate an application that we have developed based on the use of two types of microcontrollers; it will implement a temperature sensor for a BLDC motor.

This project will be able to show the temperature of the motor in a display and at the same time it will store the data allowing us to view the last saved data, also will be noticed three light messages in accordance to the operation state.

This project uses two different microcontrollers, the LPCXpresso 1769 board and the 32-bit AVR Butterfly Kit, which were programmed using two programming tools called LPCXpresso 4 and AVR Studio 4.

Keywords: AVR Butterfly, LPCXpresso.

I. INTRODUCCION

Nuestro proyecto persigue hacer el análisis de temperatura de los motores bldc (brushless), que desde este momento en adelante solo lo mencionaremos como los motores bldc para referirnos a ellos, este análisis comienza con la medición de temperatura usando un transductor activo para monitoreo en tiempo real, seguido del almacenamiento de los datos obtenidos en memoria externa usando el microcontrolador AVR Butterfly y por ultimo mostrar mensajes de operación con otro microcontrolador de 32bits LPCXPRESSO que estará conectado mediante comunicación serial también llamada UART con la tarjeta microcontroladora AVR Butterfly.

Este proceso lo hemos dividido en cinco partes: Medición de temperatura del Motor BLDC, procesamiento digital de datos de temperatura, monitoreo en tiempo real (Graficación de datos), comunicación serial entre los 2 microcontroladores y presentación de mensajes de operación.

II. COMUNICACIÓN UART O USART

La comunicación serial es un protocolo muy común (no hay que confundirlo con el Bus Serial de Comunicación, o USB) para comunicación entre dispositivos que se incluye de manera estándar en prácticamente cualquier computadora. La mayoría de las computadoras incluyen dos puertos seriales RS-232. El concepto de comunicación serial es sencillo. El puerto serial envía y recibe bytes de información un bit a la vez. Aun y cuando esto es más lento que la comunicación en paralelo, que permite la transmisión de un byte completo por vez, este método de comunicación es más sencillo y puede alcanzar mayores distancias.

Típicamente, la comunicación serial se utiliza para transmitir datos en formato ASCII. Para realizar la comunicación se utilizan 3 líneas de transmisión: (1) Tierra (o referencia), (2) Transmitir, (3) Recibir. Debido a que la transmisión es asincrónica, es posible enviar datos por un línea mientras se reciben datos por otra. Existen otras líneas disponibles para realizar handshaking, o intercambio de pulsos de sincronización, pero no son requeridas. Las características más importantes de la comunicación serial son la velocidad de transmisión, los bits de datos, los bits de parada, y la paridad. Para que dos puertos se puedan comunicar, es necesario que las características sean iguales. [1]

III. TARJETA LPCXPRESSO 1769

El LPC1769 mostrado en la Figura 1 es un microcontrolador basado en la arquitectura ARM Cortex-M3 para aplicaciones embebidas que requieren un alto nivel de integración y baja disipación de potencia. El procesador ARM Cortex-M3 es un núcleo de nueva generación que ofrece mejoras en el sistema

de depuración, tales como modernas características de depuración y un mayor nivel de encapsulación de sus bloques.

Las versiones de alta velocidad (LPC1769 y LPC1759) operan a velocidades de hasta 120 MHz. Otras versiones pueden operar a 100 MHz. La ARMCortex-M3 de la CPU incorpora 3 etapas y utiliza una arquitectura de Harvard con instrucción local separada y buses de datos, así como un tercer autobús para los periféricos. La arquitectura ARMCortex-M3 de la CPU también incluye una unidad de procesamiento interno que soporta variantes.

