



ESCUELA SUPERIOR POLITECNICA DEL LITORAL
CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLOGÍA



Control mediante joystick de tarjeta AVR Butterfly (con microcontrolador ATmega169) mediante comunicación SPI con tarjeta LPCXpresso controladora de motor BLDC y presentación en display de mensajes de operación. (Noviembre 2012)

Lidia Mantilla ⁽¹⁾, Juan Maroto ⁽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⁾
lidimant@espol.edu.ec ⁽¹⁾, amaroto@espol.edu.ec ⁽²⁾, cvaldiv@fiee.espol.edu.ec ⁽³⁾

Resumen

Este proyecto permite el desarrollo e implementación de un controlador de motor BLDC mediante un joystick incorporado en la tarjeta AVR Butterfly y la presentación del control de motor por medio de un display de mensajes, usando el protocolo de comunicación SPI - Serial Peripheral Interface, el mismo que es programado en la tarjeta LPCXpresso1769. En este trabajo se aborda de forma general conceptos teóricos sobre el funcionamiento del protocolo SPI y su aplicación en el microcontrolador LPCXpresso por medio del cual realizaremos la comunicación entre dispositivos; se analizan herramientas de hardware, conceptos, funcionamiento y uso de motores BLDC. Se explica el diseño del proyecto, funcionamiento del código, diagrama de bloques, diagrama de flujo, algoritmo de la parte del control de motores y las herramientas de software utilizadas: LPCXpresso y AVR STUDIO4 que permiten programación en lenguaje C. En la última parte se muestran los diagramas esquemáticos de los ejercicios para llevar a cabo el proyecto y la lista de elementos utilizados para su implementación.

Palabras Claves: AVR Butterfly, LPCXpresso, motor BLDC, comunicación SPI.

Abstract

This project allows the development and implementation of a BLDC engine driver through a joystick built into the AVR Butterfly motherboard and through a messages display to control the engine, using the SPI(Serial Peripheral Interface) communication protocol, which is set in the LPCXpresso1769 card. This paper presents general theoretical concepts of SPI protocol functionality, its application in the LPCXpresso microcontroller, through which we will enable communication between devices; it is analyzed hardware tool, concepts, functioning and use of BLDC engines. It is explained the project design, code functioning, block diagrams, control engine algorithms and used of software tools: LPCXpresso and AVR STUDIO4, which allows C language programming. In the last part it is shown the schematic diagrams of the required steps and the elements to implement the project.

Keywords: SPI, AVR Butterfly, LPCXpresso, BLDC motor, SPI communication.

1. Introducción.

El uso de los microcontroladores se ha incrementado para controlar diferentes dispositivos como motores BLDC, juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems, el sistema de arranque de nuestro carro entre otros. Por eso este proyecto tiene como objetivo establecer comunicación SPI entre tarjetas

LPCXpresso1769 entre maestro y esclavo, mostrar la programación del microcontrolador LPCXpresso1769 y el AVR Butterfly, implementar el control de motores mediante el uso del joystick integrado al AVR y, por último, mostrar los resultados del control del motor BLDC por medio de un display, para entender el funcionamiento del protocolo SPI, haciendo uso de varias herramientas que incluyen el software del LPCXpresso y el diseño en prototipo del proyecto.

En este proyecto se usa la comunicación en serie ya que realiza el envío de datos de un bit a la vez sobre un canal de comunicación y es utilizada en casi todas las comunicaciones y redes de computadoras, por costo de cables y la programación es más sencilla de realizar e interpretar al momento de implementar.

1.1. Antecedentes.

En este proyecto usamos un motor sin escobillas para realizar una aplicación de comunicación SPI, este tipo de motores son eléctricos y mucho más eficientes casi no requieren mantenimiento; son utilizados desde hace años en la industria en general, su ventaja es que pueden durar muchos años y también se han venido utilizando en los aviones y barcos RC.

