



# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

## **TESINA DE SEMINARIO**

**“CONTROL DE MOTORES SIN ESCOBILLAS (BLDC) Y CON SENSORES USANDO EL MICROCONTROLADOR ARM CORTEX3 CON 32 BITS DE LPCXPRESSO, MEDIANTE COMANDOS ENVIADOS DESDE TARJETA BUTTERFLY (CON MICROCONTROLADOR ATMEGA169)”**

Previa la obtención del Título de:

**INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL**

Presentado por:

Guartatanga Cedeño Pablo Arturo

Córdova Alvarado Danny Fabricio

GUAYAQUIL – ECUADOR

AÑO 2012

## AGRADECIMIENTO

A Dios, a mis padres Enrique Córdova y Amarilis Alvarado.

A mis amigos y familiares quienes también me han apoyado en cumplir mis metas y mis sueños.

A mis maestros, profesores y tutores que a lo largo de mi vida han compartido sus conocimientos conmigo.

***Danny Córdova A.***

A Dios, a mis padres Pablo Guartatanga y Graciela Cedeño.

A todos y cada uno de los que nos ayudaron con sus conocimientos, consejos y sus palabras de aliento para seguir adelante en la carrera.

***Pablo Guartatanga C.***

## DEDICATORIA

Dedico este trabajo a Dios, a mis padres, a mis tutores y a cada uno de quienes han sido ayuda para mí en este proceso.

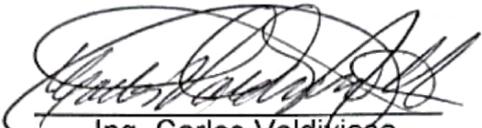
***Danny Córdova A.***

A Dios por ser nuestro creador, amparo y fortaleza y estar presente cuando más lo necesitamos, y por hacer palpable su amor a través de cada uno de los que nos rodea.

A mis padres y profesores, que sin esperar nada a cambio, han sido pilares en nuestro camino y así, forman parte de este logro que nos abre puertas inimaginables en nuestro desarrollo profesional.

***Pablo Guartatanga C.***

## TRIBUNAL DE SUSTENTACIÓN



Ing. Carlos Valdivieso

PROFESOR DE SEMINARIO DE GRADUACIÓN



Ing. Ronald Ponguillo

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

## DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta tesina, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

(Reglamento de exámenes y títulos profesionales de la ESPOL)



---

Danny Córdova Alvarado



---

Pablo Guartatanga Cedeño

## RESUMEN

El presente documento corresponde al seminario de graduación de “Microcontroladores Avanzados”, el objetivo del proyecto es crear un prototipo para el “Control de motores BLDC con sensores usando el microcontrolador ARM Cortex-M3 de la tarjeta LPCXpresso 1769” de NXP Semiconductors dada a sus excelentes características y la tarjeta Butterfly con microcontrolador ATmega169 para él envió de comandos hacia la tarjeta LPCXpresso 1769 para el control del motor BLCD, también se hace uso de la tarjeta LPCXpresso Motor Control Kit como driver para el motor BLCD.

En el primer capítulo, se describe de forma general el funcionamiento del proyecto, como también los antecedentes, motivación, identificación del problema, objetivos principales, limitaciones, descripción de sus partes, etc.

En el segundo capítulo, se describe la estructura interna del motor BLCD con sensores, como también su funcionamiento, características, ventajas y desventajas y las herramientas de hardware a ser utilizadas en este proyecto como la tarjeta LPCXpresso LPC1769, LPCXpresso Motor Control Kit y de la tarjeta Butterfly como también las herramientas de software LPCXpresso v4.1.5\_219 para la programación de la tarjeta LPCXpresso 1769 y AVR

STUDIO 4 para la tarjeta Butterfly, y también la teoría detrás de su funcionamiento.

El tercer capítulo, trata del diseño e implementación del proyecto, así como la programación para la tarjeta LPCXpresso LPC1769 mediante el software para dicha tarjeta una breve descripción y una lista de materiales de los elementos que contiene el proyecto.

En el cuarto y último capítulo se detalla, desarrolla e implementa el proyecto; incluyendo el diagrama de bloques, diagrama de flujo, descripción del algoritmo en conjunto con las pruebas y simulaciones.

## ÍNDICE GENERAL

RESUMEN.....	VI
ÍNDICE GENERAL .....	VIII
ÍNDICE DE FIGURAS.....	XI
ÍNDICE DE TABLAS.....	XIII
ABREVIATURAS.....	XIV
INTRODUCCIÓN.....	XV
CAPÍTULO 1.....	1
1. Descripción General del Proyecto.....	1
1.1 Antecedentes .....	1
1.2 Motivación .....	2
1.3 Identificación del Problema.....	3
1.4 Objetivos Principales .....	3
1.5 Limitaciones del Proyecto.....	4
1.6 Descripción de las Partes del Proyecto .....	5

CAPÍTULO 2.....	7
2. Sustentación Teórica .....	7
2.1 Motores BLDC .....	7
2.2 LPCXpressoLPC1769 .....	21
2.3 LPCXpresso v4.1.5_219.....	22
2.4 LPCXpresso Motor Control Kit.....	26
2.5 LPCXpressoLPC1114 .....	28
2.6 Tarjeta AvrButterfly.....	31
CAPÍTULO 3.....	39
3. Configuración e Implementación del “Motor Control Kit” .....	39
3.1 Ensamblando el Motor Control Kit: .....	39
3.2 Paso 1: Montaje de la tarjeta LPCXpresso LPC1114.....	42
3.3 Paso 2: Conexión del Motor BLCD .....	42
3.4 Paso 3: Conectando la Alimentación .....	44
3.5 Programando la LPCXpresso LPC1114: .....	45
3.6 Conexión de las Tarjetas .....	46
CAPÍTULO 4.....	48
4. Pruebas Experimentales del Proyecto .....	48
4.1 Parámetros del Motor Programados En la LPC1114 .....	49
4.2 Programando la LPCXpressoLPC1769: .....	51

4.3	Programando la AVR ButterFly:.....	53
4.4	Pruebas del Control de Velocidad del Motor.....	55

CONCLUSIONES

RECOMENDACIONES

BIBLIOGRAFÍA

ANEXOS

## ÍNDICE DE FIGURAS

Figura 1.1: Diagrama de Bloques del Proyecto.....	5
Figura 1.2: Motor Control Kit.....	6
Figura 2.1: Circuito Equivalente de un Motor DC .....	8
Figura 2.2: Curva Torque-Velocidad de un Motor DC .....	10
Figura 2.3: Corte Transversal de un Motor BLCD .....	14
Figura 2.4: Conmutación Electrónica de un motor BLCD.....	16
Figura 2.5: Curvas de Voltaje de las fases de un motor BLDC .....	18
Figura 2.6: Constitución interna de un motor BLDC .....	20
Figura 2.7: LPCXpresso IDE.....	23
Figura 2.8: Motor Control Kit.....	28
Figura 2.9: Diagrama de Bloques de la Tarjeta AvrButterfly.....	33
Figura 2.10: Periféricos externos de la Tarjeta AvrButterfly. ....	35
Figura 2.11: Diagrama de las conexiones de los periféricos externos con el microcontrolador ATmega169.....	37
Figura 3.1: LPCXpresso 1769 sin espadines .....	40
Figura 3.2: Soldando los espadines a la tarjeta .....	40
Figura 3.3: Tarjeta LPCXpresso 1769 montada sobre protoboard .....	41
Figura 3.4: Conector de los Sensores Efecto Hall.....	42
Figura 3.5: Conector de alimentación trifásica del motor BLCD .....	43
Figura 3.6: Motor BLDC conectado a los sockets .....	43

Figura 3.7: Motor Control Kit .....	44
Figura 3.8: Programando la tarjeta LPCXpresso 1114 .....	45
Figura 3.9: Conexión de las tarjetas 1 .....	46
Figura 3.10: Conexión de la tarjetas 3 .....	47
Figura 4.1: Señales del Joystick en LPC1769.....	55
Figura 4.2: Parámetros y control del Motor en la tarjeta LPCXpresso 1114 ..	56
Figura 4.3: La tarjeta motor control kit en funcionamiento.....	58
Figura 4.4: Motor BLDC controlado por Joystick de la tarjeta AvrButterfly. ...	59

# ÍNDICE DE TABLAS

Tabla 1: Comparación de un motor DC con un motor BLDC.....	11
Tabla 2: Descripción de periféricos de la tarjeta Butterfly .....	36

## ABREVIATURAS

BLDC	Brushless DC
DC	Direct Current
PM	Permanent Magnet
PMSM	Permanent Magnet Synchronous Motor

## INTRODUCCIÓN

Desde la creación de los motores, ya sea motores diseñados para la industria como también los utilizados para proyectos donde se necesiten motores pequeños, desde entonces ha existido la necesidad de controlar su arranque, velocidad y giro del rotor. El principal objetivo de este proyecto es la implementación del control del motor BLCD usando el microcontrolador ARM Cortex-M3 para el control del arranque, inversión de giro y para el incremento o decremento de la velocidad.

Para llevar acabo este proyecto se hizo uso de varias herramientas como el software LPCXpresso v4.1.5\_219 y el AVR STUDIO 4 para poder programar el cerebro del Motor Control Kit y la tarjeta butterfly para el envío de comandos respectivamente. En el proyecto se usa la comunicación UART para enviarles los comandos desde la tarjeta butterfly hacia la LPCXpresso LPC1769, como veremos mas adelante el microcontrolador AMR cortex-M3 nos permite también realizar otro tipo de comunicaciones de la butterfly hacia la LPCXpresso LPC1769 como son SPI e I<sup>2</sup>C, que son muy usadas para este tipo de proyectos.

# **CAPÍTULO 1**

## **1. Descripción General del Proyecto**

### **1.1 Antecedentes**

En el mercado [1] se puede encontrar muchos tipos de motores BLDC y una amplia gama de equipos para su respectivo funcionamiento y diferentes tipos de control.

Por lo que para nuestro proyecto implementaremos el control de un motor BLDC con sensores mediante la tarjeta microcontroladora LPCXpresso LPC1114, la cual es comandada por periféricos como lo son un joystick contenidos en la tarjeta Butterfly.

Los microcontroladores se programan con los software's indicados que se detallarán en los próximos capítulos. Con la LPCXpresso v4.1.5\_219 se

programará la tarjeta LPCXpressoLPC1114 que pertenece al kit del “LPCXpresso Motor Control Kit” en la cual va la programación principal del control del motor BLDC, y con el AVR STUDIO 4 se programará la tarjeta Butterfly que contiene al joystick que se comandará al sistema.

## **1.2 Motivación**

Nuestra motivación viene de la necesidad de optimizar el rendimiento de la energía eléctrica y mejorar las aplicaciones de la automatización en la industria, ya que lo que más se utilizan son los motores; por ende es prioritario aplicar la nueva tecnología en motores BLDC y aprender su funcionamiento y nuevas aplicaciones.

Y es por esto que los motores BLDC han ganado popularidad en la industria ya que han podido llegar a superar la eficiencia y su rendimiento con respecto a otro tipo de motores, así como su tiempo de vida útil, esto sin contar que no necesitan de mucho mantenimiento, entre otras ventajas que se detallarán más adelante en este trabajo.

### **1.3 Identificación del Problema**

La creciente demanda de mejoras de la automatización en la industria requiere de mayor consumo de energía tanto por parte de los motores como por parte de los equipos electrónicos que los comandan, tales como drivers, variadores, controladores, arrancadores, etc.

Es por eso los fabricantes han optimizado y mejorado el diseño, fabricación y funcionamiento de los motores BLDC; así como también han mejorado los equipos y sistemas de control para estos.

Es por eso que hemos tomado la opción que nos da la empresa NXP con sus microcontroladores CórteX, los cuales se encuentran entre sus productos las tarjetas LPCXpresso. Así, nos dedicaremos a la implementación del control de un motor BLDC de bajo voltaje con un microcontrolador de última generación [2].

### **1.4 Objetivos Principales**

Entre los objetivos de este trabajo tenemos:

- Aprender las características, composición y el funcionamiento del motor BLDC.
- Programar el control del motor en la tarjeta LPCXpressoLPC1114.
- Aprender el funcionamiento del Motor Control Kit de NXP.

- Programar el mando del motor (encendido, apagado y cambio de velocidades) en la tarjeta LPCXpressoLPC1769.
- Identificar las entradas y salidas del joystick y display para implementar su funcionamiento con otros periféricos externos como la tarjeta Butterfly.
- Implementar el prototipo de un control de velocidad para un motor BLDC con los materiales ya descritos.

### **1.5 Limitaciones del Proyecto**

Entre las limitaciones del proyecto tenemos las siguientes:

- Se trata de un prototipo.
- Por lo tanto no se puede comercializar o aplicar directamente a un sistema.
- El espacio que ocupa es demasiado grande con respecto a otros sistemas.

## 1.6 Descripción de las Partes del Proyecto

El proyecto comprende dos partes, una física y la parte de software.

### 1.6.1 Composición Física:

En la Figura 1.1 se muestra el Diagrama de bloques que comprende las partes físicas del proyecto como lo son las tarjetas y el motor.

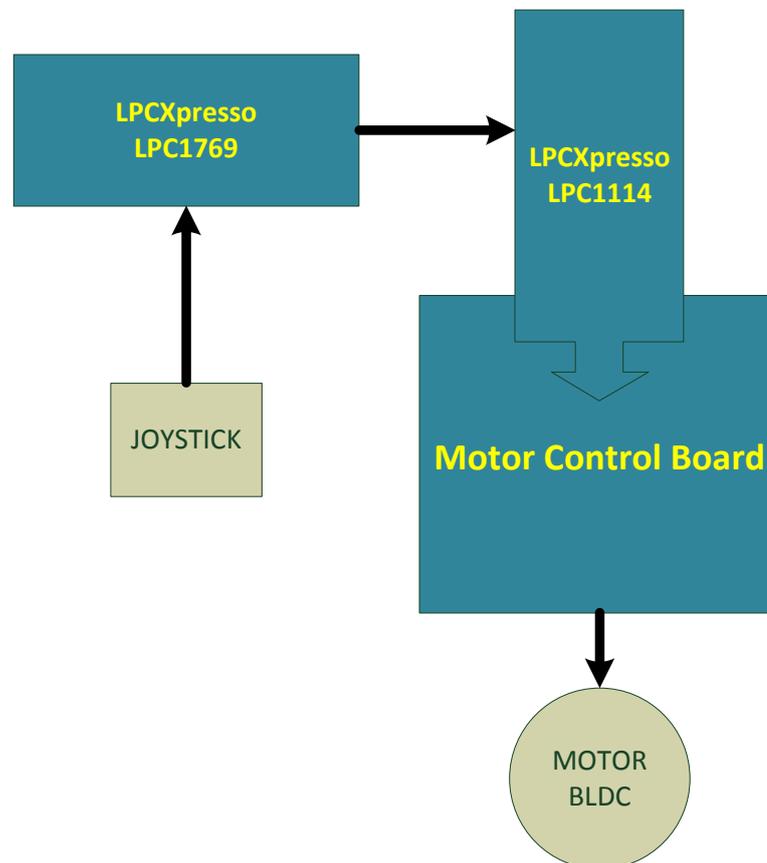


Figura 1.1: Diagrama de Bloques del Proyecto

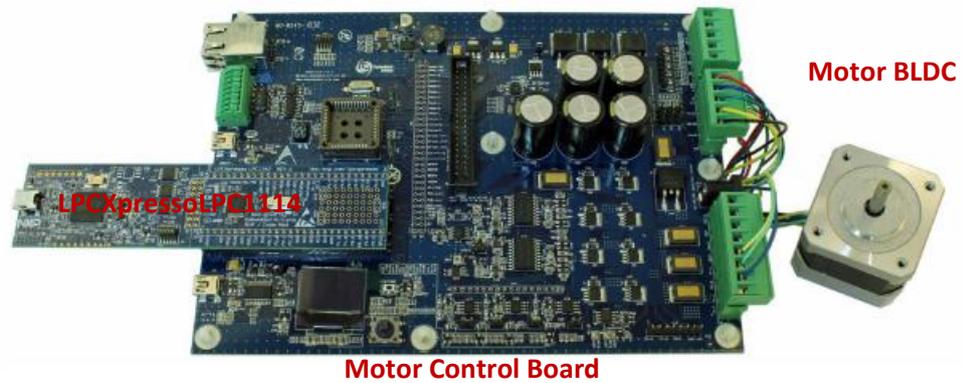


Figura 1.2: Motor Control Kit

### 1.6.2 Software

Utilizaremos, como habíamos especificado, el software LPCXpresso v4.1.5\_219 que se detallará mas adelante para la programación de la tarjeta LPCXpresso LPC1769 y AVR STUDIO 4 para la programación de la tarjeta butterfly.