El complemento periférico de la LPC1769 incluye hasta 512 KB de memoria flash, 64 kB de memoria de datos, Ethernet MAC, una interfaz USB que puede ser configurada como Host, dispositivo o OTG, 8 canales de uso general controlador DMA, 4 UARTs, 2 canales CAN, 2 controladores de SSP, interfaz SPI, 3 interfaces I2C, dos de entrada más de I2S 2-salida interfaz, 8 canales de ADC de 12 bits, 10 bits DAC, el motor de control PWM, codificador de cuadratura, 4 temporizadores de uso general, de 6 de salida de propósito general PWM, ultra bajo de energía RTC con suministro de la batería por separado, y hasta 70 pines de propósito general I/O. [2]

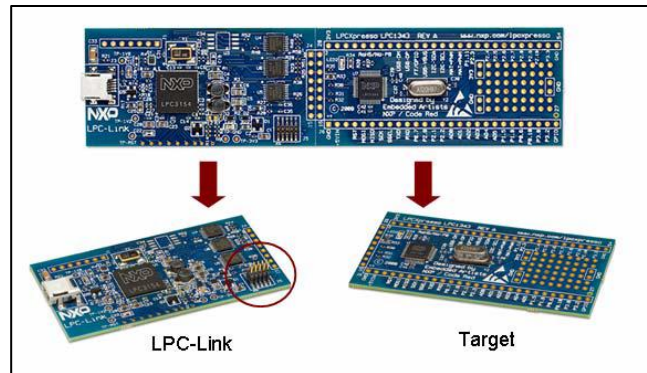


Figura 1 Tarjeta LPCXpresso 1769 [2]

A. CONEXIÓN PINES LPCXpresso 1769

Los pines a utilizar se muestran en la figura 2. son los que utilizaremos para el envío y recepción de datos en la tarjeta LPCXpresso, esta tarjeta posee 4 módulos para la comunicación UART en este caso utilizaremos el canal 1.

Pin	Type	Description
RXD1	Input	Serial Input. Serial receive data.
TXD1	Output	Serial Output. Serial transmit data.

UART1-TX / I2C1-SDA	P0.0	TXD3/SDA1	—P0[0]	← J6-9
UART1-RX / I2C1-SCL	P0.1	RXD3/SCL1	—P0[1]	← J6-10

Figura 2. Pines para comunicación UART

IV. TARJETA AVR BUTTERFLY

El Kit AVR Butterfly se diseñó para demostrar los beneficios y las características importantes de los microcontroladores ATMEL.

El AVR Butterfly utiliza el microcontrolador AVR ATmega169, que combina la Tecnología Flash con el más avanzado y versátil microcontrolador de 8 bits disponible. En la Figura 3 se puede apreciar el Kit AVR Butterfly. [3]

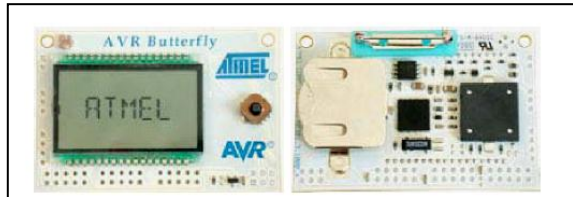


Figura 3. Kit AVR Butterfly [3]

CARACTERÍSTICAS

El Kit AVR Butterfly expone las siguientes características principales:

La arquitectura AVR en general y la ATmega169 en particular.

- Diseño de bajo consumo de energía.
- Periféricos:
 - Controlador LCD.
 - Memorias:
 - ✓ Flash, EEPROM, SRAM.
 - ✓ Data Flash externa.
- Interfaces de comunicación:
 - UART, SPI, USI.
- Métodos de programación
 - Self-Programming/Bootloader, SPI, Paralelo, JTAG.
- Convertidor Analógico Digital (ADC).
- Timers/Counters:
 - Contador de Tiempo Real (RTC).
- Modulación de Ancho de Pulso (PWM).

El AVR Butterfly está proyectado para el desarrollo de aplicaciones con el ATmega169 y además puede usarse como un módulo dentro de otros productos.

B. CONECTORES DEL AVR BUTTERFLY

Algunos de los pines de I/O del micro-controlador ATmega169 están disponibles en los conectores del AVR Butterfly. Estos conectores son para comunicación, programación y entrada al ADC del ATmega169. En la Fig 4. se puede apreciar los conectores del AVR Butterfly.