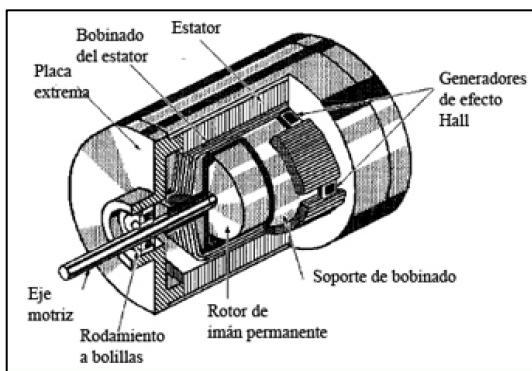


Figura 1.1. Vista en corte de un motor DC sin escobillas. [2]

1.2. Identificación del Problema.

En este proyecto se realiza la sincronización de programación y compatibilidad entre dispositivos.

Entre los factores de dificultad tenemos la comunicación por medio del protocolo SPI entre maestro y esclavo de las tarjetas LPCXpresso, este tipo de comunicación requiere sincronización y el uso adecuado de ciertas librerías que vienen en el software del fabricante; seguido de la comunicación entre la tarjeta LPCXpresso y el AVR Butterfly debido a que son dispositivos que trabajan con diferente software de programación por pertenecer a diferentes fabricantes.

Con el fin de desarrollar un sistema de control y manejo de motores BLDC, se realiza la interacción entre estos microcontroladores para aprovechar las ventajas y utilidades, demostrando que el trabajo integrado de los mismos es factible, aunque tengan diferentes características de fábrica y se manejen con sistemas de software distintos.

2. Fundamentos Teórico.

En este capítulo realizamos un análisis profundo sobre el funcionamiento y características de elementos como: protocolo de comunicación serial SPI, motores

BLDC, programación en el microcontrolador LPC1769 y el microcontrolador ATMEGA169, el cual se programa mediante el software AVR Studio 4.

2.1. Interfaz Periférica Serial – SPI.

SPI - Serial Peripheral Interface - es un subsistema de comunicación serial independiente, un estándar establecido por Motorola que emplea un bus de 4 líneas para interconectar dispositivos periféricos de baja y media velocidad, e incluso microcontroladores y dispositivos como los AVR. La comunicación se realiza basada en la interacción maestro – esclavo, donde el maestro llama al esclavo para establecer el proceso de transmisión y recepción de la información.

La comunicación SPI trabaja de manera full dúplex, esto quiere decir que puede enviar y recibir información de manera simultánea, elevando la transferencia de los datos. Con esta interfaz se obtiene un dispositivo maestro que inicia y controla la comunicación, y uno o más esclavos que reciben y transmiten hacia el maestro.

El maestro es aquel que inicia la transferencia de datos sobre el bus y genera las señales de reloj (SCK) y control (MISO, MOSI). Un esclavo es un dispositivo el cual es controlado por el maestro. Cada esclavo es controlado sobre el bus a través de una línea selectora llamada Chip Select o Select Slave (SS), esto quiere decir que el esclavo es activado solamente cuando esta línea es seleccionada. [3]

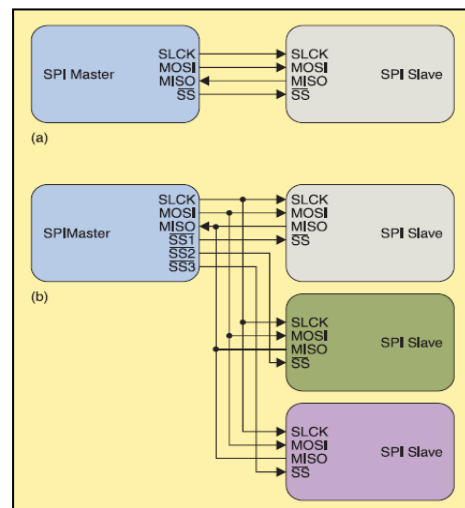


Figura 2.1. a) Conexión entre un dispositivo maestro y un esclavo. b) Conexión entre un dispositivo maestro y varios esclavos. [3]

2.2. Características SPI.

SPI puede transmitir información desde un periférico maestro hacia uno o varios esclavos, envía y recibe simultáneamente por el mismo canal, la

transferencia de datos puede ser LSB o MSB, la implementación en hardware es extremadamente simple.

No hay control de flujo por hardware, ni señal de asentimiento, puede ocurrir que el servidor envíe información sin estar conectado ningún cliente y no se daría cuenta de nada, no permite tener varios servidores conectados al bus y sólo funciona en distancias cortas.