## **CAPÍTULO 2**

### **2. Sustentación Teórica**

#### **2.1 Motores BLDC**

##### **2.1.1 Principio de funcionamiento del motor BLDC**

El motor BLDC es un motor sincrónico, que puede ser modelado como un motor BDC. La conmutación de los motores BLDC se realiza a través de un sistema de control electrónico en comparación con la conmutación mecánica de los motores BDC [3]. Los motores BLDC consisten de un estator devanado y un rotor de imán permanente y de esta forma evitamos el problema de como transferir la corriente a una armadura en movimiento mientras que un motor BDC consiste de un estator de imán permanente y un rotor devanado [4].

El circuito equivalente de un motor DC es como se muestra en la figura 2.1.

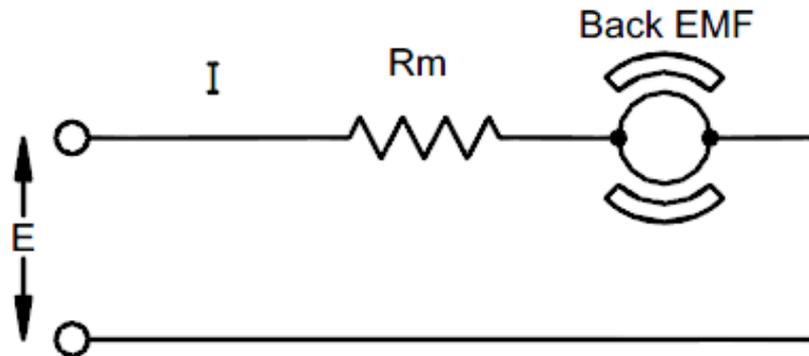


Figura 2.1: Circuito Equivalente de un Motor DC

Si se aplica un voltaje a través de los terminales del motor, una corriente  $I$ , circulara a través de las bobinas de resistencia  $R_M$ , el motor generara una fuerza contra-electromotriz (EMF). Esta fuerza contra-electromotriz es proporcional a la velocidad de operación  $w$ , por una constante  $K_b$ .

$$FECM = w * K_b \quad (2.1)$$

Y es directamente opuesta al voltaje aplicado en los terminales del motor. La ecuación que describe el circuito mostrado en la figura 2.2 es:

$$E = V_{fuente} = I * R_M + w * K_b \quad (2.2)$$

En condiciones iniciales, cuando  $\omega = 0 \text{ rpm}$  y no hay presencia de fuerza contra-electromotriz. La ecuación 2 se reduce a  $I = V_{fuente}/R_M$ . Como la velocidad establece el incremento de la fuerza contra-electromotriz. Entonces la ecuación queda:  $V_{fuente} = I * R_M + w * K_b$  y despejando de la ecuación I, obtenemos que:

$$I = \frac{(V_{fuente} - \omega K_b)}{R_M} \quad (2.3)$$

A una determinada velocidad y bajo condiciones sin carga. La magnitud de la fuerza contra-electromotriz llegara a ser igual a la magnitud de la fuente de voltaje. En este punto de operación, el numerador de la ecuación (3) será 0, dejando sin voltaje disponible para que produzca una corriente. La velocidad del motor observada en este punto es referida como la velocidad "sin carga"  $\omega_{NL}$  en un motor con resistencia de bobinado  $R_M$ .

Los motores BLDC tienen mejor Velocidad vs Torque en la Figura 2.2 se muestra dicha curva esta curva es basándose en un motor de imán permanente ideal. La velocidad y el torque de un motor BLDC depende de la fuerza del campo magnético generado por las bobinas cuando estas son energizadas, la cual depende de la corriente que atraviesa dichas bobinas.

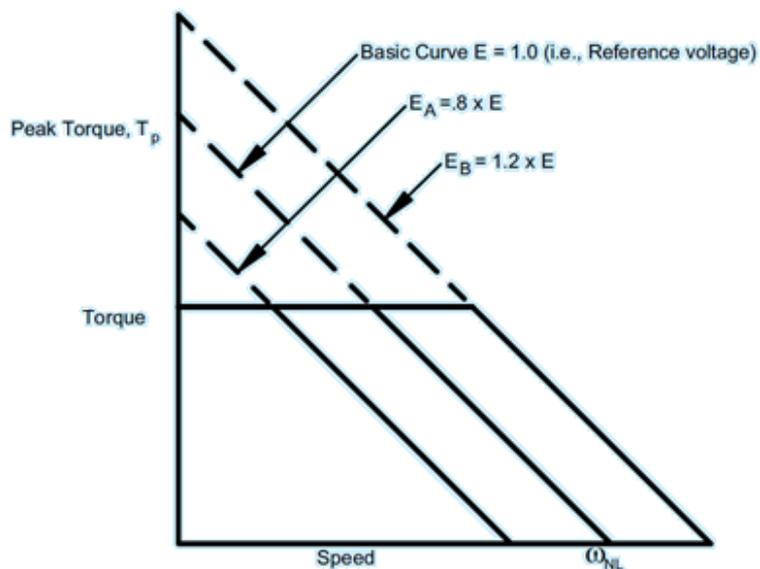


Figura 2.2: Curva Torque-Velocidad de un Motor DC

Los motores BLDC también llamados Motores Sincrónicos DC de Imán Permanente tienen muchas ventajas sobre los motores BDC y motores de inducción. Cuando nosotros hablamos de la funciones eléctricas de un motor, no debemos deberíamos olvidar la importancia del devanado y la conmutación del motor. La conmutación se refiere al proceso por el cual convertimos la corriente directa de entrada a corriente alterna y propiamente distribuida a cada uno de los devanados en la armadura.

En motor DC convencional, la conmutación se lleva acabo por unas escobillas y un colector en contraste con un motor BLDC [5]. En la Tabla 2.1 detallamos las ventajas del motor BLDC sobre los motores DC convencionales.

	<b>Motor DC</b>	<b>Motor BLDC</b>
Estructura Mecánica	Estator de imán permanente	Rotor de imán permanente
Mantenimiento	Mayor mantenimiento y más revisiones periódicas.	Poco mantenimiento y pocas revisiones.
Durabilidad	Moderada.	Larga vida útil.
Curva Velocidad vs Par	Moderada. A altas velocidades la fricción de las escobillas se incrementa, reduciendo el par.	Plana. Operación a todas las velocidades con la carga definida.
Conmutación	Conmutación mecánica por contacto entre las escobillas y el colector.	Conmutación electrónica usando transistores.
Detección de la posición del rotor	Detección automática por las escobillas.	Sensores Hall, codificadores ópticos, etc.
Ruido eléctrico	Moderado. Debido al arco voltaico que se produce entre las escobillas y el colector	Bajo ruido. Debido a que no hay escobillas ni colector.
Inversión de giro del rotor	Se invierten los terminales de alimentación.	Reordenando la secuencia lógica.
Inercia del rotor	Alta. Limita las características dinámicas.	Baja. Debido al rotor de imán permanente.
Conexión del bobinado	Se lo conecta en delta siendo esta la más utilizada.	Se puede conectar su bobinado en estrella y en delta.

Tabla 1: Comparación de un motor DC con un motor BLDC

### 2.1.2 Sensores efecto Hall

En este proyecto se presenta la utilización de un motor BLDC con sensores de tipo efecto Hall, este tipo de sensor es muy usado debido a su relativo bajo costo.

Este tipo de dispositivos como su nombre lo indica están basados en la teoría del Efecto Hall, esta teoría nos dice que, cuando se coloca un conductor por el cual pasa una corriente en presencia de un campo magnético perpendicular a la dirección de la corriente, se genera una diferencia de potencial en una dirección perpendicular tanto a la corriente como al campo magnético, a este fenómeno se lo conoce como *efecto Hall* en honor a quien observo el fenómeno por primera vez Edwin Hall. [7]

Los motores BLCD como hemos mencionado tiene un rotor de imán permanente con un bobinado trifásico en el estator, la conmutación de un motor BLCD es controlado electrónicamente [8].

Para poder hacer girar el rotor del motor BLCD los devanados del estator deben ser energizados en una secuencia. Es importante saber la posición del rotor para saber cual devanado será energizado siguiendo la secuencia de energización. La posición del rotor es detectada usando los sensores de efecto Hall acoplados en el estator.

Muchos de los motor BLCD tienen tres sensores de efecto Hall dentro del estator en un extremo de no conducción del motor.

Siempre que los polos del rotor pasen cerca de los sensores ellos entregaran una señal alta o baja indicando así el polo norte o sur que esta pasando en ese momento por el sensor. Por las señales entregadas por los sensores Hall se puede determinar la secuencia exacta de conmutación.

La Figura 2.3 muestra un corte transversal realizado a un motor BLCD con un rotor que tiene alternado los polos N y S. Los sensores Hall son incrustados en la parte interna del estator en el motor. El introducir los sensores dentro del estator es un proceso complejo, ya que cualquier desalineación de estos sensores Hall con respecto al rotor generaría un error al momento de determinar la posición correcta del rotor.

Para simplificar el montaje sobre de los sensores Hall sobre el estator algunos motores pueden tener los sensores Hall sobre un pequeño rotor adicional al rotor principal. Cada vez que el rotor gira los sensores Hall dan el mismo efecto como si estuvieran colocados en el rotor principal.

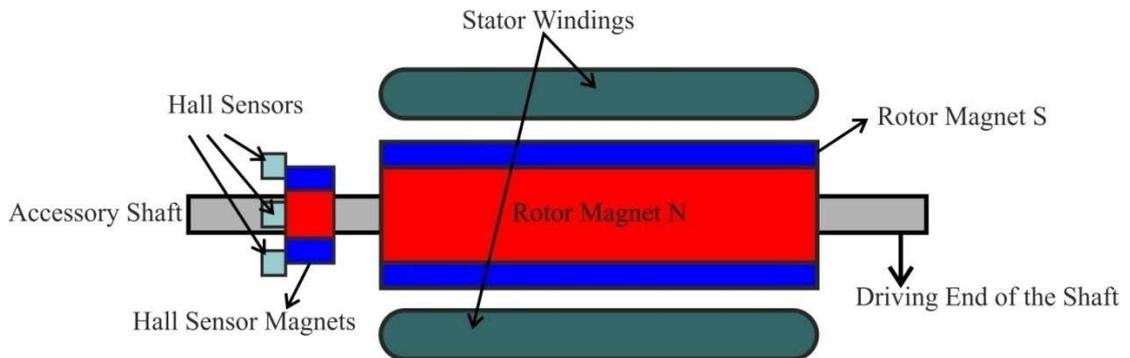


Figura 2.3: Corte Transversal de un Motor BLCD

Típicamente los motores BLDC tienen tres devanados que pueden ser conectados en estrella o delta y este tipo de motor necesita tres señales con un desfase de 120 grados para control, por lo que una regulación en bucle cerrado puede ser utilizada para mejorar el rendimiento del motor.

### 2.1.3 Rotación

En los motores BLDC los devanados son distribuidos en forma trapezoidal para generar una onda trapezoidal de la fuerza contra electromotriz.

Las técnicas de conmutación generalmente usadas son trapezoidal o llamadas conmutación de bloques, donde solamente dos fases estarán conduciendo en alguna parte del tiempo. Un método alternativo para la conmutación del motor es la llamada conmutación sinusoidal en la cual todas

las tres fases estarán conduciendo en algún tiempo dado. Los motores PMSM también indistintamente llamados como motores BLCD el cual tiene sus devanados distribuidos de forma sinusoidal de acuerdo para este tipo de conmutación. El par generado por motores PMSM es suave en comparación con los motores BLDC ya que su par tiene más ondulaciones. Pero el par máximo desarrollado por los motores PMSM es menor en comparación con el desarrollado por los motores BLCD.

#### **2.1.3.1 Control del motor BLCD usando sensores Hall**

El método de conmutación trapezoidal es el más simple para controlar el motor BLCD y fácil también al momento de realizar la implementación de este tipo de control. Para conmutación correcta y para la rotación del motor, la información de la posición del rotor es muy crucial. Solamente con la información de la posición del rotor, la conmutación electrónica en el puente inversor se estará cambiando de apagado a encendido y viceversa para asegurar la dirección correcta del flujo de corriente en las bobinas respectivas. Sensores de efecto Hall, se utilizan en general como sensores de posición para la conmutación trapezoidal.

Para obtener el Par máximo, el flujo del estator y el flujo del rotor deben de estar lo más cerca posible a  $90^\circ$  y para ello se debe alimentar los devanados de la mejor forma, por lo tanto el controlador necesita algún medio para determinar la orientación del rotor / posición (con respecto a las bobinas del

estator), tales como sensores de efecto Hall, que están montados en o cerca de entrehierro de la máquina para detectar el campo magnético de los imanes que pasan del rotor. Cada sensor emite un alto nivel de  $180^\circ$  de una rotación eléctrica, y un nivel bajo para los otros  $180^\circ$ . Los tres sensores tienen un desplazamiento relativo  $60^\circ$  el uno del otro. Esto divide una rotación en seis fases.

