



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



Control mediante joystick de tarjeta AVR Butterfly (con microcontrolador ATmega169) mediante comunicación SPI con tarjeta LPCXpresso controladora de motor BLDC

Walter Orellana⁽¹⁾, Gian Banchón⁽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⁾
walanore@espol.edu.ec⁽¹⁾, gcbancho@espol.edu.ec⁽²⁾, cvaldiv@fielc.espol.edu.ec⁽³⁾

Resumen

El objetivo principal de este proyecto es diseñar un sistema maestro – esclavo mediante una comunicación SPI entre las tarjetas AVR Butterfly y LPCXpresso para controlar un Motor BLDC mediante un Joystick.

Se presenta un enfoque práctico y específico, para enlazar y comprender toda la teoría del funcionamiento de este protocolo; haciendo uso de varias herramientas como el software AVR STUDIO 4, para programar el kit AVR BUTTERFLY que trabaja con microcontrolador ATmega169; y el software LPCXpresso 4, para programar el microcontrolador LPC1769.

Palabras claves: Microcontrolador ATMEGA169, Microcontrolador LPC1769, AVR Butterfly, LPCXpresso, SPI

Abstract

The main objective of this project is design a master – slave system through SPI communication between AVR Butterfly and LPCXpresso cards to control a BLDC motor using a joystick.

It offers a practical and specific approach, to link and understand the whole theory of operation of this protocol, making use of various software tools such as AVR Studio 4 to program the AVR Butterfly kit which works with ATmega169 microcontroller, and the LPCXpresso 4 software, to program the LPC1769 microcontroller.

Keywords: ATMEGA169 Microcontroller, LPC1769 Microcontroller, AVR Butterfly, LPCXpresso, SPI

1. Introducción

El objetivo del trabajo fue el diseño e implementación de un control mediante Joystick de un motor BLDC (DC Sin Escobillas), usando el protocolo SPI (Interfaz Serial Periférica) entre la AVR Butterfly y la LPCxpresso. Para mostrar el funcionamiento de este protocolo de comunicación usamos y aprovechamos las diferentes características que nos ofrecen las tarjetas mencionadas anteriormente.

Este trabajo consta de 3 ejercicios previos necesarios para la realización del proyecto y el control del motor BLDC; desarrollados en lenguaje C.

Todas las plataformas manejan un dispositivo Maestro que envía las señales y un dispositivo esclavo recibe las señales. Estos fueron dispuestos en diferentes combinaciones mediante las tarjetas AVR Butterfly y el LPCxpresso; siendo una maestro y la otra esclavo o viceversa

2. Descripción del Proyecto

A medida que transcurren los años, los circuitos integrados van evolucionando: se fabrican en tamaños cada vez más pequeños, con mejores características y prestaciones, mejoran su eficiencia y su eficacia, y se permite así que mayor cantidad de elementos sean empaquetados (integrados) en un mismo chip [1].

Para realizar la comunicación entre estos circuitos integrados nos valemos de la SPI (Serial Peripheral Interface Bus o bus serial de interfaz de periféricos) es un estándar de enlace de datos seriales sincronizados por un reloj que operan en modo full duplex. Los dispositivos se comunican en modo maestro/esclavo donde el dispositivo maestro inicia el data frame (trama de red o marco de datos). Múltiples dispositivos esclavos están permitidos en líneas Slave Select (SS) individuales [2].

Este tipo de comunicación fue elegida para el control de nuestro motor, ya que minimiza el número de conductores, pines y el tamaño del circuito integrado, reduciendo los costos de fabricar, montar y probar la electrónica