2.3. Herramientas de Software.

En este proyecto usamos tres tipos de programas: AVR Studio 4 para la programación del microcontrolador ATmega169, LPCXpresso 4 para realizar la programación de la tarjeta LPC1769 y PROTEUS 7.7 Service Pack 2 para la simulación del proyecto.

2.4. Herramientas de Hardware.

Usamos dos herramientas de hardware para la implementación de este proyecto: la tarjeta AVR Butterfly de ATmega y la tarjeta LPC1769 de LPCXpresso.

2.4.1. AVR Butterfly

La tarjeta AVR Butterfly es de tamaño reducido y tiene una gran capacidad de procesamiento de órdenes e información consumiendo poca energía; puede trabajar como Convertidor Analógico – Digital (ADC), contador en tiempo real (RTC) y como herramienta de modulación de ancho de pulso (PWM). Está compuesta por el microcontrolador ATmega169 y dispositivos periféricos como el joystick de control y puertos de comunicación con dispositivos externos. Entre las características tenemos que el encapsulado con el que trabaja es de tipo MLF; como periféricos utiliza el controlador de LDC, memorias flash, EEPROM, SRAM y memoria DataFlash externa.

El AVR Butterfly puede trabajar haciendo uso de protocolos de comunicación como SPI, UART e I2C. Esta tarjeta utiliza métodos de programación como Self-Programming/Bootloader, ISP, paralelo y JTAG.



Figura 2.2.: Vista de doble cara de la tarjeta AVR Butterfly, desarrollada por el fabricante ATmega.

En la figura 2.3 se muestra un esquema simplificado de los periféricos con los que cuenta la tarjeta AVR Butterfly y los componentes que conforman el microcontrolador ATmega169, para conseguir la comunicación entre el microcontrolador y un PC con un puerto RS-232 es necesario configurar la UART y el reloj interno del sistema para conseguir la tasa de transferencia deseada para la comunicación, la cual no podría llevarse a cabo sin la circuitería elevadora de tensión de 3V a 12V con la que dispone la tarjeta AVR Butterfly.

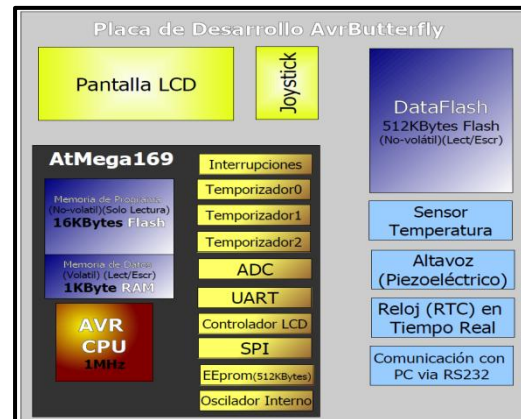


Figura 2.3.: Esquema simplificado de los periféricos internos y externos que conforman la tarjeta AVR Butterfly [15]

Es necesario realizar la comunicación entre el microcontrolador ATmega169 a través del periférico SPI para almacenar y restaurar datos del dispositivo de memoria DataFlash, y habilitar el periférico gestor de interrupciones para poder darle funcionalidad a la interfaz de entrada con el usuario (joystick), es necesario habilitar el periférico gestor de interrupciones, para que cada vez que se toque un botón se acceda a una rutina de interrupción asociada que la resuelva satisfactoriamente, esta aplicación nos será de gran utilidad para el manejo del motor BLDC.

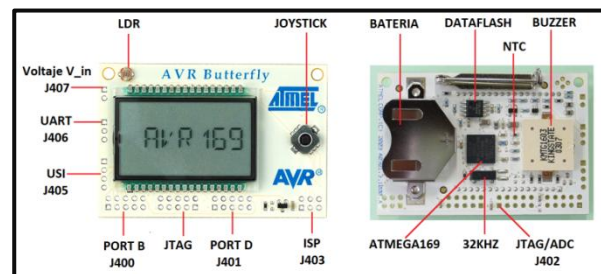


Figura 2.4.: Distribución de los periféricos externos en la tarjeta AVR Butterfly.