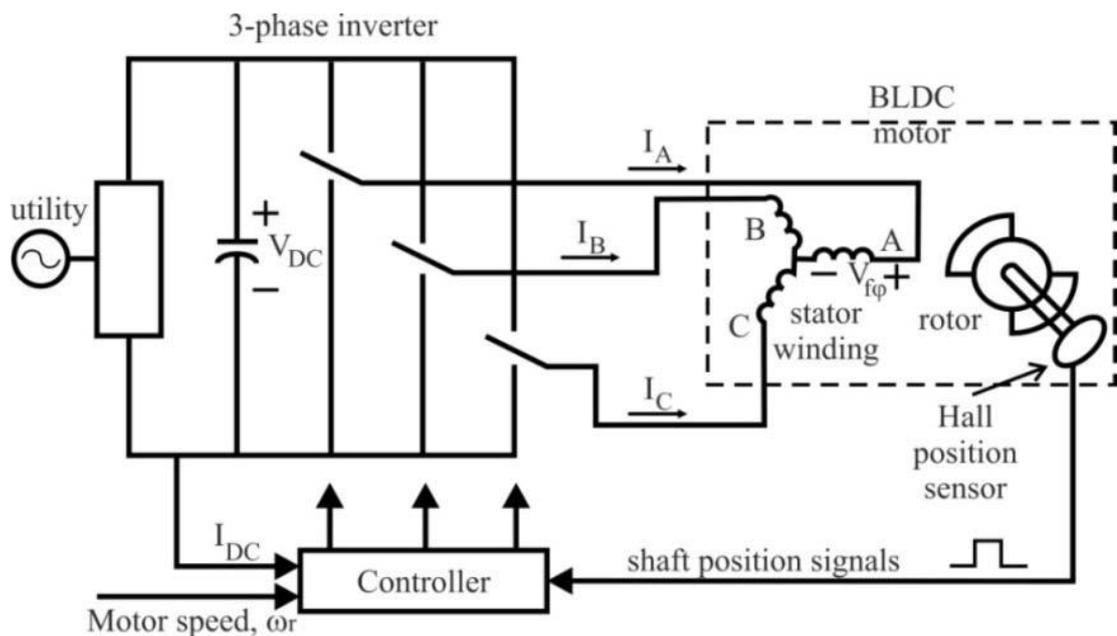


Figura 2.4: Conmutación Electrónica de un motor BLCD

El proceso de cambiar la corriente fluye a través de sólo dos devanados en el estator para cada rotación de  $60^\circ$  eléctricos del rotor se denomina conmutación electrónica. El motor es alimentado desde un inversor trifásico, y conmutación puede ser simplemente desencadenada por el uso de señales de sensores de posición que se montan en los puntos apropiados en todo el estator. Cuando se monta a intervalos de  $60^\circ$  eléctricos y alineados con los bobinados del estator de fase estos interruptores Hall entregan pulsos digitales que pueden ser decodificados en la secuencia de conmutación deseada de tres fases. Un motor BLDC con un inversor de seis pasos y sensores Hall de posición se muestra en la Figura 2.4.

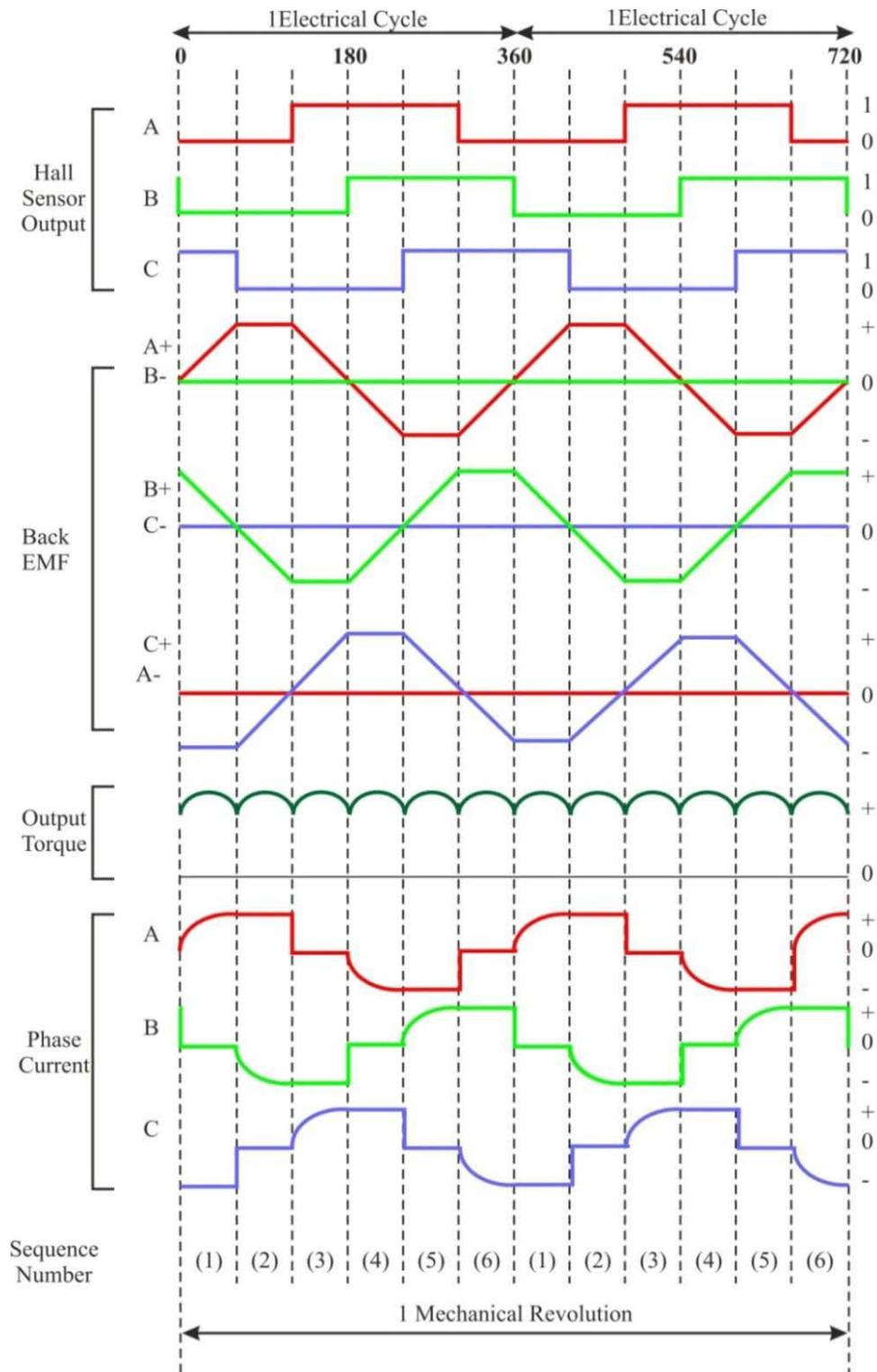


Figura 2.5: Curvas de Voltaje de las fases de un motor BLDC

Tal unidad generalmente también tiene un bucle de corriente para regular la corriente del estator, y un bucle de velocidad externa para control de velocidad. La velocidad del motor se puede controlar si el voltaje a través del motor se cambia, lo que puede lograrse fácilmente variando el ciclo de trabajo de la señal PWM para controlar los seis conmutadores del puente de tres fases.

Sólo dos interruptores del inversor, uno en el banco inversor superior y uno en el banco inversor inferior, están llevando a cabo en cualquier instante. Estos eventos de conmutación discretas aseguran que la secuencia de la realización de pares de terminales del estator se mantiene.

La figura 2.5 muestra un ejemplo de las señales del sensor Hall con respecto a la espalda-CEM y la fase actual. Uno de los sensores Hall cambia el estado cada 60 grados eléctricos de rotación. Teniendo en cuenta esto, se necesitan seis pasos para completar un ciclo eléctrico. Sin embargo, un ciclo eléctrico no puede corresponder a una revolución completa mecánica del rotor. El número de ciclos eléctricos para ser repetidos para completar una rotación mecánica está determinado por los pares de polos del rotor. Para cada par de polos del rotor, un ciclo eléctrico se ha completado. El número de ciclos eléctricos o rotaciones es igual a los pares de polos del rotor [9]. Esta secuencia de la realización de pares es esencial para la producción de un par de salida constante.

Número de revoluciones eléctricas = Número de polos \* Número de revoluciones mecánicas

En resumen, las unidades de motor de imanes permanentes requieren un sensor de posición del rotor para realizar adecuadamente la conmutación de fase, pero hay varios inconvenientes cuando tales tipos de sensores de posición se utilizan. Los principales inconvenientes son el coste incrementado y el tamaño del motor, y una disposición especial debe ser hecha para el montaje de los sensores. Además, sensores Hall son sensibles a la temperatura y por lo tanto el funcionamiento del motor está limitado, en la figura 2.6 se muestra la estructura interna del motor BLCD, lo que podría reducir la fiabilidad del sistema debido a los componentes adicionales y el cableado. Para reducir costes y mejorar la fiabilidad tales sensores de posición puede ser eliminado. Para este fin, muchos esquemas sensorless se ha informado de la posición (y velocidad) de control de motores BLDC.

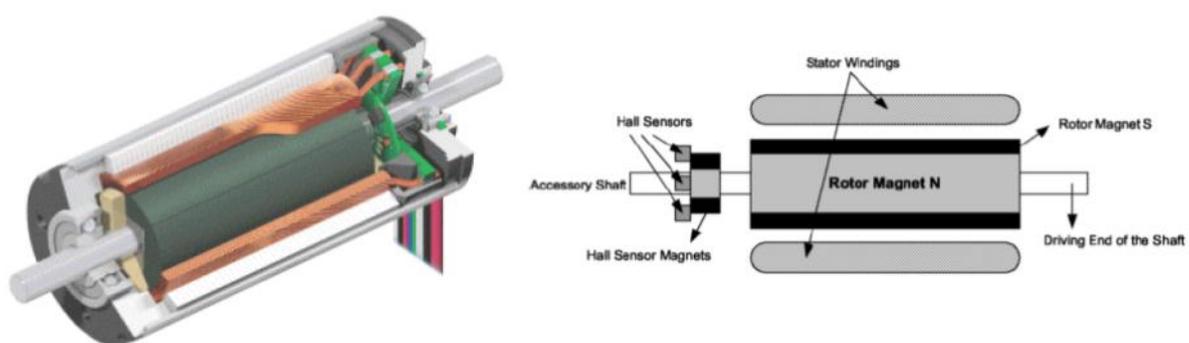


Figura 2.6: Constitución interna de un motor BLDC

## **2.2 LPCXpressoLPC1769**

### **2.2.1 Introducción**

LPCXpresso es una nueva, económica plataforma de desarrollo disponible de NXP. El software consiste de un mejorado IDE basado en Eclipse, un compilador de C de GNU, bibliotecas, y un depurador de GDB reforzado. El hardware consiste en la LPCXpresso Tarjeta de Desarrollo que tiene un LPC-link ponga a punto interfaz y un NXP microcontrolador LPC ARM. LPCXpresso es una solución que les permite a los ingenieros desarrollar sus aplicaciones.

El LPCXpresso IDE, producido por Code Red Technologies ([www.code-rojo-tech.com/lpcxpresso /](http://www.code-rojo-tech.com/lpcxpresso/)), es basado en la plataforma de desarrollo de Eclipse popular e incluye varias mejoras LPC-específicas. Es un toolchain del GNU industria-normal con una biblioteca del C perfeccionada que da diseña todas las herramientas necesario desarrollar las soluciones del software de calidad superior rápidamente y rentablemente. El ambiente de programación en C incluye todas las características de niveles profesionales.

El LPCXpresso la tarjeta designada, juntamente desarrollada por NXP, el Code Red Technologies, y Embedded Artists, incluye un depurador de JTAG integrado (el LPC-link. La porción designada de la tarjeta puede conectar a las tarjetas de la expansión para proporcionar una variedad mayor de

interfaces, y dispositivos de I/O. El depurador del LPC-link en la tarjeta proporciona un USB de gran velocidad a JTAG/SWD una al IDE y puede conectarse a otro ponga a punto los blancos como un prototipo del cliente.

### **2.3 LPCXpresso v4.1.5\_219**

El IDE de LPCXpresso es un ambiente de desarrollo de software favorablemente integrado para NXP LPC Microcontroladores que incluye todas las herramientas necesario desarrollar las soluciones de software de calidad altas en un oportuno y costo la moda eficaz. LPCXpresso es basado en Eclipse con muchos LPC las mejoras específicas. También ofrece la última versión de la industria la cadena de herramienta de GNU normal con una propiedad perfeccionada que biblioteca del C que proporciona la calidad profesional labra con herramienta al costo bajo. El LPCXpresso IDE puede construir un ejecutable de cualquier tamaño con la optimización del código llena y apoya un transmite límite de 128 kB después de la registración. LPCXpresso apoya el ciclo de plan de producto incluido lleno moviendo más allá de la evaluación de la astilla aborda y desarrollo de apoyo en las tarjetas del blanco externas (Figura 2.7:7).

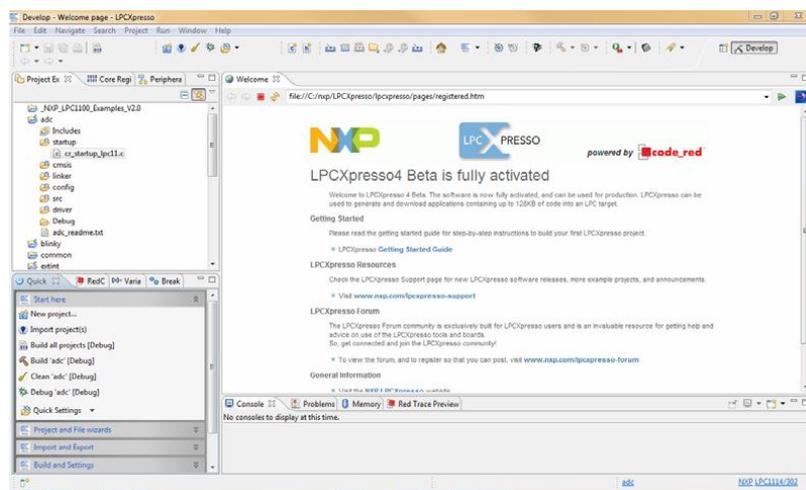


Figura 2.7: LPCXpresso IDE

### 2.3.1 LPC-Link el depurador de JTAG/SWD

La tarjeta de LPCXpresso contiene un depurador de JTAG/SWD llamado el “LPC-Link” y un MCU designado. LPC-Link está provisto con un 10-pin el título de JTAG (resaltó en la imagen anterior) y conecta transparentemente al por designado USB (los USB unen y otro ponga a punto los características son proporcionados por el ARM9 de NXP basó LPC3154 MCU). Cortando las huellas entre LPC-Link y el blanco hará un depurador de JTAG autosuficiente al LPC-eslabón. Esto permite conectar la plataforma de LPCXpresso a un blanco externo y desarrollaba para una variedad ancha de NXP Corteza-M0, Corteza-M3, y ARM7/9 basó las aplicaciones. Los productos del microcontrolador actualmente apoyados incluyen LPC1700, LPC1300,

LPC1200, y serie de LPC1100 y miembros específicos de los LPC2000 y familias de LPC3000.

### **2.3.2 Los requisitos del sistema**

El Sistema operativo

Microsoft® Windows - el 32-bits de XP (SP2 o mayor) Microsoft® Windows - 32-bits de la Vista o 64-bits Microsoft® Windows - Windows 7 32-bitso o de 64-bits y para Linux en sus diferentes distribuciones.

La RAM del sistema: 512 mínimos de MB (1 GB recomendado).

El Disco duro: 300 MB de espacio disponible.

El Adaptador de Screen/Display: El mínimo del 1024x768 recomendado.

### **2.3.3 La Conexión de Internet**

El internet de gran velocidad se recomienda transmitir y registrar el software

Nota: LPCXpresso puede instalar y puede correr en otras distribuciones de Linux. Sin embargo, se han probado sólo las distribuciones listadas sobre. La

virtualización del escritorio labra con herramienta apoyando a un Linux o invitado de Windows con el apoyo de USB puede usarse para ejecutar LPCXpresso en otras plataformas del software.