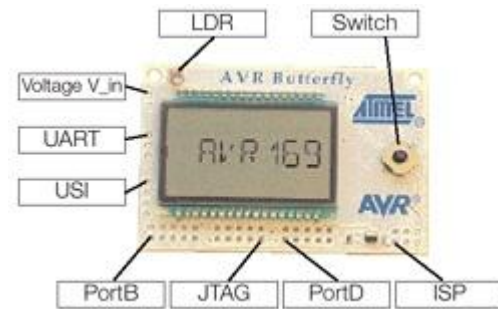


Fig. 4. Conectores del AVR Butterfly

B. DISTRIBUCION DE PINES PARA LA COMUNICACIÓN SERIAL ENTRE EL AVR Y LA TARJETA LPCXPRESSO.

La comunicación UART requiere de tres líneas: TXD, RXD y GND. TXD es la línea para transmitir datos hacia la tarjeta LPCXPRESSO, RXD es la línea para recepción de datos y GND es la tierra común. En la Tabla 1, se observa la distribución de los pines para la comunicación serial, a la izquierda los pines del AVR.

TABLA. 1.

AVR BUTTERFLY UART
Pin 1 (RXD)
Pin 2 (TXD)
Pin 3 (GND)

Distribución de pines, AVR Butterfly

En la Fig. 5 se observa cómo se debe hacer el cableado para la comunicación, a través de la interfaz serial RS-232, entre el AVR Butterfly y la LPCXpresso.

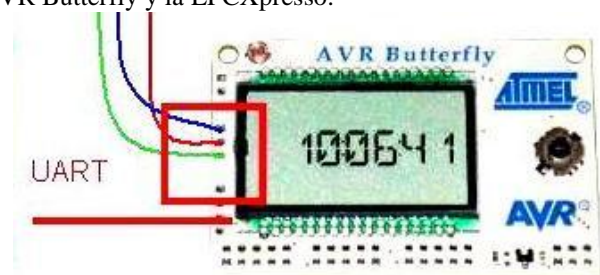


Fig. 5. Conexiones para interfaz UART del AVR Butterfly

V. TRANSDUCTOR LM35

En nuestro proyecto usaremos el transductor LM35 que nos ayudara a medir la temperatura del motor y por hacer el análisis en base de los datos adquiridos, ya que este transductor es del tipo activo, es decir necesita polarización y tiene una salida de voltaje lineal con respecto a la temperatura y escalado en grados Celsius, lo que nos da mas facilidad en la

medición y simplicidad en la adquisición de estos datos medidos.

El LM35 es un sensor de temperatura con una precisión calibrada de 1°C. Puede medir temperaturas en el rango que abarca desde -55° a + 150°C. La salida es muy lineal y cada grado centígrado equivale a 10 mV en la salida.

Sus características más relevantes son:

- Precisión de ~1,5°C (peor caso), 0.5°C garantizados a 25°C.
- No linealidad de ~0,5°C (peor caso).
- Baja corriente de alimentación (60uA).
- Amplio rango de funcionamiento (desde -55° a + 150°C).
- Bajo costo.
- Baja impedancia de salida.

Su tensión de salida es proporcional a la temperatura, en la escala Celsius. No necesita calibración externa y es de bajo costo. Funciona en el rango de alimentación comprendido entre 4 y 30 voltios.

Como ventaja adicional, el LM35 no requiere de circuitos adicionales para su calibración externa cuando se desea obtener una precisión del orden de ± 0.25 °C a temperatura ambiente, y ± 0.75 °C en un rango de temperatura desde 55 a 150 °C.

La baja impedancia de salida, su salida lineal y su precisa calibración inherente hace posible una fácil instalación en un circuito de control.

Debido a su baja corriente de alimentación (60uA), se produce un efecto de autocalentamiento reducido, menos de 0.1 °C en situación de aire estacionario. [4]

A. CONEXIÓN PINES LM35

La Figura 6 nos muestra la configuración para los pines del LM35

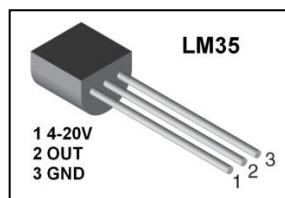


Figura 6. Pines LM35

VI. HERRAMIENTAS DE PROGRAMACION

LPCXpresso v 4.1.5

El LPCXpresso es un toolchain completo para evaluación y desarrollo con microcontroladores de NXP.