Entre los periféricos más importantes podemos destacar el NTC que es la resistencia negativa de temperatura o sensor de temperatura de la tarjeta, el

LDR que es la resistencia dependiente de la intensidad de la luz, el Joystick de cinco posiciones de las cuales utilizaremos cuatro para manejo del motor BLDC, el oscilador de cuarzo de 32KHz para la generación de la onda del reloj en tiempo real, el buzzer (piezoeléctrico) o altavoz para la emisión de sonidos de un solo tono, la memoria DataFlash de 512Kbytes de capacidad de almacenamiento y el periférico de comunicación UART, cuyos pines de salida están directamente conectados al puerto RS232 del ordenador.

A. Capacidad de Almacenamiento.

La tarjeta AVR Butterfly cuenta con dos circuitos integrados con capacidad de almacenamiento de datos. Estos dispositivos son: el microcontrolador ATmega169 y la memoria DataFlash.

El microcontrolador ATmega169 cuenta, a su vez, con tres memorias integradas en su encapsulado: Una memoria flash auto-programable de 16Kbytes, una memoria SRAM de 1Kbyte y una memoria EEPROM de 512Kbytes.

Adicionalmente podemos decir que, por motivos de seguridad, el espacio de memoria del programa se divide en dos secciones, la de programa de arranque (boot) y la de programa de aplicación. Además de este tipo de memorias internas, la tarjeta AVR Butterfly cuenta con un periférico de almacenamiento que es una memoria DataFlash de 512Kbytes de capacidad (4Mbits).

B. Características

Algunas características se describen a continuación:

- Microcontrolador ATmega169 con encapsulado tipo MLF.
- Pantalla LCD de 120 segmentos que se usa en conjunto con el joystick de 5 direcciones para mostrar las capacidades de control del microcontrolador ATmega169.
- Altavoz piezoeléctrico utilizado para reproducir sonidos.
- Cristal de cuarzo de 32KHz para el RTC. **2.7.1.1.**
- Memoria DataFlash de 4 Mb para almacenar datos.
- Convertidor de nivel RS-232 para establecer comunicación entre unidades fuera del kit.
- Periférico NTC o termistor de coeficiente de temperatura que nos ayuda a medir y sensar temperatura.
- Resistencia dependiente de la luz o LDR que nos permite sensar y medir intensidad luminosa.
- Emulación del JTAG para la depuración.
- Una interfaz USI que nos sirve como interfaz adicional de comunicación.
- Terminales externas para conectores tipo Header.

- Batería de 3V que proporciona energía para el funcionamiento de la tarjeta AVR Butterfly.
- Bootloader usado para la programación mediante la PC y compatibilidad con la herramienta de software AVR Studio 4.

2.4.2. Tarjeta LPCXpresso

El LPCXpresso es un toolchain completo que nos permite evaluar y desarrollar sobre microcontroladores de NXP, como por ejemplo el LPC1769 utilizado en este trabajo.

Está compuesto por:

- IDE LPCXpresso y herramientas de desarrollo.
- Un compilador y linker GNU.
- Debugger GDB.
- Una target board de LPCXpresso, la cual trabaja como una BaseBoard o hardware adicional del sistema.

El target board es un microcontrolador con todo lo necesario para encendido del sistema y también cuenta con una herramienta que tiene integrados un programador y un debugger. En éste se encuentra ensamblado el microcontrolador LPC1769, el cual es un sistema ARM Cortex-M3, con una memoria Flash de 512Kb y 64Kb de memoria SRAM, puertos Ethernet y USB y que trabaja en modo On the Go.

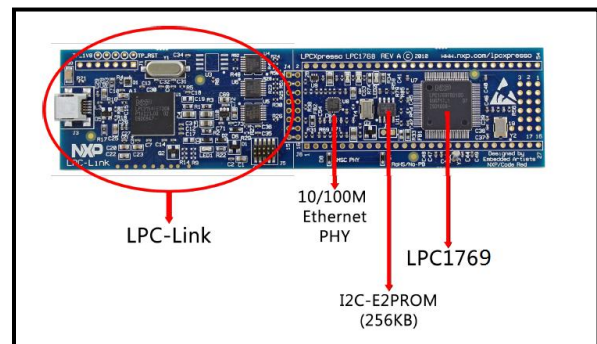


Figura 2.5.: Componentes básicos del kit de la tarjeta LPCXpresso [18]

A. Periféricos de la tarjeta LPCXpresso

Entre los periféricos generales podemos mencionar:

- Socket para LPCXpresso.
- Módulo MBED.
- 50 pines de expansión dual
- Interfaz USB y encendido de batería.