### **2.3.4 El proceso de la instalación**

LPCXpresso se instala en un solo directorio, de su opción. Pueden instalarse las versiones múltiples simultáneamente sin cualquier problema. El proceso de la instalación es doble click el botón el archivo del instalador después de transmitir. Entonces haga clic “próximo” en el wizard del arreglo. Para instalar bajo el linux, el archivo transmitido debe marcarse como el chmod primero usando ejecutable +r. - el bits Microsoft® Windows - Windows 7 32-bits o 64-bits

Linux - Ubuntu 9, 10, y 11

Linux - 12 y 13

La RAM del sistema: 512 MB mínimos (1 GB recomendado)

El Disco duro: 300+ MB de espacio disponible.

El Adaptador de Screen/Display: El mínimo del 1024x768 recomendado

La Conexión de Internet:

El internet de gran velocidad se recomienda transmitir y registrar el software.

Nota: LPCXpresso puede instalar y puede correr en otras distribuciones de Linux. Sin embargo, se han probado sólo las distribuciones listadas. El virtualizador del escritorio labra con herramienta apoyando a un linux o invitado de Windows con el apoyo de USB puede usarse para ejecutar LPCXpresso en otras plataformas del software.

LPCXpresso se instala en un solo directorio, de su opción. Pueden instalarse las versiones múltiples simultáneamente sin cualquier problema. El proceso de la instalación es doble click el botón el archivo del instalador después de transmitir. Entonces haga clic “próximo” en el wizard del arreglo. Para instalar bajo el linux, el archivo transmitido debe marcarse como el chmod primero usando ejecutable +r.

## **2.4 LPCXpresso Motor Control Kit**

Este sistema le hace posible para usted familiarizarse inmediatamente con el prototipo de mando de motor. Es una plataforma universal para voltaje bajo para el mando de motor. Con esta plataforma es posible controlar BLDC, BLAC, stepper y motores de DC con escobillas. La tarjeta se ha diseñado para evaluación y no para aplicación final.

En la figura 2.8 se muestra el LPCXpresso Motor Control Kit para el desarrollo de este proyecto.

#### **2.4.1 Lista de Elementos:**

- Socket para LPCXpresso LPC1114, LPC11C24 y LPC1343
- Socket para LPCXpresso LPC176x
- Socket para LPC1xxx en PLCC44
- Conector 4 fases (basado en NXP PMSN2R6-40YS NMOSFET).
- Salida de Voltaje (en tres fases y la tierra virtual)
- Conector para sensores Hall & QEI.
- Sensor de Temperatura
- 15W suministro de poder (+11V, +5V, +3.3V) en la tarjeta.
- Interface de Comunicación
- Conexión USB
- Conexión Ethernet.
- Interfaz de RS422/485
- Interfaz UART-a-USB
- Joystick

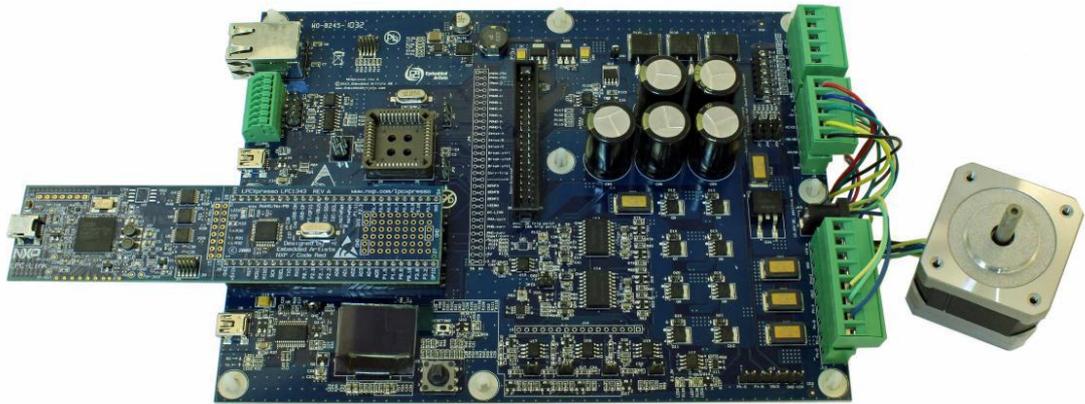


Figura 2.8: Motor Control Kit

## 2.5 LPCXpressoLPC1114

Los LPC111x/LPC11Cxx son un Basado en ARM Cortex\_M0 32, económico la familia de MCU, diseñado para las aplicaciones de microcontroller de 8/16-bits, mientras ofreciendo la actuación, de bajo consumo, simple, la instrucción y memoria que se dirige junto con el tamaño del código reducido comparó a las arquitecturas del 8/16-bits existentes. Las LPC111x/LPC11Cxx operan a las frecuencias de CPU de a 50 MHz.

El complemento periférico del LPC111x/LPC11Cxx incluye 32 kB de memoria, a 8 kB de memoria de los datos, un director de C\_CAN (LPC11Cxx), un Modo rápido Más la interfaz del I2C-autobús, un RS-485/EIA-485 UART, a dos SPI una con SSP. Los características, cuatro

cronómetros del propósito generales, un 10-bits ADC, ya 42 propósito general I/O

Los drivers de C\_CAN y En-sistema de llamada que Programan las herramientas vía C\_CAN son incluido en el LPC11Cxx. Además, parte LPC11C2x están provistos con una LATA del en-astilla el trans-recibidor.

Comentario: Este usuario las tapas manuales las partes de LPC111x/LPC11Cxx. La serie consiste de la serie de LPC1100 (parte LPC111x/101/201/301), la serie de LPC1100L (las partes LPC111x/102/202/302), y la serie de LPC11C00 (parte LPC11C1x/301 y LPC11C2x/301). Los LPC1100L incluyen el poder perfila y los LPC11C00 incluyen el Director de C\_CAN y en-astilla ENLATAN a chóferes.

Las mejoras de LPC111x/LPC11Cxx

La serie Ofrece la apreciación global

La serie de LPC1100 I2C, SSP, UART, GPIO,

- Los Cronómetros y watchdog timer.
- El 10-bits de ADC
- La memoria de Flash/SRAM.

Serie de LPC1100L que la serie de LPC1100 ofrece más las características adicionales siguientes:

- Power perfila con el más bajo consumo de poder en los modos Activo y Suspensión.
- Que las resistencias pull-up Interiores tiran a los pines para abatanar el nivel de VDD.
- El modo del abrir-desagüe pseudo Programable para los pines de GPIO.
- WWDT con la capacidad reloj fuente cerradura.

Serie de LPC11C00 que la serie de LPC1100 ofrece más las características adicionales siguientes:

- ENLATA a director.
- En-astilla LATA chóferes.

El • En-astilla LATA trans-recibidor (LPC11C2x).

- WDT (no el windowed) con la capacidad reloj fuente cerradura.

### 2.5.1 LPCXpresso Motor Control Board

Esquemáticos: Ver Anexos.

## 2.6 Tarjeta AvrButterfly

### 2.6.1 Características Generales de la Placa de Desarrollo AvrButterfly

El kit AvrButterfly expone las siguientes características principales:

- La arquitectura AVR en general y el microcontrolador ATmega169 en particular.
- Diseño de bajo consumo de energía.
- El encapsulado tipo MLF.
- Periféricos:
  - ❖ Controlador LCD
  - ❖ Memorias:
    - Flash, EEPROM, SRAM.
    - DataFlash 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 por Ancho de Pulso (PWM).

En la siguiente figura 2.9 se muestra el diagrama de bloques de la placa de desarrollo AvrButterfly. La tarjeta AvrButterfly esta proyectado para el desarrollo de aplicaciones con el ATmega169 y además puede usarse como un modulo dentro de otros productos.

El AvrButterfly utiliza el microcontrolador AVR ATmega169V, que combina la tecnología Flash con el más avanzado y versátil micro-controlador de 8 bits disponible.

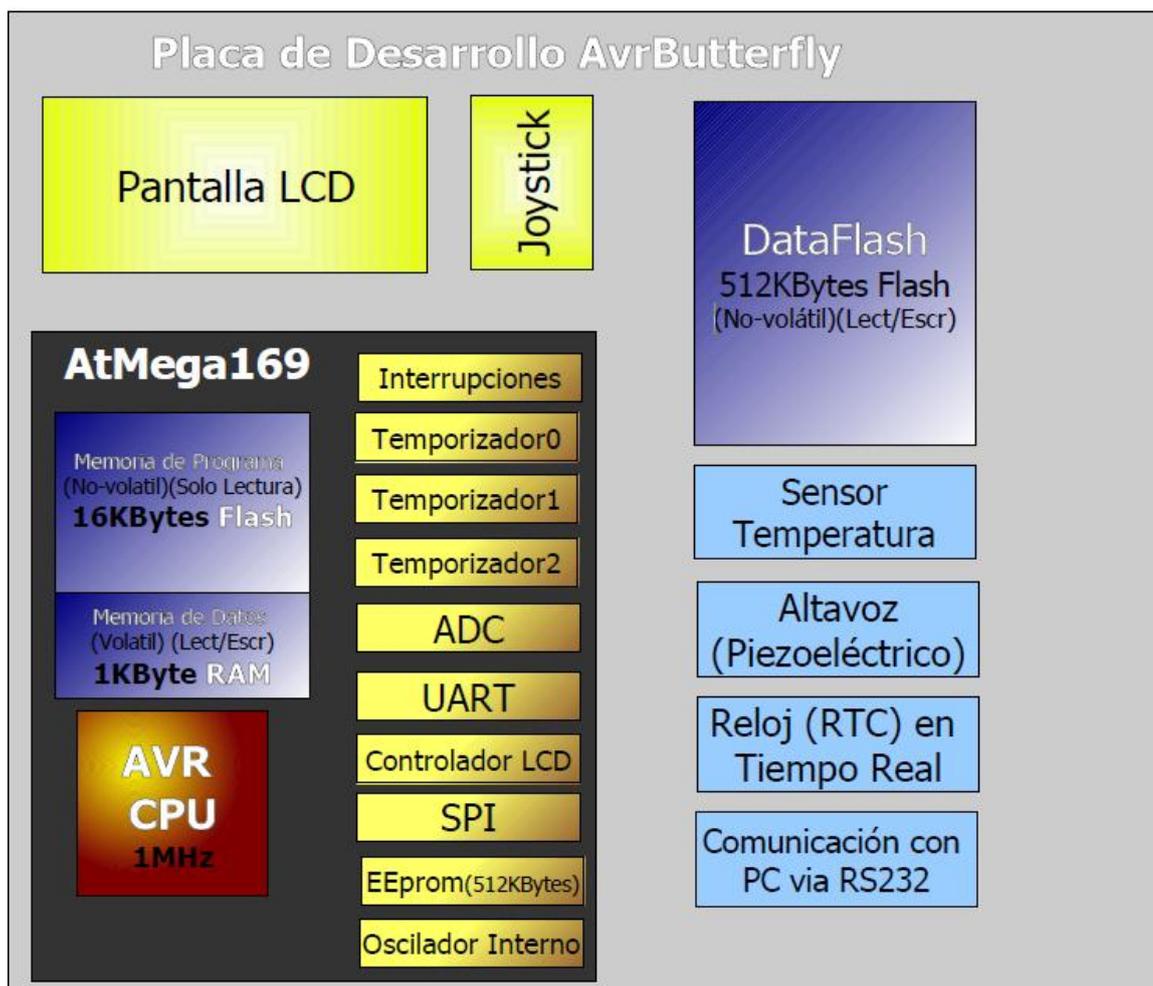


Figura 2.9: Diagrama de Bloques de la Tarjeta AvrButterfly

En la figura 2.10 Se puede observar que la tarjeta cuenta con una serie de periféricos internos y externos; y para cada uno de ellos se ha de hacer una rutina de inicialización y otras para el uso que se lesee dar.

En algunos casos es necesario utilizar u periférico interno para poder utilizar uno externo, por ejemplo:

Para utilizar el altavoz (piezoeléctrico) y generar un tono de sonido a una determinada frecuencia es necesario utilizar el **Temporizador 1** como generador PWM.

El sensor de temperatura una vez alimentado genera una señal analógica que hay que capturarla con el **convertidor analógico digital (ADC)** previamente configurado para tal efecto.

Para poder almacenar y restaurar datos del dispositivo de memoria DataFlash es necesario comunicarlo con el ATmega169 a través del periférico de comunicación **Serial Peripheral Interface (SPI)**.

Para el funcionamiento adecuado de la **Pantalla LCD**, es necesario configurar el periférico **Driver LCD**, que genera las señales adecuadas para cada segmento LCD de dicha pantalla.

Para usar el **Joystick** de cinco funciones, 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.

Para mas detalle de la distribución de los periféricos internos se muestra la figura 2 en la que podemos observar los periféricos mencionados hay más periféricos, lo que ocurre es que algunos de ellos en la tarjeta AvrButterfly tiene los pines de salida utilizados para otras funcionalidades, y por ello estos

no se pueden usar como es el caso del comparador analógico, cuya salida es utilizada por una de los pines del LCD.

La distribución de los periféricos más relevantes es como la que se muestra en la figura 2.10 donde poder ver donde se encuentra el microcontrolador ATmega169, los pines para la conexión UART, ISP, etc.

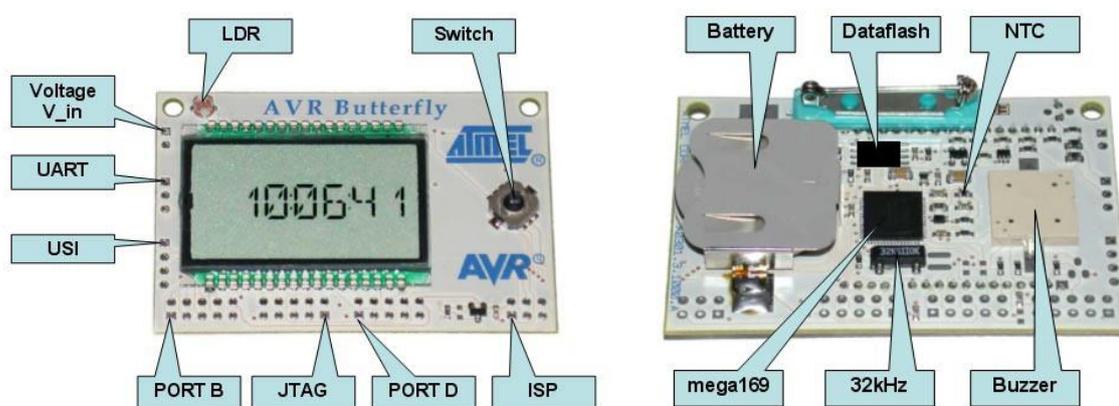


Figura 2.10: Periféricos externos de la Tarjeta AvrButterfly.

En la siguiente figura 2.11 se muestra las conexiones a los diferentes pines de los puertos del microcontrolador ATmega169 que gobierna la placa.

La placa de desarrollo AvrButterfly cuenta con dos circuitos integrados con capacidad de almacenamiento de datos, que son el microcontrolador ATmega169 y la DataFlash.

