



Control mediante joystick de tarjeta AVR Butterfly (con microcontrolador ATmega169) mediante comunicación I2C con tarjeta LPCXpresso controladora de motor BLDC

Lenin Toaza ⁽¹⁾, John Caro ⁽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)}
jojacar@espol.edu.ec ⁽¹⁾, lenfatao@espol.edu.ec ⁽²⁾, cvaldiv@fiec.espol.edu.ec ⁽³⁾

Resumen

El proyecto realizado en la materia de graduación Microcontroladores Avanzados, consiste en manejar y controlar un motor BLDC a través de un joystick combinando tarjetas de desarrollo como la LPCXpresso 1769 y la AVR Butterfly utilizando comunicación I2C (Circuitos Inter Integrados). Para la comunicación entre estas tarjetas, la AVR Butterfly trabajó en modalidad MAESTRO, mientras que la tarjeta LPCXpresso 1769 trabajó en modalidad ESCLAVO. Para la implementación de este proyecto se usó lenguaje C mediante el entorno LPCXpresso 4 y AVR Studio 4 respectivamente.

Palabras claves: BLDC, AVR Butterfly, LPCXpresso 1769, I2C

Abstract

The project implemented in the Advanced Microcontroller graduation subject, consists in handling and controlling a BLDC motor using a joystick combining the development boards LPCXpresso 1769 and the AVR Butterfly using I2C (Inter Integrated Circuit) communication. For communication between these development boards, the AVR Butterfly worked in master mode, while LPCXpresso 1769 board worked in slave mode. To implement this project we used C language in the environments LPCXpresso 4 and AVR Studio 4 respectively.

Keywords: BLDC, AVR Butterfly, LPCXpresso 1769, I2C

1. Introducción

El objetivo del trabajo fue el diseño e implementación de un control mediante Joystick de un motor BLDC, mediante comunicación I2C con tarjeta LPCXpresso. Teniendo en cuenta varios aspectos muy importantes como es la compatibilidad del LPCXpresso con otras tarjetas y combinarlas para tener un desarrollo en conjunto.

La tarjeta AVR Butterfly mediante su joystick es la que controla el sentido y la velocidad de giro del motor BLDC. Esta tarjeta envía señales a la LPCXpresso que es la tarjeta controladora del motor BLDC mediante comunicación I2C actuando la tarjeta AVR como Maestro y la LPCXpresso que recibirá esta comunicación actuará como Esclavo.

2. Protocolo I2C

I2C es un protocolo que facilita la comunicación entre distintos dispositivos como microcontroladores, memorias, monitores de computadoras, y muchos otros dispositivos.

El protocolo I2C utiliza un bus que requiere de dos señales: Una de datos y una de reloj (además de la tierra común). Este protocolo fue diseñado por la empresa Koninklijke Philips Electronics N.V. inicialmente con una tasa de transmisión de datos de 100Kbps, sin embargo actualmente se alcanzan tasas de hasta 3.4Mbps.

La señal de datos se llama SDA y la de reloj se llama SCL o SCK. Como son señales de colector abierto, es necesario agregar resistencias de pull-up que pueden tener valores entre 2KΩ y 50KΩ siendo 10KΩ su valor estándar.

El estado de reposo de ambas líneas en el bus I2C es el "1" lógico, de ahí el significado de las resistencias pull-up. El estándar I2C permite una comunicación bidireccional con los periféricos y además es posible crear entornos multi-maestros sin que entre ellos haya conflictos, tal como muestra la Figura 2.1.

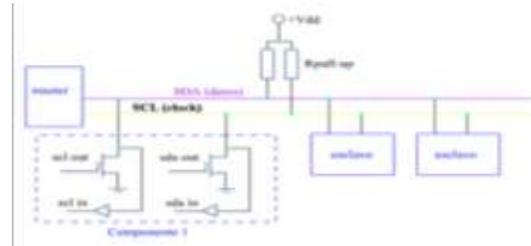


Figura 2.1 Bus I2C [1]

El protocolo I2C también permite tener conectados al bus dispositivos que funcionen con voltajes de alimentación distintos, siempre que el voltaje de SDA y SCL sea menor que todos los voltajes de polarización de los dispositivos conectados al bus I2C y que los niveles lógicos del bus (alto y bajo) puedan ser interpretados correctamente por cada uno de los dispositivos conectados. [1]

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



Figura 3.1 Kit AVR Butterfly [2]

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.[2]

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.