Entre los periféricos digitales encontramos:

- RGB-LED que puede ser controlado por PWM.

- 2 pulsadores uno usado en la activación del gestor de arranque y un interruptor giratorio con la codificación de cuadratura.
- Sensor de temperatura con salida PWM.

Entre los periféricos analógicos podemos mencionar:

- 1 Potenciómetro de ajuste de entrada analógica.
- 1 PWM de entrada y salida analógica
- 1 Salida PWM de los altavoces.

Entre los periféricos serie y UART podemos encontrar:

- 1 Puerto USB con cambio a activación automática ISP.
- 1 Interfaz RS232
- 1 Socket para módulo RF.

B. Características de la tarjeta LPCXpresso

Características principales que ofrece la tarjeta LPCXpresso con el microcontrolador integrado LPC1769:

- Pertenece a ARM Cortex-M3, integrado en el controlador de interrupciones anidadas vectoriales.
- Funciona a frecuencias de hasta 100MHz
- Memoria flash hasta 512KB.
- Chip SRAM que incluye 16 o 32KB de memoria SRAM con un bus de datos de alto rendimiento, pueden ser usados para Ethernet, USB y memoria de DMA.
- Emplea 8 canales de uso general DMA (GPDMA) en el AHB multicapa que se puede utilizar con protocolos de comunicación SSP, I2C, bus, UART, ADC y para transferencias de memoria a memoria.
- Matriz de conexión para cada maestro del AHB, estos pueden ser utilizados para uso general del controlador DMA, MAC Ethernet y la interfaz USB.
- Controlador dedicado DMA, tiene un USB 2.0 de alta velocidad, 4 entradas UART para la generación de fracción de velocidad de transmisión, FIFO interna y DMA. Una UART tiene el control de las E/S del módem.
- Posee controlador CAN 2.0B para manejo de dos canales, control sincrónico SPI, comunicación serial full dúplex y una longitud de datos programable, además de poseer dos controladores SSP con capacidades de FIFO y multiprotocolo.

Entre otros periféricos que podemos mencionar se encuentran de 70 a 100 paquetes de pines de uso general de E/S del GPIO con pines configurables para resistencias pull-up y pull-down. Al bloque de GPIO se accede a través del bus AHB multicapa para un acceso rápido y situado en la memoria de tal manera

que soporta equipos bajo el estándar Cortex-M3. Incluye 12 bits para ADC - Convertidor de Analógico Digital y 10 bits para DAC - Convertidor Digital Analógico. Los primeros ayudan con el aporte de multiplexación entre los ocho pines, las tasas de conversión son de hasta 200KHz; y los segundos incluyen un temporizador de conversión específico. Posee cuatro temporizadores para fines generales y contadores, con un total de ocho entradas de captura, cada bloque tiene un temporizador de entrada externa. También se permite el control de motor por PWM con soporte para control de motores de tres fases.

3. Proyecto: Control de motor BLDC por medio de joystick utilizando comunicación SPI de AVR a LPC.

3.1. Descripción

El proyecto consiste en realizar el control del motor BLDC por medio del joystick incorporado en la tarjeta AVR. La transmisión mediante comunicación serial SPI se va a realizar desde una tarjeta AVR Butterfly, la cual está configurada como maestro, hacia una tarjeta LPCXpresso, la cual se encuentra configurada como esclavo, y a esta última se encuentra conectado el motor BLDC, el cual recibirá las órdenes de la tarjeta maestro. El control del joystick ha sido configurado de la siguiente manera: hacia arriba se incrementa la velocidad del motor, hacia abajo se disminuye la velocidad del motor, hacia la derecha se invierte el sentido de giro del motor y hacia la izquierda o presionando el centro se realiza el encendido/apagado del motor. Para efectos de visualización, estas órdenes generadas por medio del joystick se irán presentando en la pantalla LCD incorporada en la tarjeta AVR Butterfly.

3.2. Diagrama de bloques

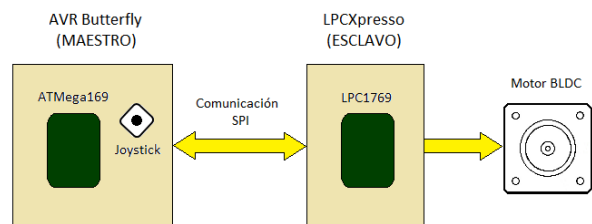


Figura 3.1: Diagrama de Bloques del Control de motor BLDC por medio del joystick utilizando comunicación SPI de AVR a LPC.