En la siguiente tabla indicamos cuales son los periféricos más significativos de la Tarjeta Butterfly:

<b>NTC</b>	Resistencia negativa de temperatura (Sensores de Temperatura).
<b>LDR</b>	Resistencia Dependiente de la Intensidad de la Luz.
<b>SWITCH</b>	Interruptor joystick de cinco posiciones.
<b>32 KHZ</b>	Oscilador de cuarzo para generación de onda de reloj de tiempo real.
<b>BUZZER</b>	Piezoeléctrico "altavoz" para la emisión de sonidos de un solo tono.
<b>DATAFLASH</b>	Memoria DataFlash de 512 Kbyte de capacidad de almacenamiento.
<b>UART</b>	Pines de salida directa al puerto RS232 del ordenador.

Tabla 2: Descripción de periféricos de la tarjeta Butterfly

El microcontrolador ATmega169 cuenta, a su vez con tres memorias integradas en su encapsulado. Una memoria no volátil Flash auto-programable de 16 Kbyte de capacidad destinada al almacenamiento del código de la aplicación que diseñemos y las constantes y cadenas de caracteres constantes.

Esta memoria hace las veces de memoria ROM de programa, ya que solo puede escribirse en ella cuando el programa no esta funcionando (debido a que precisamente lo que almacena es el programa principal, mediante un pequeño programa cargador de arranque que no se puede borrar llamado Bootloader).

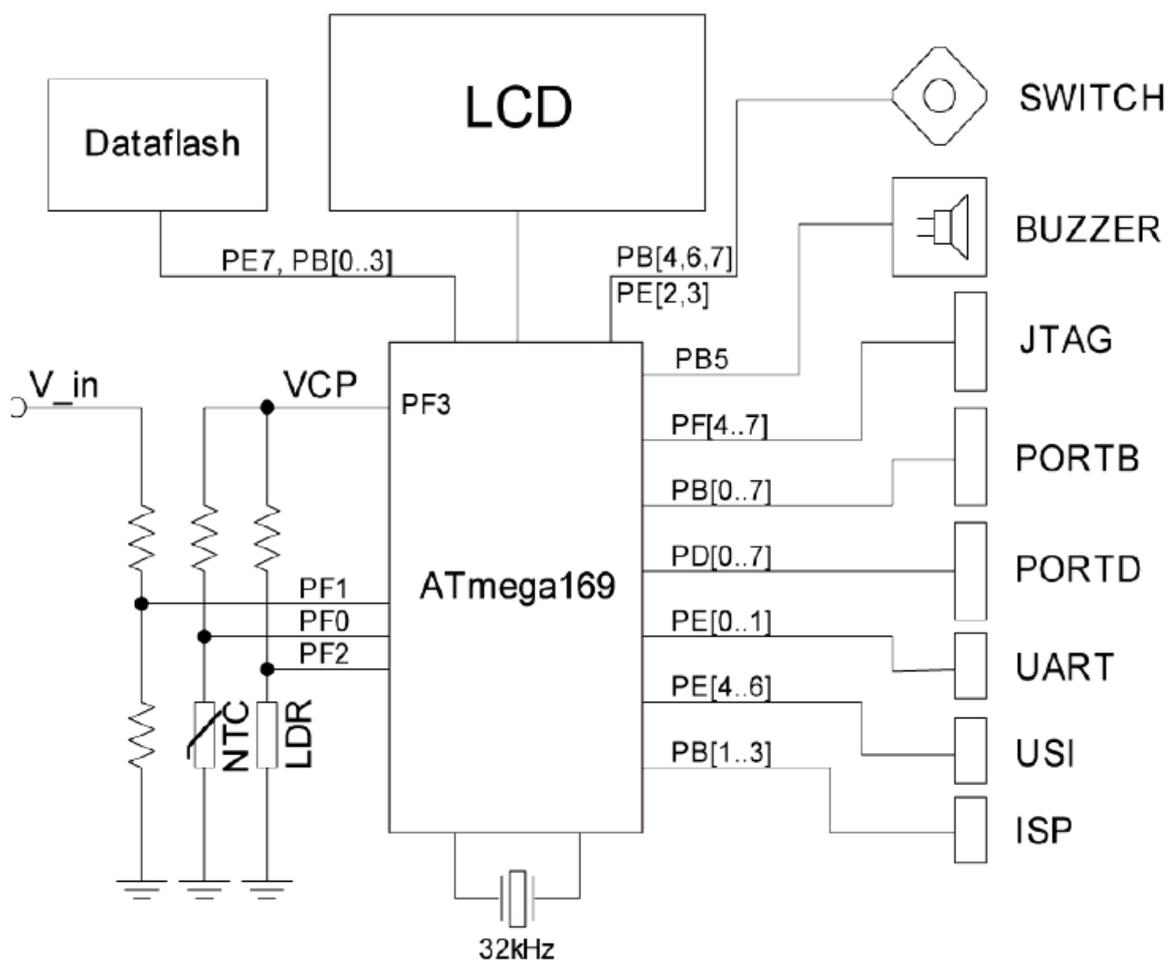


Figura 2.11: Diagrama de las conexiones de los periféricos externos con el microcontrolador ATmega169.

Además el microcontrolador también dispone de dos memorias de datos:

- ❖ Una memoria volátil SRAM de 1 Kbyte de capacidad.
- ❖ Una memoria no volátil EEPROM de 512Bytes de capacidad.

Dado que las instrucciones del AvrButterfly tiene la longitud de 16 o 32 bits, la memoria Flash de programa se organiza en dos bancos de 8 Kbyte (8K + 16). [9]

## **CAPÍTULO 3**

### **3. Configuración e Implementación del “Motor Control Kit”**

#### **3.1 Ensamblando el Motor Control Kit:**

Con ayuda de la Guía del Usuario del Motor Control Kit armamos y configuramos nuestro proyecto. Pero antes tuvimos que soldar espadines a las tarjetas LPCXpresso LPC1769 como se muestra en la figura 3.1 y en la figura 3.2.



Figura 3.1: LPCXpresso 1769 sin espadines



Figura 3.2: Soldando los espadines a la tarjeta

Es con esto que se puede montar la tarjeta en un protoboard, para que con la LPCXpresso LPC1769 tener el control de botoneras como se ve en la Figura 3.3: .

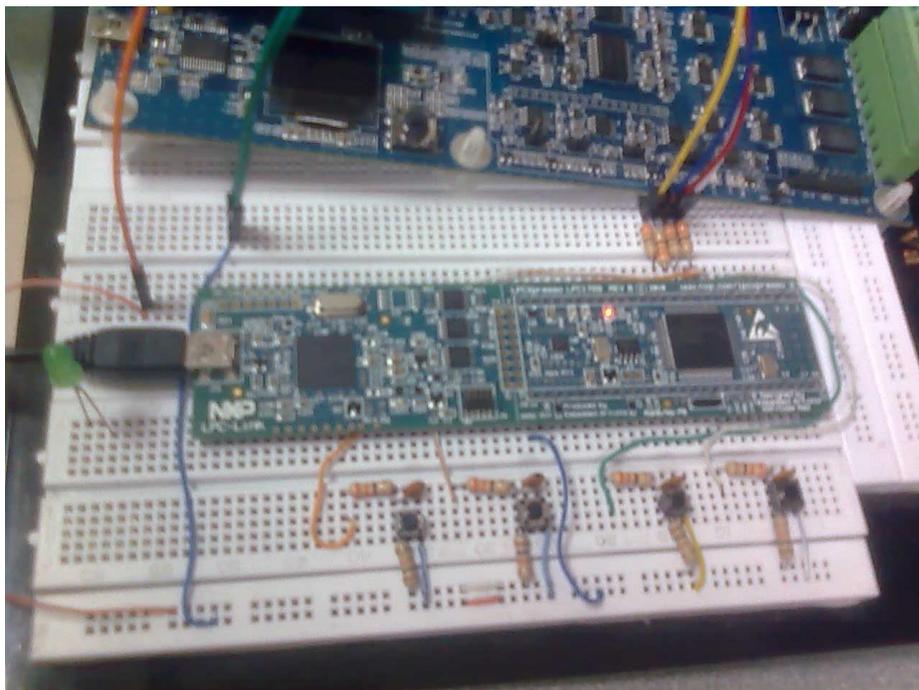


Figura 3.3: Tarjeta LPCXpresso 1769 montada sobre protoboard

Luego, ensamblamos el Motor Control Kit como lo indica el manual (NXP) en su página 22.

### 3.2 Paso 1: Montaje de la tarjeta LPCXpresso LPC1114

Se acopla la tarjeta LPCXpresso LPC1114 sobre la tarjeta de Control del Motor, pero nos aseguramos de que vaya puesta sobre la parte superior como indica la placa, donde van las LPC1114, pues la parte inferior es para las LPC176X.

También, tome en cuenta que los pines deben acoplarse todos dentro del socket, ninguno debe quedar fuera.

### 3.3 Paso 2: Conexión del Motor BLCD

Se deben conectar los cables en los sockets que van sobre la Tarjeta de Control. Entonces conectamos los cables como indica el manual en las figuras Figura 3.4: y Figura 3.5:.

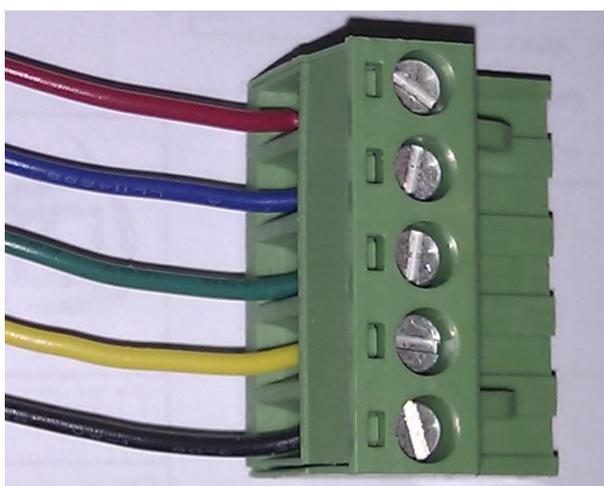


Figura 3.4: Conector de los Sensores Efecto Hall

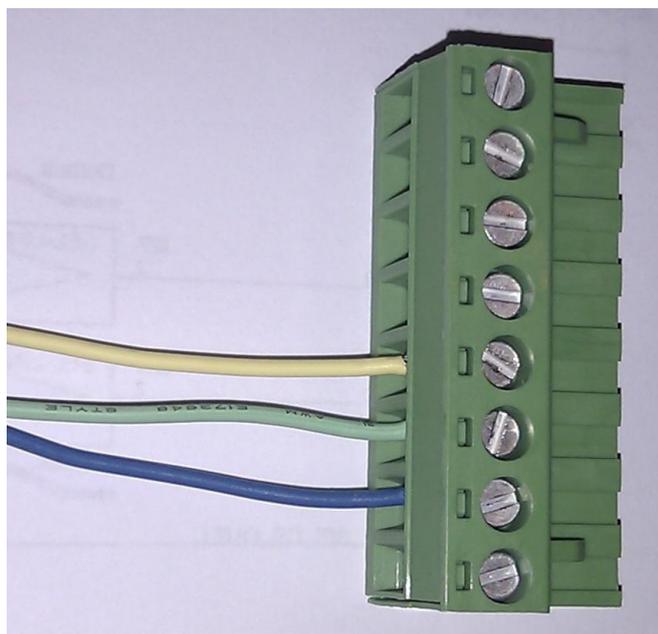


Figura 3.5: Conector de alimentación trifásica del motor BLCD

Finalmente las conexiones quedan como se muestra en la Figura 3.6:

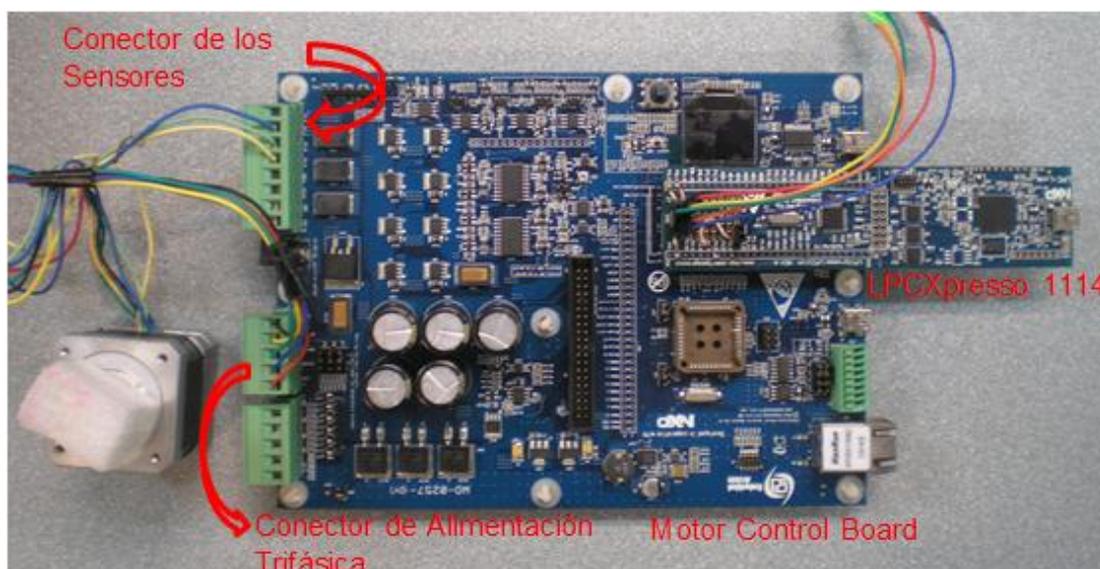


Figura 3.6: Motor BLDC conectado a los sockets

El Kit deberá quedar como se muestra en la figura 3.7.



Figura 3.7: Motor Control Kit

### 3.4 Paso 3: Conectando la Alimentación

La Tarjeta de Control del Motor debe ser siempre alimentada por una fuente externa, puesto que la LPC1114 no puede alimentarla por sí sola.

El kit viene con el cable de poder respectivo para la conexión de la fuente de poder.

Alimentamos las tarjetas y verificamos que los tres leds encienden, indicando que la entrada de poder es correcta.

### 3.5 Programando la LPCXpresso LPC1114:

Con el IDE LPCXpresso instalado, se puede programar la LPC1114 fuera o sobre la Tarjeta de Control del Motor.