Está compuesto por:

- LPCXpresso IDE y "development tools"
- IDE basado en Eclipse
- compiler y linker GNU
- GDB debugger
- LPCXpresso target board (stick)
- BaseBoard o hardware adicional(opcional)

Eclipse utiliza algunos conceptos que no siempre son comunes a otros entornos de desarrollo por lo que vamos a ver algunos de ellos.

- Workspace: Es el contenedor de nuestros proyectos. Estos proyectos pueden ser aplicaciones y/o bibliotecas. También almacena todas las configuraciones del entorno por lo que se puede mover muy fácilmente de computadora en computadora.
- Proyecto: Este puede ser de dos tipos. Biblioteca estática o una aplicación ejecutable. Contiene archivos de código fuente (.c), encabezados (.h) y cualquier otro archivo que se desee.

En general utilizaremos el workspace para intercambiar proyectos (en el sentido convencional de la palabra) ya que el mismo incluirá todas las bibliotecas necesarias.

Los proyectos pueden ser de dos tipos:

- Aplicaciones: Se compilan y se pueden descargar directamente al target.
- Bibliotecas estáticas: Se pueden compilar, pero para usarlas, un proyecto de tipo aplicación debe hacer llamadas a las funciones que este contiene. Es decir, no puede tener un main(). Este tipo de proyectos no se puede descargar por si solo al microcontrolador.

A continuación en la Figura 7 se muestra el IDE que utilizaremos para la programación de nuestra LPCXpresso.

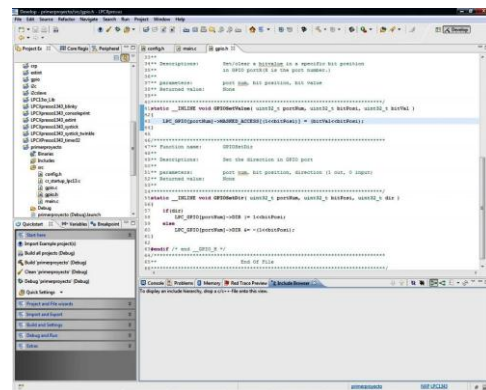


Figura 7. Ambiente programación LPCXpresso

AVR STUDIO 4

AVR Studio es un Entorno de Desarrollo Integrado (IDE) para escribir y depurar aplicaciones AVR en el entorno de Windows 9x/Me/NT/2000/XP/7. Ver Fig.8.

AVR Studio 4 soporta varias de las fases por las cuales se atraviesa al crear un nuevo producto basado en un microcontrolador AVR.

Éste tiene una arquitectura modular completamente nueva, que incluso permite interactuar con software de otros fabricantes. AVR Studio 4 proporciona herramientas para la administración de proyectos, edición de archivo fuente, simulación del chip e interfaz para emulación In-circuit para la poderosa familia RISC de microcontroladores AVR de 8 bits. [5]

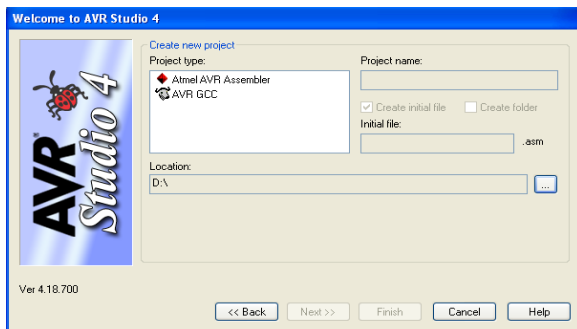


Fig.8. Entorno de AVR STUDIO 4

AVR Studio 4 consiste de muchas ventanas y sub-módulos. Cada ventana apoya a las partes del trabajo que se intenta emprender. En la Figura 9 se puede apreciar las ventanas principales del IDE.