El AVR BUTTERFLY trabaja como maestro de la comunicación SPI, para nuestro controlador mediante joystick del motor BLDC. El LPC1769 trabaja como esclavo de la comunicación, en donde los pines del puerto dos son configurados como salidas y todos en estado alto.

El joystick utilizado del AVR Butterfly posee cinco movimientos que son: arriba, abajo, izquierda, derecha y centro. Para nuestro proyecto se configuró mediante programación, el envío de las 4 órdenes cuyo reconocimiento se realiza en un estado bajo durante un periodo de tiempo en la LPC1769.

Tabla 3.1 Asignación de pines para el funcionamiento del motor BLDC.

JOYSTICK (Movimiento)	ORDENES	LPC1769	TARJETA GRANDE (LPCXpresso o Motor Control Board)
Arriba ≡ UP	Aumentar Velocidad	Port2 [3]	PIO3_3
Abajo ≡ DOWN	Disminuir Velocidad	Port2 [2]	PIO3_2
Derecha ≡ RIGHT	Invertir Giro	Port2 [1]	PIO3_1
Izquierda/centro ≡ LEFT/CENTRO	ON/OFF	Port2 [0]	PIO3_0

3.3. Lista de Materiales:

- AVR Butterfly.
- LPC 1769.
- Cuatro resistencias de 1K Ω y cuatro resistencias de 330 Ω .
- Cuatro LEDs.
- Una batería de 3.6V

3.4. Imágenes del funcionamiento.

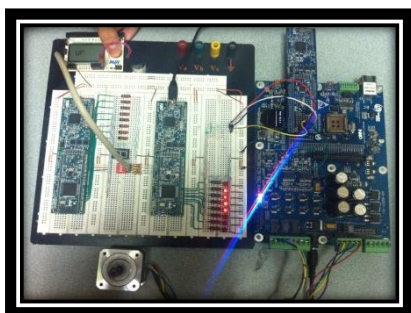


Figura 3.2.: Movimiento del joystick hacia ARRIBA

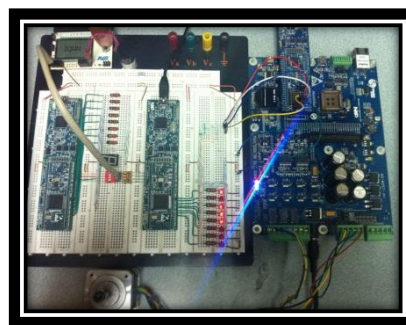


Figura 3.3.: Movimiento del joystick hacia ABAJO.

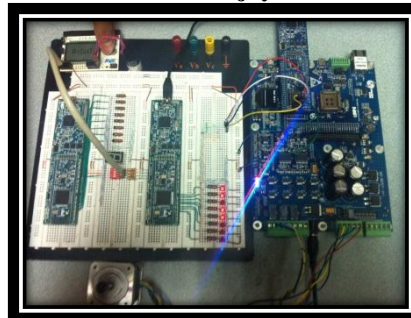


Figura 3.4.: Movimiento del joystick hacia la DERECHA.

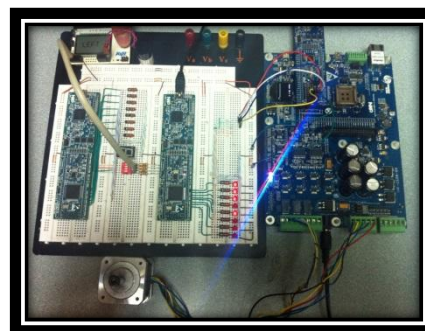


Figura 3.5.: Movimiento del joystick hacia la IZQUIERDA.