Importamos el proyecto “BLDC Sensored” al workspace, hacemos el Debug (Figura 3.8:) y la tarjeta LPC1114 quedará programada

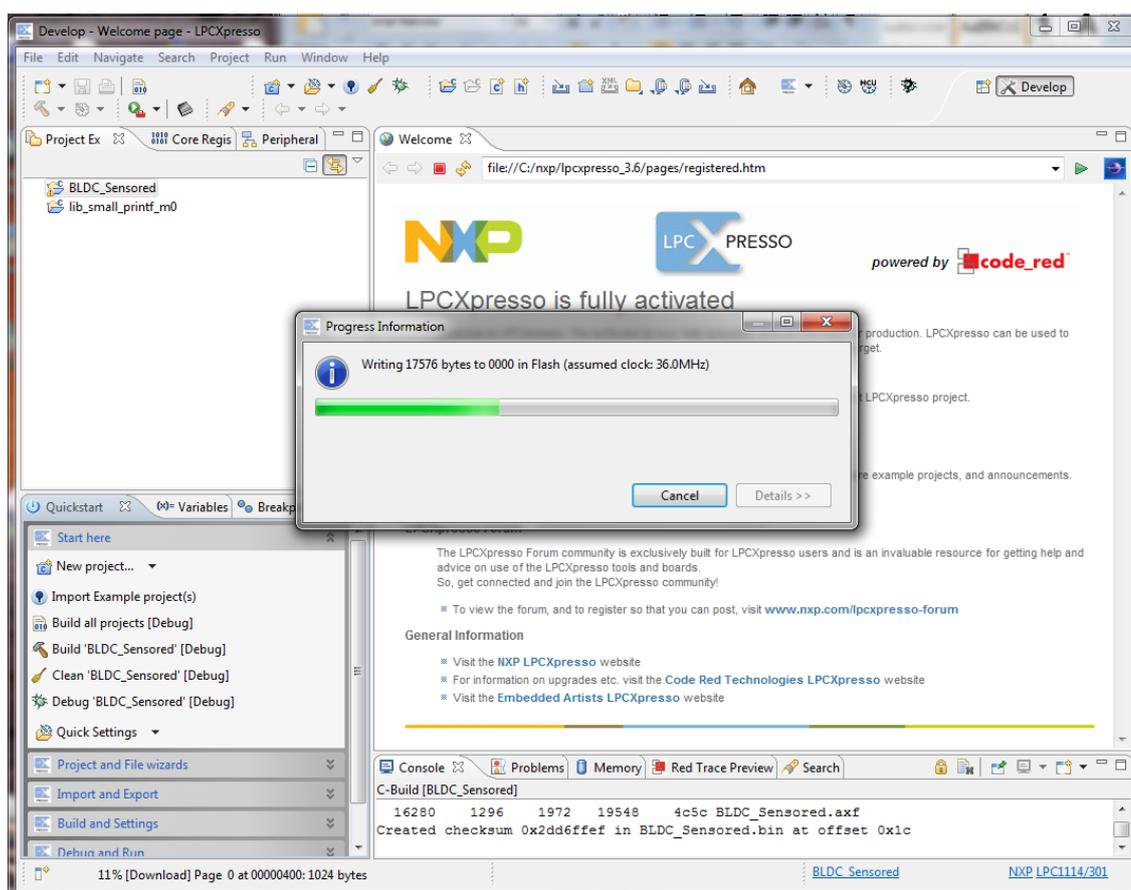


Figura 3.8: Programando la tarjeta LPCXpresso 1114

Así, pulsamos el botón de reset que está a la derecha de la pantalla OLED y ya se puede comprobar el control de velocidad del motor.

### 3.6 Conexión de las Tarjetas

En las siguientes figuras se muestra la conexión entre las tarjetas LPCXpresso 1769 y la tarjeta LPCXpresso 1114, donde se utilizaron pines sin uso en ambas tarjetas. En la tarjeta LPCXpresso 1769 se utilizaron los pines 1-3 del puerto 3. Se realizó la conexión de las tarjetas como se muestran en las Figura 3.9 y Figura 3.10:.

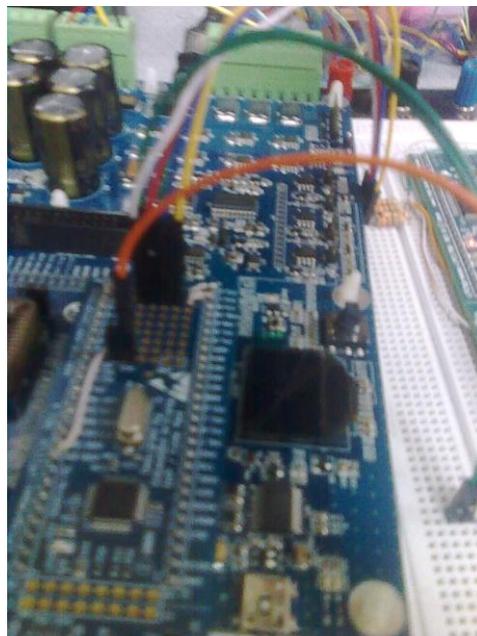


Figura 3.9: Conexión de las tarjetas 1

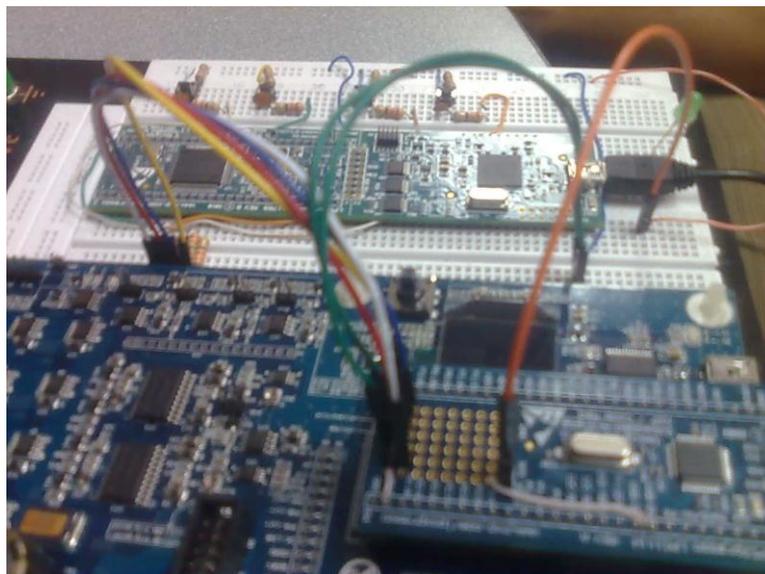


Figura 3.10: Conexión de la tarjetas 3

## **CAPÍTULO 4**

### **4. Pruebas Experimentales del Proyecto**

Con las tarjetas LPCXpresso acopladas como se mostró en capítulos anteriores y con los programas instalados en cada una, se procedió a hacer las pruebas funcionales del proyecto.

Para esto, se muestra a continuación los programas mencionados, los cuales se instalaron en la tarjeta respectiva con el software de programación LPCXpresso v4.1.5\_219.

## 4.1 Parámetros del Motor Programados En la LPC1114

En la tarjeta LPCXpresso 1114 se programó el siguiente código, que como se podrá ver, está escrito en lenguaje C.

Este es el programa que contiene los parámetros del motor:

```
void vBLDC_Init (MOTOR_TypeDef *ptr)
{
    ptr->max_mv = MOTOR_MAX_PERIOD;

    /******
    /* Default PID structure initialization      */
    /******

    ptr->Kp = 5;
    ptr->Ki = 2;
    ptr->Kd = 0;
    ptr->sp = 3000;      // RPM
    ptr->pv = 0;
    ptr->IntError = 0;
    ptr->LastError = 0;
    ptr->mv = 0;
    ptr->HALstate[0] = 0;
    ptr->HALstate[1] = 0;
    ptr->TMRval[0] = 0;
    ptr->TMRval[1] = 0;
    ptr->Deadtime = 0;
    ptr->Direction = CW;
```

```
ptr->CMT_step = 1;
ptr->Startup = 35;
ptr->max_mv = MOTOR_MAX_PERIOD;
ptr->PolePairs = 4;
ptr->Brake = 0;
ptr->Enable = 0;
ptr->RPM = 0;
ptr->RampingUp = TRUE;
init_timer32PWM(TIMER0, MOTOR_PERIOD, (MATCH0 | MATCH2 | MATCH3));
/* Setup Phase B high and low side PWM signals */
init_timer16PWM(TIMER0, MOTOR_PERIOD, (MATCH0 | MATCH1),
                CAP_DISABLE);
/* Setup Phase C high and low side PWM signals */
init_timer32PWM(TIMER1, MOTOR_PERIOD, (MATCH0 | MATCH1));
/* Make sure no match occurs */
vBLDC_Stop ();
/* enable all timers for motor driving */
enable_timer16(TIMER0);
enable_timer32(TIMER0);
enable_timer32(TIMER1);
}
```

Se puede ver los parámetros que han sido programados los cuales deben ser acordes con el motor BLDC que se está controlando.

## 4.2 Programando la LPCXpressoLPC1769:

Para recibir los mandos del joystick de la Butterfly, se programó el siguiente código principal en la tarjeta LPCXpresso 1769.

```

/*----- SEMINARIO DE GRADUACION-----
* Profesor: Ing. Carlos Valdivieso A.
* Creado: 29/03/2012
-----*/
/***** Librerías empleadas *****/
#include "LPC17xx.h"
#include "type.h"
#include "uart.h"
#include<string.h>

/***** Creación de Variables *****/
extern volatile uint32_t UART3Count;
extern volatile uint8_t UART3Buffer[BUFSIZE];
/*****
*
This program has been test on LPCXpresso 1700.
*****/

/***** Programa principal *****/
int main (void)
{
    uint32_t j;

    LPC_GPIO2->FIODIR = 0xFFFFFFFF; /* P2 definidos como salidas */
    LPC_GPIO2->FIOCLR = 0xFFFFFFFF;
    LPC_GPIO2->FIOSET = 0x1F;
    UARTInit(3, 9600); /* Configuración del BaudRate */
    while (1) /* Lazoinfinito */
    {
        if ( UART3Count != 0 )

        {
            LPC_UART3->IER = IER_THRE | IER_RLS; /* Disable RBR */

```

```

if(*UART3Buffer==0x43)
{ // ASCII C
    LPC_GPIO2 -> FIOSET = 0x1F;
    LPC_GPIO2 -> FIOCLR = 1 << 4;
    for(i=0;i<2500000;i++);
    LPC_GPIO2 -> FIOSET = 0x1F;
}
if(*UART3Buffer==0x55)
{ // ASCII U

    LPC_GPIO2 -> FIOSET = 0x1F;
    LPC_GPIO2 -> FIOCLR = 1 << 0;
    for(i=0;i<2500000;i++);
    LPC_GPIO2 -> FIOSET = 0x1F;
}
if(*UART3Buffer==0x44)
{ // ASCII D
    LPC_GPIO2 -> FIOSET = 0x1F;
    LPC_GPIO2 -> FIOCLR = 1 << 1;
    for(i=0;i<2500000;i++);
    LPC_GPIO2 -> FIOSET = 0x1F;
}
if(*UART3Buffer==0x52)
{ // ASCII R

    LPC_GPIO2 -> FIOSET = 0x1F;
    LPC_GPIO2 -> FIOCLR = 1 << 2;
    for(i=0;i<2500000;i++);
    LPC_GPIO2 -> FIOSET = 0x1F;
}
if(*UART3Buffer==0x4C)
{ // ASCII L
    LPC_GPIO2 -> FIOSET = 0x1F;
    LPC_GPIO2 -> FIOCLR = 1 << 3;
    for(i=0;i<2500000;i++);
    LPC_GPIO2 -> FIOSET = 0x1F;
}

    UART3Count = 0;
    /* Re-enable RBR */
    LPC_UART3->IER = IER_THRE | IER_RLS | IER_RBR;
}
}
}
/***** Fin del código *****/

```

### 4.3 Programando la AVR ButterFLy:

Mediante el IDE correspondiente AVR STUDIO 4, se programó la tarjeta Butterfly con el siguiente código:

```
//----- SEMINARIO DE GRADUACION-----
/* Profesor: Ing. Carlos Valdivieso A.
/* Creado: 28/03/2012
//-----
/****** Librerías Empleadas *****/
#include<avr/io.h>
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include <avr/delay.h>
#include <inttypes.h>
#include "mydefs.h"
#include "LCD_functions.h"
#include "LCD_driver.h"
#include "button.h"
#include "usart.h"

/****** Programa principal *****/
int main(void)
{
uint8_t input;
char *cadena;
int i;
ACSR = (1<<ACD); // Deshabilita comparador analógico
DIDRO = (7<<ADCOD); // Deshabilita entrada digital en PF0-2
PORTB = (15<<PB0); // Activa Pull-ups
PORTE = (15<<PE4); // Activa Pull-ups
Button_Init(); // Inicializa interrupción por cambio de pin en el joystick
LCD_Init(); // Inicializa el LCD
CLKPR = (1<<CLKPCE); //Activa el reloj del Preescalador

/****** Configuración del Preescalador *****/
//set prescaler = 8, Inter RC 8Mhz / 8 = 1Mhz
CLKPR = (0<<CLKPS1) | (1<<CLKPS0);
USART_Init(25.04); // Configuración de los BaudRates
//sei();
```

```

//***** Lazo lectura instrucciones joystick*****
while (1)
{
    if (statetext){
LCD_puts_f(statetext, 1);
LCD_Colon(0);
statetext = NULL;
    }

    input = getkey();    // Lee los botones del joystick

    switch (input) {
        case KEY_ENTER:
statetext = PSTR("UNSIGNED");
Usart_Tx('C');
        break;
        case KEY_NEXT:
statetext = PSTR("REVERSE");
Usart_Tx('R');
        break;
        case KEY_PREV:
statetext = PSTR("ON-OFF");
Usart_Tx('L');
        break;
        case KEY_PLUS:
statetext = PSTR("RISE_SPEED");
Usart_Tx('U');
        break;
        case KEY_MINUS:
statetext = PSTR("REDUCE_SPEED");
Usart_Tx('D');
        break;
        default:
break;
    }
}
return 0;
}
//***** Fin del Código*****

```

Importamos el proyecto “BLDC Sensored” al workspace,

#### 4.4 Pruebas del Control de Velocidad del Motor

Con los programas instalados en cada tarjeta se procede a hacer las pruebas respectivas, que en este caso, son las de velocidad del motor, verificando que el motor funcione correctamente y que responda a los mandos programados especificados anteriormente.

Y efectivamente, es así como con el joystick logramos controlar la velocidad del motor BLDC.

En la Figura 4.1: se puede ver la tarjeta LPC1769 con las señales de prueba:

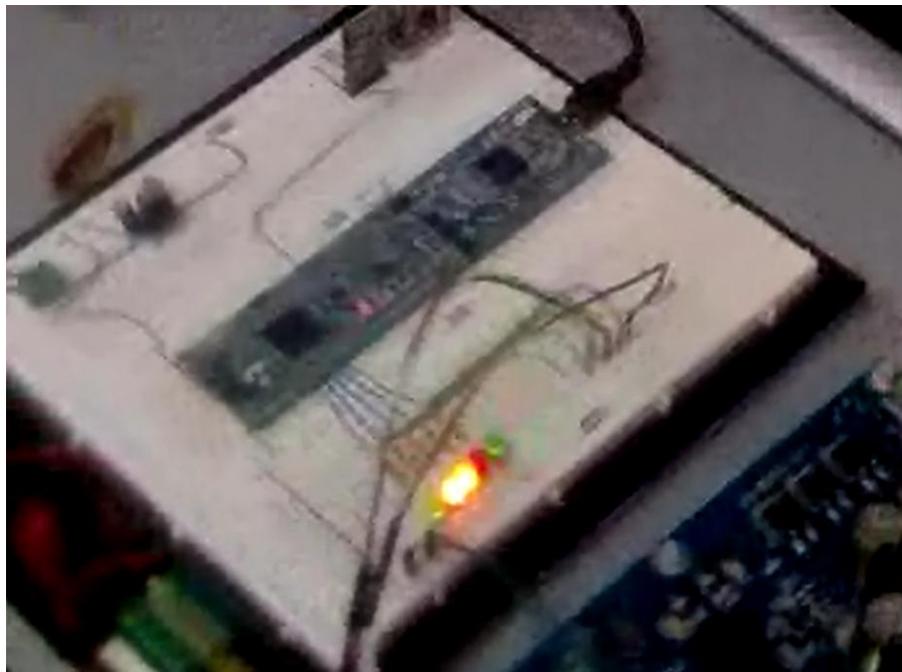


Figura 4.1: Señales del Joystick en LPC1769

Y esas señales de prueba son recibidas por la LPCXpresso 1114 como se puede ver en la Figura 4.2: y conectada sobre la Motor Control Board envía las señales al motor y recibe las señales de los sensores de efecto Hall del mismo.