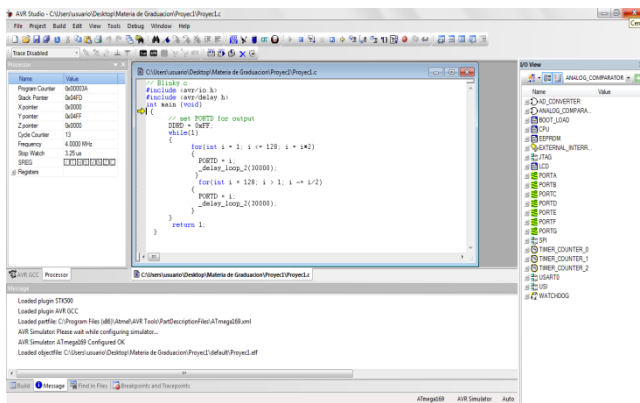


Fig. 9. Ventana Principal IDE

WINAVR

La distribución WinAvr, que es una recopilación de programas de software libre diseñados para facilitar las tareas de programación y desarrollo de los microcontroladores Avr. Dicha distribución WinAvr incorpora además del compilador gcc de consola, un editor de texto especialmente diseñado para ayudar al programador y hacer el código más legible mediante su resaltado con colores.

El programador vía puerto serie mprog.exe, que permite transferir el programa compilado, que se encuentra en un archivo llamado main.hex, a la memoria flash del microcontrolador utilizando únicamente un cable de tres líneas.

En pocas palabras WinAVR es un conjunto de herramientas de desarrollo para micro-controladores RISC AVR de Atmel, basado en software de código abierto y compilado para funcionar en la plataforma Microsoft Windows. WinAVR incluye las siguientes herramientas:

- Avr-gcc, el compilador de línea de comandos para C y C++.
- Avr-libc, la librería del compilador que es indispensable para avr-gcc.
- Avr-as, el ensamblador.
- Avrdude, la interfaz para programación.
- Avarice, la interfaz para JTAG ICE.
- Avr-gdb, el depurador.
- Programmers Notepad, el editor.
- MFile, generador de archivo makefile.

VII. PROYECTO TERMINADO

Este proyecto consiste en medir la temperatura de un motor, mostrar estos datos en la AVR Butterfly además de permitir almacenarlos y poder acceder a ellos, luego enviar datos mediante comunicación serial a la tarjeta LPCXpresso y mostrar mensajes de operación en base a la temperatura medida, La figura 10 muestra el funcionamiento de todo el proyecto

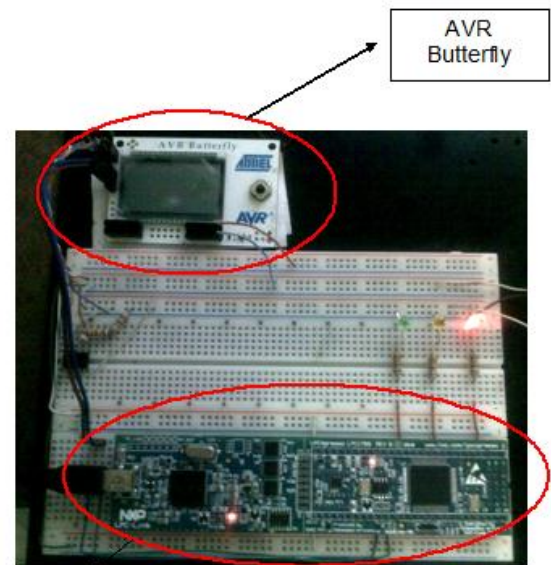
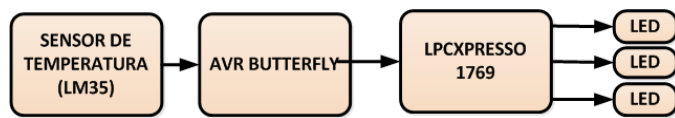


Figura 4.11 Proyecto funcionando

Figura 10. Proyecto funcionando

DIAGRAMA DE BLOQUES



FUNCIONAMIENTO

Podemos observar su funcionamiento en la figura 11.