4. Conclusiones.

1. Se utilizó el protocolo de comunicación serial sincrónica SPI para transmisión – recepción mediante la asignación maestro – esclavo, lo cual nos permitió la comunicación entre tarjetas AVR Butterfly y LPCXpresso por medio de un mismo canal, optimizando el uso de los microcontroladores (ATMega169 y LPC1769), logrando así la interacción entre distintos elementos de diferentes fabricantes, reduciendo cableado y programación de pines.
2. Los microcontroladores de los fabricantes ATMega y LPCXpresso son de gran utilidad para el control de motores y otras funciones debido a que tienen gran capacidad de almacenamiento,

poco retardo en sus tiempos de respuesta al momento de realizar el envío y/o recepción de datos y su sencilla programación en comparación con algoritmos complejos que poseen otros tipos de microcontroladores, lo que los convierte en elementos de gran utilidad en diferentes campos de estudio.

3. La tarjeta AVR Butterfly, al poseer una pantalla LCD y un joystick integrados a su estructura, se convierte en un microcontrolador muy funcional para la operación de giro y velocidad de un motor BLDC, lo que nos permitió aprovechar espacio en la circuitería e implementación del proyecto puesto que en otros microcontroladores estos elementos no vienen incorporados, además se demostró que, pese a que se realizó la comunicación entre dos tarjetas de distinto fabricante, se pudo lograr la comunicación entre ellas, lo que las convierte en herramientas de amplia interacción con otros dispositivos similares.
4. Mediante la implementación de los ejercicios y el proyecto pudimos comprender y tener una visión más amplia acerca del funcionamiento de los microcontroladores ATmega169 y LPC1769 y los motores BLDC, así como la forma en que se realiza la comunicación entre dispositivos mediante protocolo SPI e identificar de manera clara la función que desempeñan el maestro y el esclavo en el dispositivo en el cual han sido configurados.
5. En general, la aplicación de microcontroladores en el uso industrial y otras áreas es un campo que se encuentra innovando y cambiando constantemente, y cada vez podemos observar nuevos modelos de microcontroladores con mayor capacidad, herramientas incorporadas y programación más sencilla. Por cumplir con todos estos parámetros, podemos concluir que los microcontroladores que han sido objeto de estudio en esta tesina son herramientas nuevas y con gran ventaja sobre los microcontroladores ya conocidos y estudiados en cursos anteriores durante nuestro proceso de desarrollo académico.

5. Recomendaciones.

1. Se requiere trabajar con una fuente de alimentación de 3V o más para que se pueda realizar exitosamente la comunicación SPI entre las tarjetas. Al trabajar con una fuente de menor voltaje, se puede cometer errores en la comunicación y el maestro podría estar enviando datos erróneos al esclavo, ejecutando mal la transmisión de los datos.

2. Al momento de cablear es importante verificar continuidad entre los puntos que están siendo conectados para de esta manera evitar fallos al momento de hacer funcionar nuestro proyecto, además se requiere utilizar una sola tierra de referencia y un solo punto Vcc (3V) para ahorro de espacio y evitar problemas con el voltaje aplicado a los pines de las tarjetas, las cuales son sensibles a las variaciones bruscas de voltaje.
3. Para la programación y uso de los pines del microcontrolador es necesario revisar el datasheet de cada uno de ellos y verificar qué pines se encuentran reservados por el fabricante, qué pines pueden ser usados como E/S y qué pines sirven para funciones específicas, además se requiere verificar que se encuentren correctamente conectados los 4 puertos que permiten la comunicación SPI entre maestro y esclavo para poder realizar la transmisión.

6. Referencias.

[1] AN857, Control de Motor DC, Disponible en: <http://ww1.microchip.com/downloads/en/appnotes/00857a.pdf>.

Fecha de consulta febrero 2012.

[2] Conexión entre un dispositivo maestro y un esclavo, Disponible en: http://quierobits.com/wp-content/uploads/2011/02/spi_topology1.png.

Fecha de consulta marzo 2012.

[3] Introducción a LPCXpresso y repaso de lenguaje C – Seminario de sistemas embebidos – FIUBA Autor Alan Kharsansky Abril 2011, Disponible en: http://laboratorios.fi.uba.ar/lse/seminario/material-2011/Sistemas_Embebidos-2011_2doC-Intro_a_LPCXpresso_y_repaso_lenguaje_C-Kharsansky.pdf

Fecha de consulta abril 2012.

[4,5] Esquema de los periféricos externos e internos del AVR Butterfly, Disponible en: http://bibing.us.es/proyectos/abreproy/11175/fichero/v01%252Frv_T4_Hardware_v03d.pdf.

Fecha de consulta abril 2012.