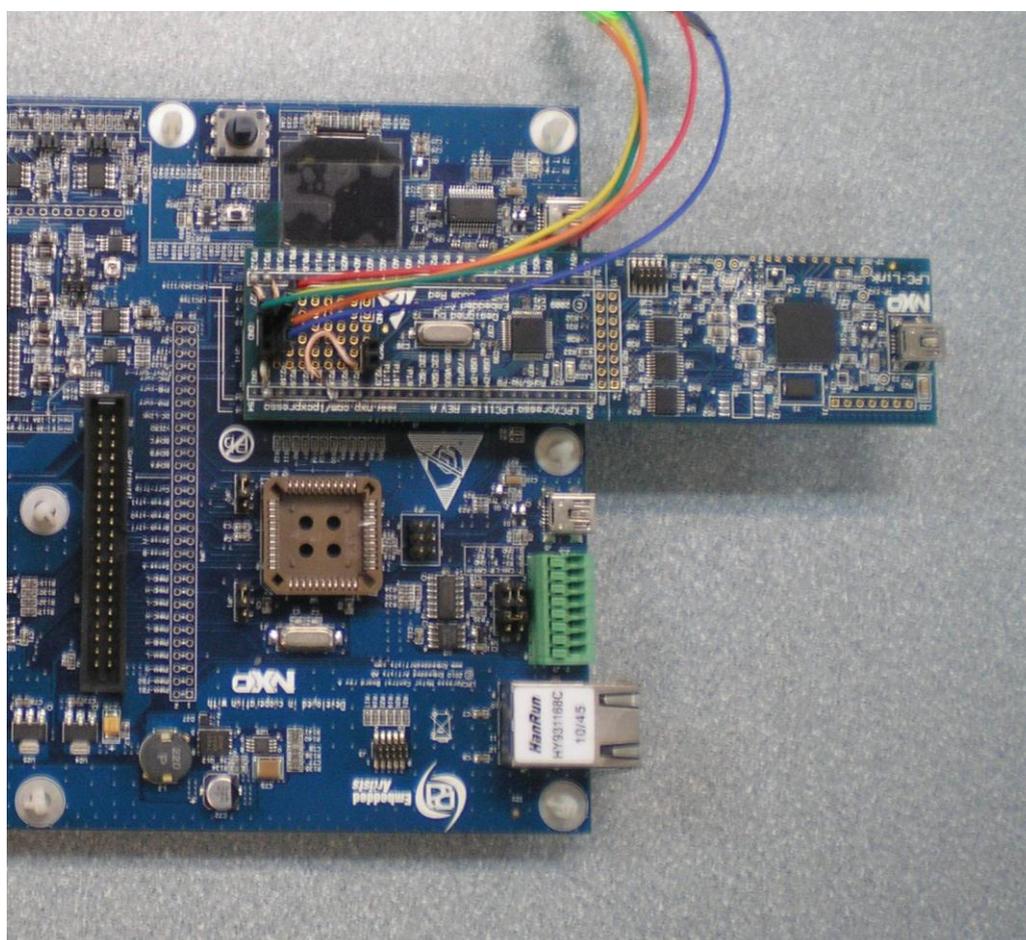


Figura 4.2: Parámetros y control del Motor en la tarjeta LPCXpresso 1114

En la figura 4.3 se puede ver el Motor Control Board o Motherboard, que como se especificó en capítulos anteriores, es la interface entre el circuito de control y fuerza, del motor y se puede apreciar su correcto funcionamiento (Los Led's nos indican que la tarjeta se encuentra en estado operativo).

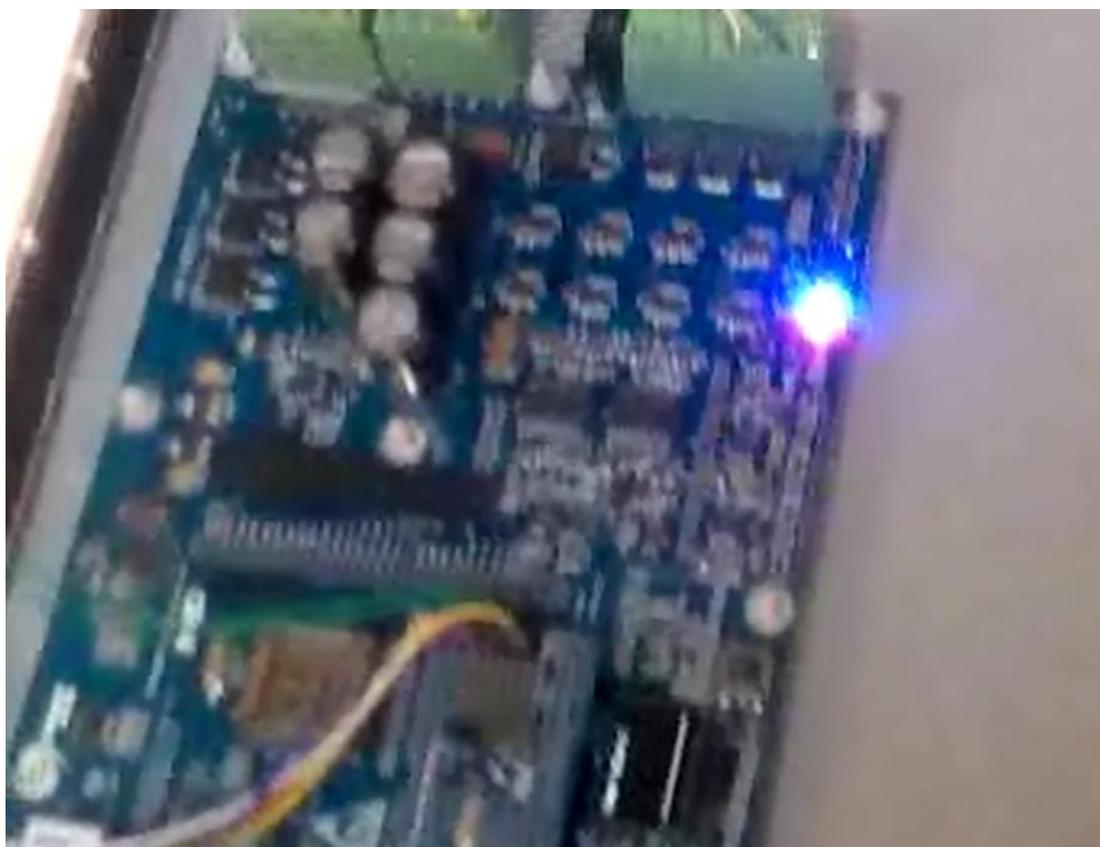


Figura 4.3: La tarjeta motor control kit en funcionamiento

Finalmente se puede observar en la figura 4.4 como el motor responde a las señales enviadas desde la tarjeta AvrButterfly como se esperaba desde un principio el motor responde a los mandos de la tarjeta butterfly.



Figura 4.4: Motor BLDC controlado por Joystick de la tarjeta AvrButterfly.

## **CONCLUSIONES**

1. Mediante la teoría correspondiente a Microcontroladores y lo documentado en esta Tesina, complementamos lo anteriormente dicho implementando un prototipo para control de un motor BLDC, haciendo uso de varias herramientas como el software LPCXpresso v4.1.5\_219 y sus respectivas tarjetas programables como lo son la LPCXpressoLPC1769 y la LPCXpressoLPC1114.
2. Se programó el mando del motor (encendido, apagado y cambio de velocidades) en la tarjeta LPCXpressoLPC1769 mediante el software ya mencionado el cual es el LPCXpresso v4.1.5\_219. Así mismo, el control del motor, todas sus características y los algoritmos de funcionamiento con sus respectivas librerías fueron programadas en la tarjeta LPCXpressoLPC1769.

3. En este proyecto se logró optimizar el proceso de control de velocidad de un motor BLDC y también mejorar el rendimiento y consumo de energía, ya que de por sí las características y prestaciones de este tipo de motores hacen posible que se pueda hacer el control correcto del motor con sus parámetros, y este control no fuera posible sin la avanzada tecnología y velocidad de los dispositivos electrónicos de NXP como lo son las LPCXpresso.
4. Existen diferentes formas de obtener la lectura de los sensores del motor BLCD para generar la secuencia correcta para el giro del rotor de imán permanente, y a su vez existen diferentes métodos de control del motor BLCD los cuales fueron programados en la LPCXpressoLPC1114 mediante el LPCXpresso v4.1.5\_219 como se mencionó antes.
5. Se identificaron las entradas y salidas del joystick y del display en la tarjeta LPCXpresso Motor Control para implementar su funcionamiento con otros periféricos externos como la tarjeta Butterfly o con botoneras independientes las cuales serían leídas por la LPCXpressoLPC1769 y transmitidas por medio de estas a la LPCXpressoLPC1114.
6. Para controlar la velocidad del motor BLDC usamos una señal PWM con un duty cycle que no excede el 90% y para ello se utilizaron los TIMER0 y TIMER1 en modo de 32 bits.

7. El control de velocidad para motores BLCD con sensores efecto Hall utilizados en este proyecto muestra la fácil implementación y aplicación para el uso didáctico, y experimental donde se puede comprobar el funcionamiento de sus etapas, como la etapa de control y potencia del motor.

## **RECOMENDACIONES**

1. Tener cuidado al manipular la tarjetas LPCXpresso PC1769, LPCXpresso LPC1114, y la tarjeta LPCXpresso Motor Control ya que sus circuitos integrados son muy sensibles a cargas electrostáticas. Para esto se debe sostener las tarjetas con las yemas de los dedos por los bordes y no tocar los terminales ni los circuitos integrados.
2. Se recomienda leer meticulosamente los datasheets del Kit AVR Butterfly y también los del User Manual del Motor Control Kit, para conocer la correcta configuración de los componentes de hardware y de sus pines ya que cualquier conexión errónea puede generar daños irreversibles en los componentes.

3. Leer el User Manual de la tarjeta LPCXpresso LPC1769 para conocer las funciones y los diferentes usos que se le puede dar a la tarjeta, y las características del Microcontrolador ARM Cortex-M3 LPC1769. De esta forma se pueden aprovechar todos los recursos y características de estas tecnologías.
4. Ver los diferentes programas en la página de NXP sobre la tarjeta LPCXpresso LPC1769 para poder comprender más la programación de la tarjeta y así poder realizar, a futuro, otros proyectos.
5. Como se trata de un prototipo, el proyecto no está listo para ser introducido al mercado aún. Una de las desventajas que pesan sobre este hecho es que el espacio que ocupa es demasiado grande con respecto a otros sistemas, de modo que
6. Por lo anteriormente dicho, se puede ver que por el diseño de las tarjetas, este proyecto no se puede aplicar directamente a un sistema completo puesto que su configuración no es universal y está diseñado solamente para el propósito indicado en el manual. Se podría también implementar el sistema para otro motor con otras características, pero estas deben ser programadas en el código de la LPCXpressoPLC1114.

## **BIBLIOGRAFÍA**

[1] Allied Motion Technologies Inc., Brushless Motors,

<http://www.alliedmotion.com/Products/Product.aspx?p=2>, abril 2012

[2] NXP Semiconductors, LPC1769/68/67/66/65/64/63,

[http://www.nxp.com/documents/data\\_sheet/LPC1769\\_68\\_67\\_66\\_65\\_64\\_63.p](http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf)

df , Abril de 2012

[3] Digi-Key Corporation, LPC1700 Microcontroller,

<http://dkc1.digikey.com/mx/es/tod/NXP/LPC1700/LPC1700.html>, Marzo de

2012

[4] Ingenia-cat - Motion Control Department, Técnicas de control para motores Brushless, <http://www.ingeniamc.com/Es/-Control-techniques-for-brushless-motors.pdf>, Marzo de 2012

[5] Gabriel Gross Igor, Control de un motor brushless con aumento de la resolución del resolver mediante "oversampling", <http://upcommons.upc.edu/pfc/bitstream/2099.1/2626/1/34499-1.pdf>, Marzo de 2012

[6] Centre for Electrical Machines and Power Electronics, Brushless DC Motors, [http://services.eng.uts.edu.au/cempe/subjects\\_JGZ/ems/ems\\_ch12\\_nt.pdf](http://services.eng.uts.edu.au/cempe/subjects_JGZ/ems/ems_ch12_nt.pdf), Marzo de 2012

[7] Microchip, Brushless DC Motor Control Made Easy, <http://ww1.microchip.com/downloads/en/AppNotes/00857a.pdf>, Marzo de 2012

[8] Motor Brushless, Tutorial del Motor DC, [www.motorbrushless.es/documentos/tutorial-motor-dc.pdf](http://www.motorbrushless.es/documentos/tutorial-motor-dc.pdf), Marzo de 2012

[9] Gabriel Gross Igor, Control de un motor brushless con aumento de la resolución del resolver mediante "oversampling", <http://upcommons.upc.edu/pfc/bitstream/2099.1/2626/1/34499-1.pdf>, Marzo de 2012

[10] Atmel Corporation, Sensorless control of 3-phase brushless DC motors, <http://www.atmel.com/Images/doc8012.pdf>, Marzo de 2012

[11] Cenidet, Control de un motor Brushless para aplicación a vehículos eléctricos, [http://www.cenidet.edu.mx/subaca/web-elec/tesis\\_mc/243MC\\_dlc.pdf](http://www.cenidet.edu.mx/subaca/web-elec/tesis_mc/243MC_dlc.pdf), Marzo de 2012

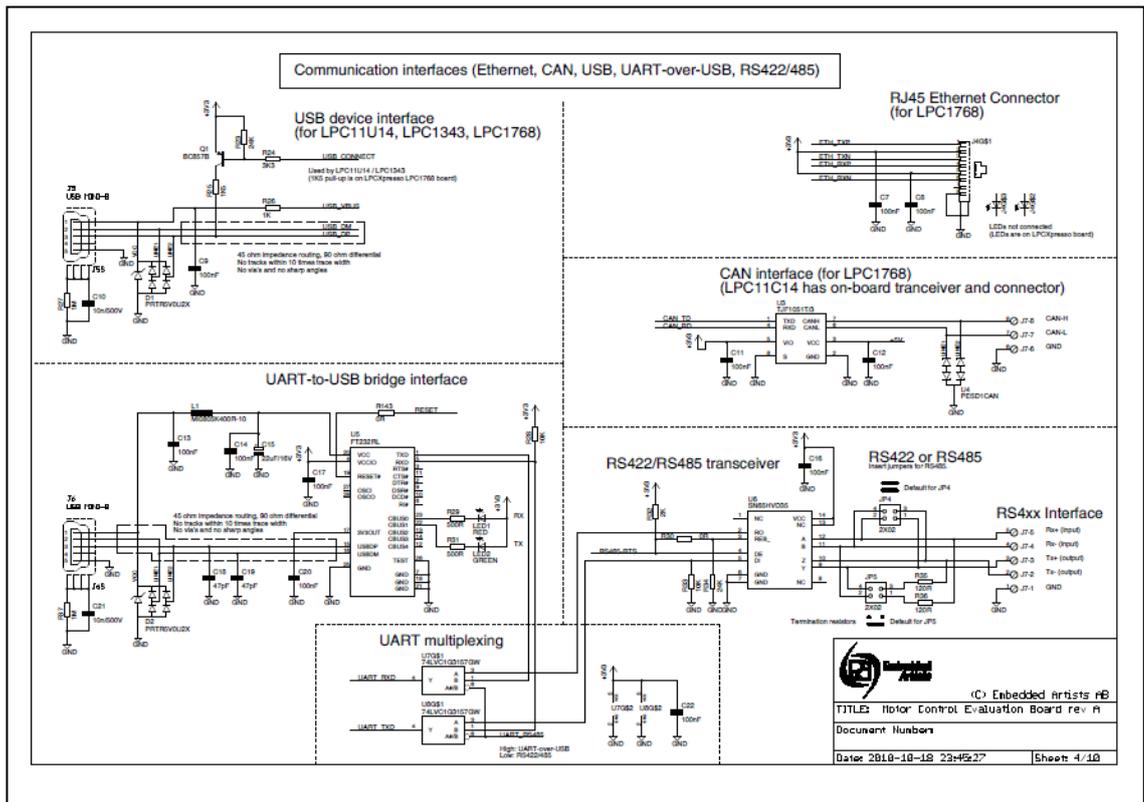
[12] Sensors, Position and Speed Control of Brushless DC Motors Using Sensorless Techniques and Application Trends, <http://www.google.com.ec/url?sa=t&rct=j&q=motores%20brushless%20pdf&source=web&cd=41&ved=0CDAQFjAAOCg&url=http%3A%2F%2Fwww.mdpi.com%2F1424-8220%2F10%2F7%2F6901%2Fpdf&ei=JoeZT-3IJouo8ATW6umBBg&usg=AFQjCNFROTbu6OwFz9t6AuI9U8wIBIWIXQ&cad=rja>, Abril de 2012

[13] Microsemi, Speed Control of Brushless DC Motors-Block Commutation with Hall Sensors, [http://www.actel.com/documents/Dual\\_Kit\\_Block\\_Commutation\\_With\\_Hall\\_Sensors\\_UG.pdf](http://www.actel.com/documents/Dual_Kit_Block_Commutation_With_Hall_Sensors_UG.pdf), Abril de 2012.