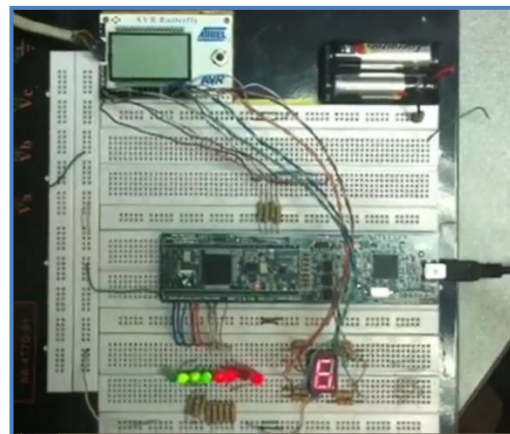


Figura 2.1: Implementación del contador de 0 a 9 entre LPC – AVR

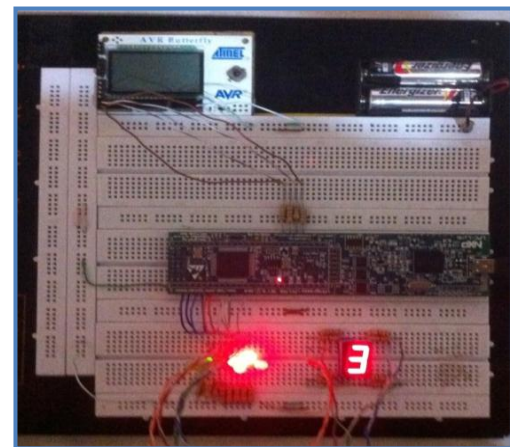


Figura 2.2: Implementación del contador de 0 a 9 entre AVR – LPC

3. Herramientas de Hardware

En esta sección se detallan los elementos utilizados para la implementación del proyecto.

3.1 AVR Butterfly (ATmega169)

El AVR Butterfly contiene un microcontrolador ATmega169, el cual va a realizar el comando de las diferentes funciones de las que es capaz este kit. Hace uso del joystick, el LCD y el buzzer.

Soporta programación ISP y programación mediante bootloader por medio de un puerto serial

Su alimentación se realiza a través de una batería tipo botón de 3 V la cual proporciona la energía necesaria para su funcionamiento a razón de 600 mAh.

El ATmega169 tiene entre sus características principales 16K de memoria FLASH, 1K de SRAM, 512 bytes de EEPROM y posee 4 canales PWM además de un comparador analógico [4].

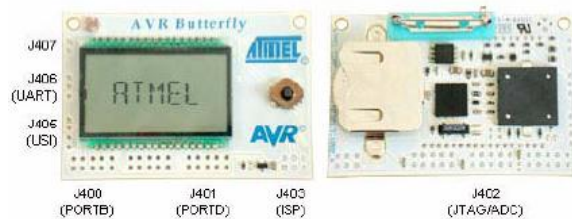


Figura 3.1: Kit AVR Butterfly [6]

Características del Kit AVR Butterfly:

- LCD de 120 segmentos.
- Joystick de cuatro direcciones con empuje el centro de la entrada del usuario.
- Elemento piezoeléctrico para reproducir sonidos.
- Cristal de 32 kHz para el reloj en tiempo real.
- 4Mbit DATAFLASH de almacenamiento de datos.

- RS-232-convertidor de nivel, para comunicarse con las unidades fuera de borda.
- NTC termistor para medir la temperatura [4].

3.2 LPCXpresso (LPC1769)

El LPC1769 es un microcontrolador basado en ARM Cortex-M3 para aplicaciones embebidas que requieren un alto nivel de integración y baja disipación de potencia. El procesador ARM Cortex-M3 es la próxima generación de los CORE, que ofrece mejoras en el sistema de depuración, tales como modernización características y un mayor nivel de apoyo a la integración del bloque. Las versiones de alta velocidad (LPC1769 y LPC1759) operan a velocidades de hasta 120 MHz de frecuencia. Otras versiones operan a velocidades de hasta una frecuencia de 100 MHz [3].

Características

- Los procesadores de ARM cortex-M3 corren a frecuencias de hasta 100 MHz (LPC1768/67/66/65/64/63) o de hasta 120 MHz (LPC1769).
- ARM Cortex-M3 contiene un controlador anidado de vector de interrupciones.
- Hasta 512 kB de memoria de programación flash en un chip. Mayor acelerador de memoria flash permite alta velocidad de operación de 120 MHz con cero estados de espera.
- Programación ISP e IAP a través de un chip gestor de arranque de software.
- Cuatro modos de reducción de energía: Sleep, Deep-sleep, Power-down, and Deep power-down [3].