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.
- Solo 3,3 V de voltaje de alimentación.[3]

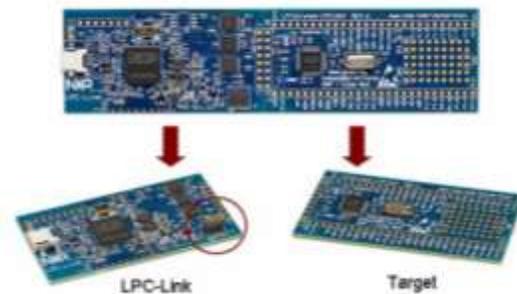


Figura 3.2 LPC1769 [3]

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. [4]

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]

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 I2C entre tarjeta AVR Butterfly y LPCXpresso.

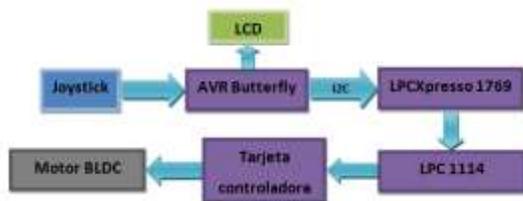


Figura 5.1 Diagrama de bloques proyecto final

6. Funcionamiento del proyecto

En la Figura 6.1 se muestra el funcionamiento total de la comunicación I2C entre la AVR Butterfly como maestro y la LPCXpresso 1769 como esclavo controlando a otra tarjeta que es la LPC 1114 que a su vez es la encargada de enviar los comandos necesarios al driver (controlador) del motor BLDC.

El maestro es el encargado de dar la señal de inicio, seguido de la dirección del esclavo de la LPCXpresso 1769 que en este caso es 0xA0. Una vez que la LPC recibe esta dirección, envía un ACK y el AVR procede a enviar el dato, que es el byte correspondiente a la posición del joystick en ese momento.

Una vez que la LPCXpresso 1769 recibe el dato envía un acuse de recibo ACK y almacena el dato temporalmente en el buffer de lectura, e inmediatamente lo coloca en el puerto 2.

Los valores de los datos que se envían tienen un propósito, y es de simular unas botoneras en configuración pull-up. Como en la LPC 1114 solo se necesitan cuatro botoneras que realizan todas las funciones de mando, entonces, del puerto 2 de la LPCXpresso 1769 solo van a estar funcionando los 4 pines menos significativos, y por eso es que los cuatro bits más significativos de los datos son siempre bajos, mientras que tres de los cuatro bits menos significativos son altos y uno de ellos es bajo, dependiendo de la posición del joystick, de tal manera que simule el hecho de que una de las cuatro botoneras se ha presionado, que representa el bit que está en bajo, mientras que las otras tres botoneras restantes que se encuentran sin presionar representan los tres altos.

En resumen, cuando el joystick se encuentra en estado de reposo, se envía el valor predeterminado de "0x0F" lo que hace que los pines P2.0, P2.1, P2.2 y P2.3 permanezcan en alto, simulando los altos de las pull-up. Cuando el joystick se encuentra presionado en el centro se envía "0x07" lo que hace que el pin P2.3 sea un bajo, mientras que los demás pines permanezcan en alto. Cuando el joystick vuelva a su estado predeterminado el pin P2.3 vuelve a ser un alto, lo que simula una botonera que se ha presionado y soltado. Un bajo seguido de un alto en el pin P2.3 hace que el motor arranque, siempre y cuando no se encuentre actualmente funcionando, en ese caso, ocasionará que el motor se detenga.

Cuando el joystick se encuentra presionado hacia la derecha se envía "0x0B" y luego que el joystick deja de presionarse se envía el valor predeterminado "0x0F" lo que hace que el pin P2.2 se ponga en bajo y luego en alto, y todos los demás pines permanezcan en alto. Esto hace que el motor invierta su giro.