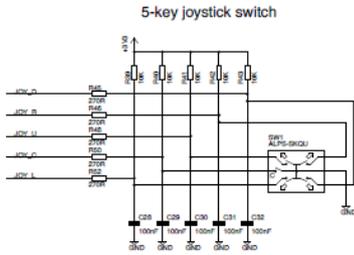
[14] Trinamic Motion Control, QMot QBL4208 family Manual, [http://www.trinamic.com/tmc/media/Downloads/QMot\\_motors/QBL4208/QBL4208\\_manual.pdf](http://www.trinamic.com/tmc/media/Downloads/QMot_motors/QBL4208/QBL4208_manual.pdf), Abril de 2012

# ANEXOS

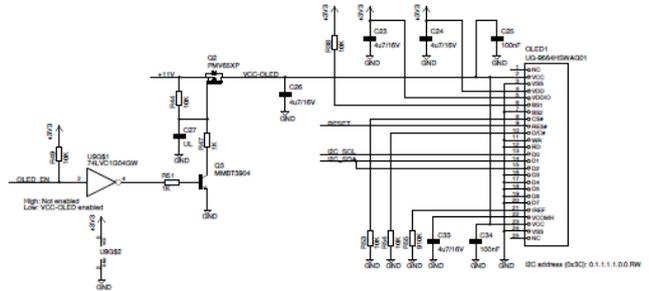
## ESQUEMÁTICOS DE LA MOTOR CONTROL BOARD



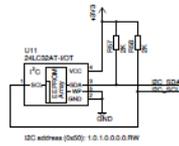
User interface with OLED and 5-key joystick



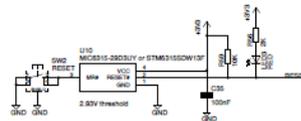
96x64 White OLED with I2C interface



32Kbit I2C-E2PROM



Reset generation

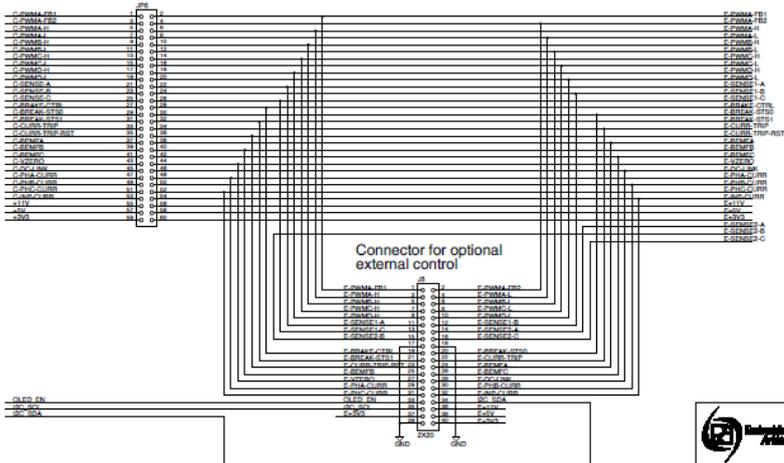


(C) Embedded Artists AB  
 TITLE: Motor Control Evaluation Board rev. A  
 Document Number  
 Date: 2010-10-18 23:49:27 Sheet 5/10

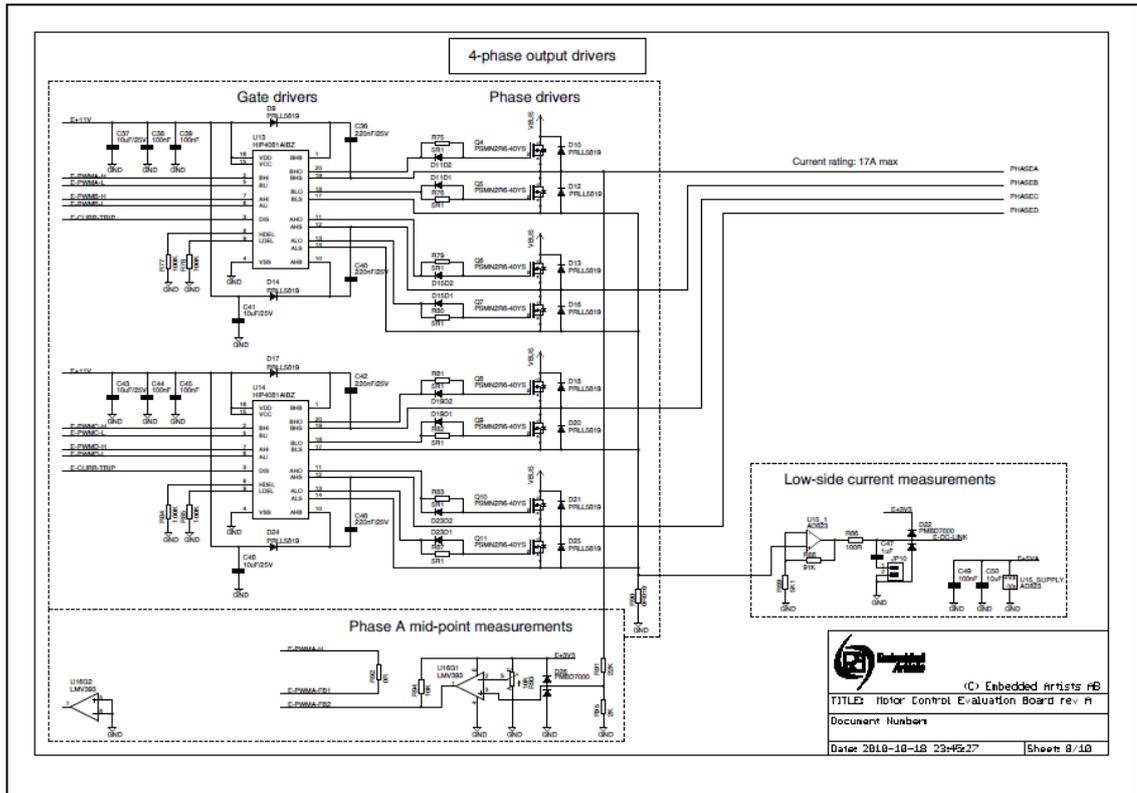
Interface between controller and motor electronics

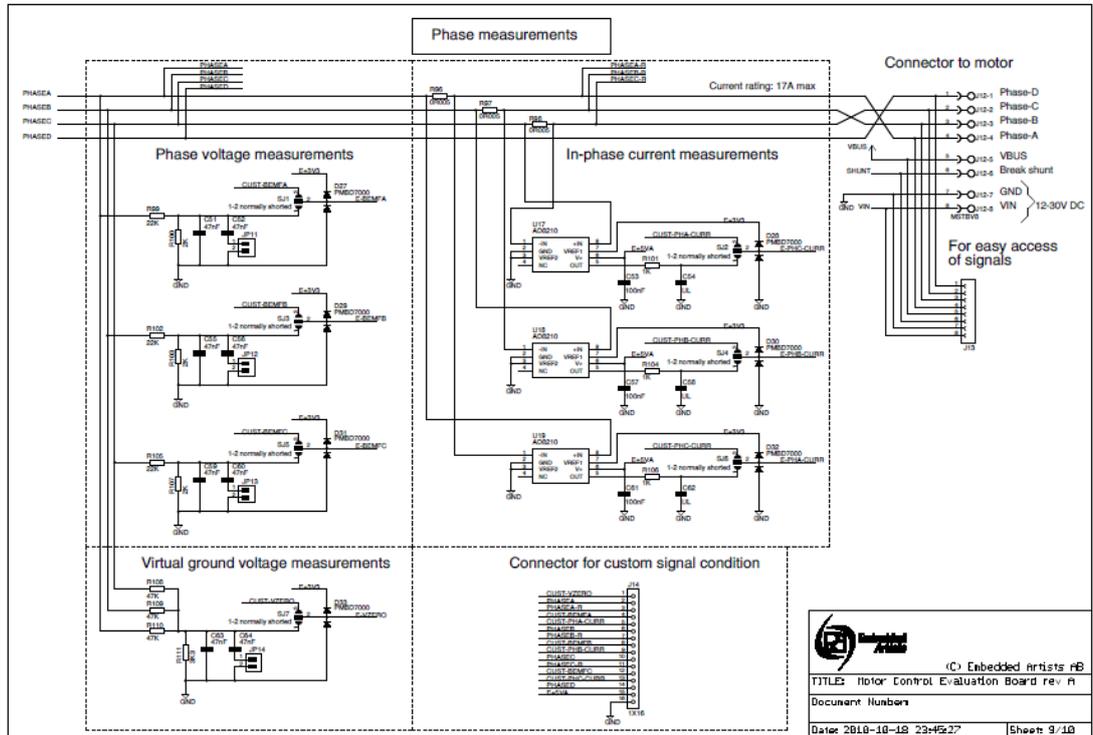
"C-" prefix on control side      "E-" prefix on motor electronic side

All signals normally connected



 © Embedded Artists AB  
 TITLE: Motor Control Evaluation Board rev. A  
 Document Number:  
 Date: 2010-10-18 23:45:27 Sheet 6/10






  
 (C) Embedded Artists AB
   
 TITLE: Motor Control Evaluation Board rev. A
   
 Document Number:
   
 Date: 2010-10-18 23:45:27 | Sheet 9/10

# MOTOR BLDC

25mm  
4mm  
42.3  
3.5mm  
3.5mm  
4mm  
26 AWG UL328 motor cables  
22 AWG UL1007 hall sensor cables

Rotation: CW SEE FROM SHAFT END

MANUFACTURER MODEL	42BLF01
RESISTANCE	2.2 OHMS
NOMINAL VOLTAGE	24 VDC
NO LOAD SPEED	6000 RPM
NO LOAD CURRENT	0.5 Amps
RATED TORQUE	0.063 N-m
RATED SPEED	4000 ±300 RPM
BACK EMF CONSTANT	4.4 V/KRPM
TORQUE CONSTANT	0.042 N-m/A

+	+	+	
+5V	RED	红	26 AWG UL1007
HALL A	YELLOW	黄	
HALL B	GREEN	绿	
HALL C	BLUE	蓝	22 AWG UL3285
GND	BLACK	黑	
PHASE A	YELLOW	黄	
PHASE B	GREEN	绿	
PHASE C	BLUE	蓝	

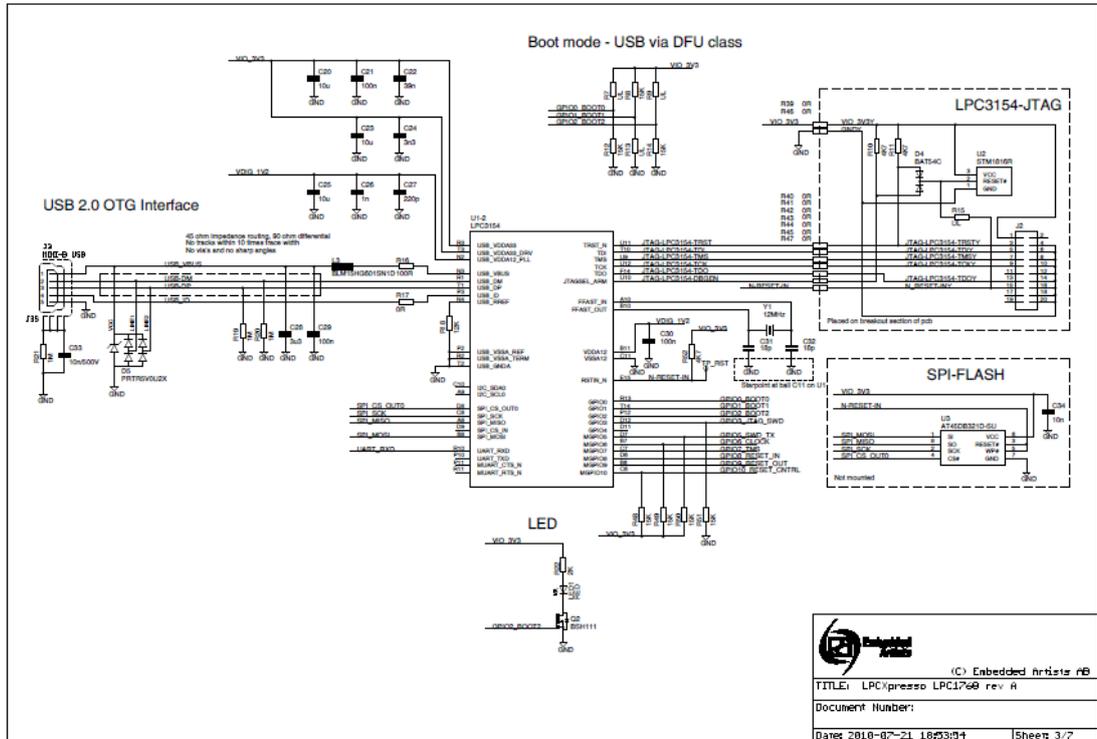
Connection diagram:

Hall sensor cable  
Note: hall sensor power supply voltage +5 to +20V DC

Wiring cable

						42BLF01	
						外形图	
						惠州台泰电机有限公司	

# ESQUEMÁTICOS DE LA LPCXPRESSOLPC1769



### SWD/JTAG Interface