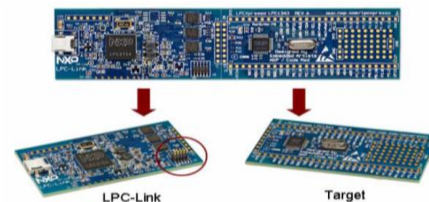


Figura 3.2: Hardware LPC1769 [5]

4. Herramientas de Software

4.1 AVR Studio 4

AVR Studio es un entorno de desarrollo integrado que incluye un editor, el ensamblador, descargador de archivos HEX y un emulador de microcontroladores.

Este software permite escribir y depurar aplicaciones AVR en entornos de Windows, que proporciona una gestión de proyectos de herramientas, fuente de editor de archivos, simulador, ensamblador y front-end para C / C + +, emulación y programación en el chip.

AVR Studio es compatible con la gama completa de herramientas de ATMEL AVR y cada versión siempre contiene las últimas actualizaciones para las herramientas y el apoyo de los dispositivos AVR nuevos.

Para programar el microcontrolador AVR Atmel usando el lenguaje de programación C, también necesitaremos de una herramienta extra dentro del AVR Studio, que es el WinAVR, el cual consiste en un compilador para AVR basado en GCC. Este aparece en AVR Studio como un plug-in [6].



Figura 4.1: Página de inicio del AVR Studio4 [8]

4.2 LPCXpresso 4

LPCXpresso (creado por Code Red) es un software altamente integrado desarrollado para trabajar con microcontroladores LPC de NXP, que incluye todas las herramientas necesarias para desarrollar soluciones de software de alta calidad, de una manera efectiva en tiempo y costo.

LPCXpresso es una nueva plataforma de bajo costo de desarrollo disponible en NXP. El software consiste de algunos aumentos para su mejora, como: IDE basado en Eclipse, un compilador de C basado en GNU, links, librerías, y un depurador GDB mejorado.

LPCXpresso se basa en Eclipse, que permite muchas mejoras específicas en LPC. También cuenta con la versión actual de una cadena de herramientas de la industria estándar GNU, con una patente basada en librerías en C. El LPCXpresso IDE puede construir un ejecutable de cualquier tamaño con optimización del código completo, y soporta un límite de descarga de 128 KB después del registro [5].

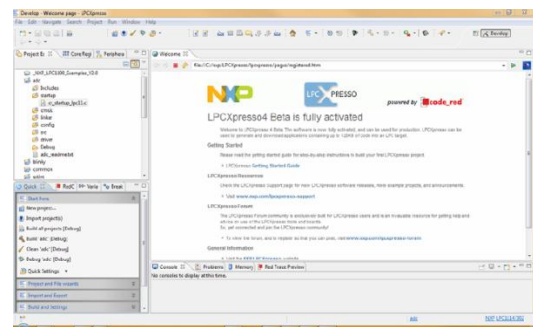
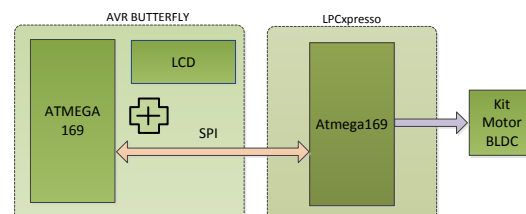


Figura 4.2: LPCXpresso IDE [5]

5. Diagrama de Bloques del Proyecto

A continuación se muestra el diagrama de bloques que describe el funcionamiento de del control del motor BLDC mediante comunicación SPI.



6. Funcionamiento del proyecto

El maestro, tarjeta AVR Butterfly (con microcontrolador Atmega169), envía mediante comunicación SPI, órdenes a la tarjeta LPCxpresso (con microcontrolador LPC169) que está como esclavo.

El motor es controlado por el Joystick de la AVR Butterfly, que con movimiento de arriba y abajo incrementará y disminuirá la velocidad del motor respectivamente, el movimiento a la izquierda y presionar en el centro encenderá y apagará el motor, el movimiento a la derecha invertirá el giro.