En la tarjeta AVR Butterfly:

- Si la Temperatura 20°C se envía la letra "B"
- Si la Temperatura >math>19^{\circ}\text{C}</math> y 41°C se envía la letra "G"
- Si la Temperatura >math>40^{\circ}\text{C}</math> se envía la letra "R"

En la tarjeta LPCXPRESSO:

- Si recibe la la letra "B" Enciende LED VERDE
- Si recibe la la letra "G" Enciende LED AMARILLO
- Si recibe la la letra "R" Enciende LED ROJO

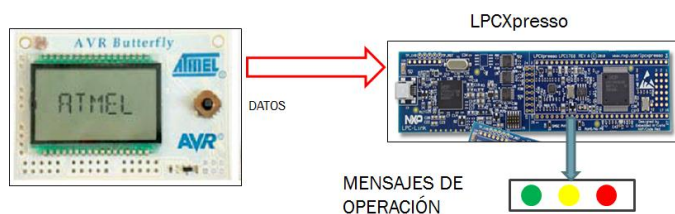


Figura 11. Funcionamiento

VIII. CONCLUSIONES

1. En base al proyecto desarrollado podemos concluir que esta tesis es la integración de diferentes tecnologías para el aprovechamiento de recursos y lograr un mismo fin ya que en este proyecto usamos 2 tipos de Microcontroladores diferentes, uno de 32 bits y el otro de 8 bits de diferentes marcas, Atmel y LPCxpreso, creados para diferentes aplicaciones sin embargo con la capacidad de trabajar juntos como lo demostramos en este proyecto.
2. Una de las más importantes acotaciones para la integración de diferentes Microcontroladores es la comunicación entre estos ya que sin importar el tipo de microcontrolador la comunicación entre ellas tienen un estándar y este tipo de comunicación para nuestro caso es la llamada serial RS232, es el enlace que nos permite comunicarnos por eso la importancia de conocer bien el estándar de comunicación a usar para poder configurar ambas partes y realizar una comunicación con éxito.

3. Al medir temperatura es necesario tener en cuenta varios factores como por ejemplo saber que rangos de temperatura se medirá, que tipo de resolución se desea en la medición y qué tipo de sensor se usará ya que dependiendo del tipo de sensor se debe diseñar el esquema de la implementación, en algunos sensores es necesario un circuito adicional de amplificación, compensación e incluso linealización.
4. La lectura de parámetros analógicos mediante el módulo ADC es una herramienta poderosa y muy versátil, tenerla en el AVR butterfly fue muy ventajoso para nuestro proyecto sin embargo es necesario también hacer ajustes a la lectura obtenida para efectos de escalabilidad es decir poner la lectura en el rango numérico que usaremos para procesar la información y para efectos de calibración.

IX. RECOMENDACIONES

1. Es importante entender primero el funcionamiento individual de cada uno de los microcontroladores que involucra este proyecto antes de combinar sus funciones ya que cada uno de ellos posee su propia estructura y lenguaje.
2. Revisar minuciosamente la sintaxis y tener en cuenta que todas las librerías estén incluidas antes de compilar el código.
3. Documentar correctamente el código para un mejor entendimiento del mismo.
4. Es importante mencionar que en la configuración de la comunicación UART, siempre se tiene que asegurar que tengan la misma velocidad de transferencia tanto en el AvrButterfly como en la tarjeta LPCXPRESSO, ya que si no se tiene esto no existirá comunicación entre ambos dispositivos.

X. REFERENCIAS

[1] Universidad de Catarina. Comunicación serial. Disponible en:

http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/morales_h_oe/capitulo3.pdf. (Consultado el 22 de Marzo del 2012)

[2] NXP Semiconductors. LPCXPRESSO Microcontrollers.

Disponible en: <http://ics.nxp.com/lpcxpreso/~LPC1769/>. (Consultado el 19 de Marzo del 2012).

[3] ATMEL. Avr Butterfly. Disponible en:

<http://www.atmel.com/Images/doc4271.pdf> (Consultado el 22 de Marzo del 2012).

[4] Hazael, I. (s.f.). Monografias.com. Transductores y Sensores. Disponible en:

<http://www.monografias.com/trabajos31/transductores-sensores/transductores-sensores.shtml> (Consultado el 23 de Febrero del 2012).

[5]ATMEL. AVR Studio 4. Disponible en:

<http://www.elec.uow.edu.au/avr/guides/Getting-started-C-programming-Atmel-AVR.pdf>(Consultado el 22 de Marzo del 2012).