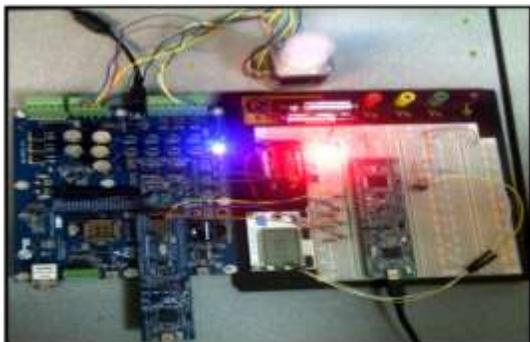


Figura 6.1: Implementación del controlador de motor BLDC mediante Joystick

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	UP
Abajo	Disminuir Velocidad	DOWN
Derecha	Cambio de Giro	INVERT
Izquierda o Centro	Arranque/Pare	START

Tabla 6.1: Órdenes enviadas por el joystick

7. Conclusiones

1. Se han presentado diversos ejemplos en donde se establece el protocolo de comunicación serial I2C. Por lo tanto concluimos que este protocolo es eficaz ya que requiere solo de dos señales, SDA y SCL para transmitir o recibir datos a velocidades relativamente altas (100Kbps como tasa de transmisión estándar).
2. Se ha experimentado con las tarjetas de desarrollo LPCXpresso y la AVR Butterfly. Estas tarjetas presentan grandes ventajas frente al uso de microcontroladores por sí solos, ya que en sus tarjetas integran circuitos de protección, módulos, dispositivos terminales, programadores y debuggers. Esto permite ahorrar el uso de circuitería externa para el diseño de prototipos.
3. Se ha logrado establecer el control del motor BLDC, el cual posee mayor eficiencia que un motor DC con escobillas, ya que conmuta

electrónicamente y no mecánicamente. Además, su ruido eléctrico es muy bajo y no tiene caída de tensión puesto que no posee escobillas. Estas ventajas han permitido que su uso se esté incrementando para diversas aplicaciones en la actualidad.

8. Recomendaciones

1. Cuando se programe la LPC es recomendable evitar la manipulación de los circuitos con los dedos, debido a que se puedan ocasionar descargas electrostáticas que puedan dañar la tarjeta. En el caso de que el programa no pueda ser ejecutado mediante el debugger de la tarjeta LPCXpresso, verificar si el cable USB está bien conectado de lo contrario cambie de cable.
2. La programación del AVR Butterfly se la debe hacer con la tarjeta energizada y presionando un botón del joystick, teniendo en cuenta que si tiene puertos virtuales instalados en su computadora deberá desinstalarlos ya que el programa elige los puertos virtuales del uno al cuatro.
3. En el bus I2C es importante colocar una resistencia de pull-up a cada línea, tanto a la SDA como a la SCL, los rangos de valores de resistencias aceptables es muy amplio, desde 1.8 K Ω hasta 47 K Ω , sin embargo escoger valores bajos de resistencia incrementa el consumo de los integrados pero disminuye la sensibilidad al ruido y mejora el tiempo de los flancos de subida y bajada de las señales. Un valor recomendado es de 4.7 K Ω .

9. Referencias

- [1] Fernández Moreno Antonio, Conceptos básicos del bus I2C, <http://www.uco.es/~el1mofer/Docs/IntPerif/BusI2C.pdf>, fecha de consulta Abril 1 del 2012.
- [2] Escuela Superior Politécnica del Ejército, Características del Kit AVR Butterfly en español, <http://www.espe.edu.ec/repositorio/T-ESPE-014271.pdf>, fecha de consulta Marzo 30 del 2012



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



[3] NXP Semiconductors. LPCXpresso Microcontrollers. Disponible en: <http://ics.nxp.com/lpcxpresso/~LPC1769/>, fecha de consulta Marzo 30 del 2012

[4] ATMEL. AVR Studio 4, <http://www.elec.uow.edu.au/avr/guides/Getting-started-Cprogramming-Atmel-AVR.pdf>, fecha de consulta Marzo 22 del 2012.

[5] NXP Semiconductors, Lpc1769/68/67/66/65/64/63, www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf, fecha de consulta marzo 17 del 2012.