Todas estas órdenes se irán presentando en el LCD de la AVR Butterfly. Esas órdenes se saldrán en los cuatro primeros bits del puerto 2 de la LPCxpresso ; que irán conectados al kit del motor (LPC1114).

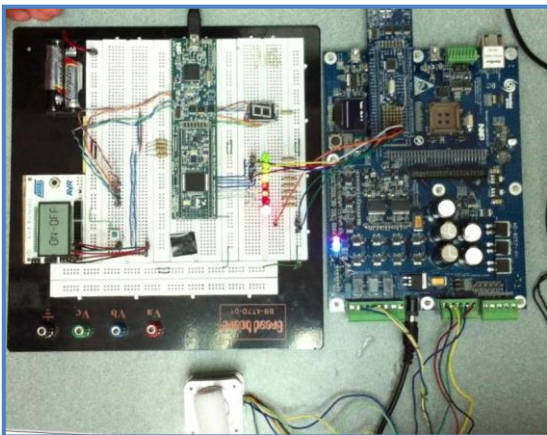


Figura 6.1: Implementación del controlador de motor BLDC mediante Joystick

El joystick tiene 5 movimientos. Para nuestro proyecto se configuró el envío de 4 órdenes, como se muestra en la tabla 6.1 cuyo reconocimiento se realiza en un estado bajo.

JOYSTICK (movimiento)	ORDENES	LCD
Arriba	Aumentar Velocidad	SPEED+
Abajo	Disminuir Velocidad	SPEED-
Derecha	Cambio de Giro	INVERT
Izquierda o Centro	Arranque/Pare	ON-OFF

Tabla 6.1: Órdenes enviadas por el joystick

7. Pruebas

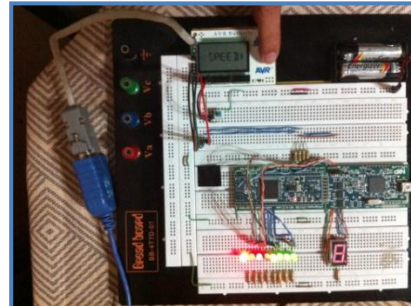


Figura 7.1: Movimiento del joystick hacia ARRIBA

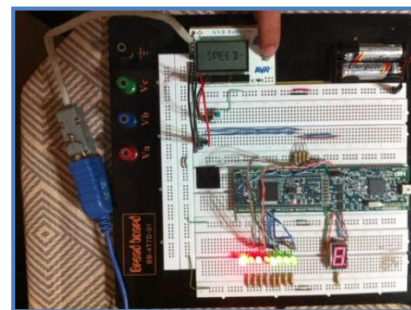


Figura 7.2: Movimiento del joystick hacia ABAJO

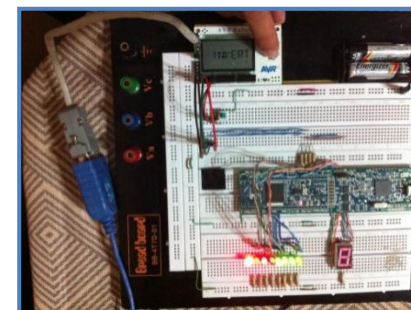


Figura 7.3: Movimiento del joystick hacia DERECHA

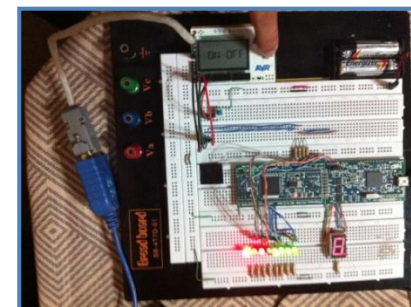


Figura 7.4: Movimiento del joystick hacia IZQUIERDA/CENTRO



8. Conclusiones

1. La El protocolo SPI es una interfaz de comunicación que permite controlar varios periféricos pudiendo utilizar para todos ellos un mismo bus de datos. Esta comunicación nos permitió diseñar un sistema maestro – esclavo, cuyo fin consiste en el control del motor BLDC mediante un intercambio de datos utilizando comunicación SPI entre las dos tarjetas empleadas en nuestro proyecto (AVR - LPC).
2. Esta nueva estrategia de control presentada, a través de modernos microcontroladores, nos brinda una característica destacable del sistema, debido al gran grado de integración que existe entre esta variedad de tarjetas. Las tarjetas de control utilizadas en este proyecto resultan mucho más compactas que la de los sistemas de control basadas en componentes análogos. Facilitando así su manejo, instalación y programación.
3. En cuanto a los resultados experimentales, estos han demostrado que la estructura de control propuesta, basada en el manejo mediante joystick de un motor BLDC, permite debido al gran grado de integración de las tarjetas y al fácil control del joystick, una total confiabilidad en el mando del motor.
4. El desarrollo de este proyecto fue realizado en lenguaje C haciendo uso de las herramientas como el AVR Studio 4 que sirve para programar el micro controlador ATmega 169, y del LPCXpresso 4 que sirve para programar la tarjeta LPC 1769, convirtiéndose ambas herramientas en el corazón de nuestro proyecto. Siendo el lenguaje C el más

óptimo en el desarrollo de software para microcontroladores, ya que permite segmentar el código fuente en varias funciones especializadas y archivos, permitiéndole al programador reusar las mismas funciones en otras aplicaciones.

9. Recomendaciones

1. Entre las cosas a tomar en cuenta cuando se trata de implementar una comunicación SPI. Una de estas es el cableado. Es importante asegurarse de que la entrada serie del microcontrolador esté conectada correctamente a la salida serie del microcontrolador con el que se realiza las comunicaciones.
2. En lo que a programación se refiere se recomienda como primer paso realizar pruebas demo, con los muestras que ya vienen incorporados en cada software, que nos permitió como por ejemplo en particular, familiarizarnos con el software AVR STUDIO 4 para la programación del AVR Butterfly, realizando pruebas como es prender la pantalla LCD y ver los comandos que tiene el Joystick.
3. Gracias a la variedad de experimentos realizados acerca de la comunicación mediante el protocolo SPI entre diferentes combinaciones de dispositivos empleados, podemos recomendar como ingenieros el empleo de este protocolo ya que según nosotros es el más rápido, eficaz y sencillo, debido a que los comandos o instrucciones necesarios para operar el protocolo son específicos y relativamente simples, como también el empleo de cuatro líneas de comunicación que en realidad es lo más destacable en lo que al protocolo se refiere.



10. Referencias

[1] Nobelprize.org, The History of the Integrated Circuit. (17/03/12) Página Internet:

http://www.nobelprize.org/educational/physics/integrated_circuit/history/index.html

[2] Eric López Pérez, Ingeniería en Microcontroladores. (09/03/12) Página Internet:

<http://www.i-micro.com/pdf/articulos/spi.pdf>

[3] User Manual LPC1769. (23/02/12) Página Internet:

http://www.nxp.com/documents/user_manual/UM10360.pdf

[4] AVR Butterfly- Atmel Corporation. (25/03/12) Página Internet:

<http://www.ddrservice.net/files/Microcontrollers/atmel/atmega169.pdf>

[5] NXP semiconductors. (01/04/12) Página Internet:

<http://ics.nxp.com/support/documents/microcontrollers/pdf/lpcxpresso.getting.started.pdf>

[6] ATMEL corporation. (23/03/12) Página Internet:

<http://www.atmel.com/Images/doc4271.pdf>

[7] NXP semiconductors. (04/04/12) Página Internet:

http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf

[8] Ikkaro. (01/04/12) Página Internet:

<http://www.ikkaro.com/book/export/html/1416>

[9] Software LPCXpresso 4, ejemplo. (17/04/12) Dirección en PC:

C:\npx\LPCXpresso_4.1.5_219\lpcxpresso\Examples\NXP\LPC1000\LPC17xx\NXP_LPCXpresso1769_MCB1700_2011-02-11.zip