



# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“CONTROL DE MOTORES SIN ESCOBILLAS (BLDC) Y SIN  
SENSORES USANDO EL MICROCONTROLADOR ARM CORTEX3  
CON 32 BITS DE LPCXPRESSO”

TESINA SE SEMINARIO

Previa a la obtención del título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

INGENIERO EN ELECTRICIDAD

ESPECIALIZACIÓN EN ELECTRÓNICA Y AUTOMATIZACIÓN  
INDUSTRIAL

PRESENTADO POR:

Oscar Roberto Saigua Labre

Emerson Javier Chang Suárez

GUAYAQUIL- ECUADOR

AÑO 2012

## **AGRADECIMIENTO**

A Dios primeramente por guiar nuestro camino y darnos siempre la inteligencia y sabiduría para saber tomar nuestras decisiones.

A nuestros padres por saber llevarnos siempre por el camino del bien y brindarnos todo su apoyo en nuestras vidas y en toda nuestra etapa estudiantil.

Al Ing. Carlos Valdivieso por saber guiarnos en el desarrollo de esta tesis y su apoyo incondicional en la ejecución de nuestro proyecto.

## DEDICATORIA

A nuestros padres por ser ejemplo de nuestras vidas, por habernos brindado la mejor educación posible y ser apoyo incondicional en toda nuestra etapa de aprendizaje, inculcándonos valores que nos ayudarán a ser mejores profesionales y personas para servir a nuestra sociedad.

**TRIBUNAL DE SUSTENTACIÓN**

A handwritten signature in blue ink, appearing to read 'Carlos Valdivieso', written over a horizontal line.

Ing. Carlos Valdivieso

**PROFESOR DEL SEMINARIO DE GRADUACIÓN**

A handwritten signature in blue ink, appearing to read 'Hugo Villavicencio V.', written over a horizontal line.

Ing. Hugo Villavicencio V.

**PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA**

## DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesina, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”.

(Reglamento de Graduación de la ESPOL)



\_\_\_\_\_  
**Oscar Roberto Saigua Labre**



\_\_\_\_\_  
**Emerson Javier Chang Suarez**

## ÍNDICE GENERAL

Agradecimiento .....	I
Índice de Figuras .....	X
Índice de Tablas.....	XII
Introducción .....	XIV
Capítulo 1 .....	1
Descripción general del proyecto.....	1
1.1 Antecedentes.....	2
1.2 Motivos para la implementación del control del motor bldc.....	3
1.3 Identificación del problema .....	4
1.4 Objetivos.....	6
1.5 Limitaciones.....	7
Capítulo 2 .....	9
Sustentacion Teórica .....	9
2.1 Resumen .....	9
2.2 Herramienta de software .....	10
2.2.1 Lpcxpresso 2.0_2.64 .....	10

2.3 Herramientas de Hardware.....	12
2.3.1 Introducción a la Tarjeta lpc1769 .....	12
2.3.2 Características de la Tarjeta lpc1769 .....	12
2.3.3 Introducción a la Tarjeta lpc1114 .....	13
2.3.4 Características Principales de Tarjeta lpc1114 .....	14
2.3.5 Motor Control.....	15
2.3.6 Motores Brushless dc (blcd) .....	16
2.3.6.1 Motor Brushless de Conmutación Trapezoidal.....	18
2.3.6.2 Motor Brushless de Conmutación Sinusoidal.....	21
2.3.6.3 Fórmulas Matemáticas para los Motores Brushless.....	22
2.3.7 Los Sensores Efecto Hall .....	25
2.3.8 Motores blcd con Sensores de Efecto Hall.....	26
2.3.9 Motores blcd sin Sensores de Efecto Hall. ....	27
2.3.10 Ventajas de usar Motores blcd sin Sensores .....	28
2.3.11 Principales Diferencias entre Motores sin Escobillas y Motores con Escobillas .....	29
2.3.12 Ventajas del Uso de los Motores Brushless .....	30
2.3.13 Desventajas del Uso de los Motores Brushless .....	31
Capítulo 3 .....	32

Desarrollo del Proyecto.....	32
3.1 Encendido Rotatorio de Leds.....	33
3.1.1 Descripción General.....	33
3.1.2 Diagrama de Bloques.....	33
3.1.3 Diagrama de Flujo .....	34
3.1.4 Descripción del Algoritmo.....	34
3.1.5 Programa Principal.....	35
3.2 Control de Motor bldc Mediante la Tarjeta lpc1114 .....	36
3.2.1 Descripción General.....	36
3.2.2 Diagrama de Bloques.....	37
3.2.3 Diagrama de Flujo Programa Principal main.c.....	38
3.2.4 Descripción del Algoritmo.....	39
3.2.5 Programa Principal.....	40
3.2.6 Diagrama de flujo Programa Controlador de Movimiento del Motor bldc handlers.c .....	48
3.2.7 Descripción del Algoritmo.....	49
3.2.8 Programa Controlador de Movimiento del Motor bldc handlers.c...	50
3.3 Control de Motor bldc Mediante las Tarjetas lpc1769 y lpc1114.....	53
3.3.1 Descripción General.....	53

3.3.2	Diagrama de Bloques.....	54
3.3.3	Diagrama de Flujo del Programa Controlador de Botoneras en la lpc1769 interfaz.c .....	55
3.3.4	Descripción del Algoritmo.....	56
3.3.5	Programa Controlador de Botoneras en la lpc1769 interfaz.c.....	57
capítulo 4 .....		58
Pruebas y Simulaciones .....		58
4.1	Resumen .....	58
4.2	Protoboard.....	59
4.3	Motor Control Kit.....	60
4.4	Encendido Rotatorio de Leds.....	61
4.4.1	Descripción General.....	61
4.4.2	Listado de Materiales .....	62
4.4.3	Diagrama de Conexiones.....	63
4.5	Control de Motor bldc Mediante la Tarjeta lpc1114 .....	64
4.5.1	Descripción General .....	64
4.5.2	Listado de Materiales .....	64
4.5.3	Diagrama de Conexiones .....	66
4.6	Control de Motor bldc Mediante las Tarjetas lpc1769 y lpc1114.....	67

4.6.1	Descripción General .....	67
4.6.2	Listado de Materiales .....	67
4.6.3	Diagrama de Conexiones .....	69

## ÍNDICE DE FIGURAS

Figura 2.1 Vista del programa LPCXpresso 2.0_2.64 [4].....	11
Figura 2.2 Tarjeta LPC1769 [5].....	12
Figura 2.3 Características de la tarjeta LPC1769 [4] .....	13
Figura 2.4 Tarjeta LPC1114 [5].....	14
Figura 2.5 Vista de los principales componentes del Motor Control Board [6]..	15
Figura 2.6 Motor BLDC [7] .....	17
Figura 2.7 Bobinado de un motor BLDC conexión en estrella BLDC [8].....	19
Figura 2.8 Diagrama de bloques de controlador.....	19
Figura 2.9 Diagrama de bloques del controlador con conmutación sinusoidal [8].....	21
Figura 2.10 Bobinado del Rotor de un Motor BLDC [9].....	22
Figura 2.11 Circuito de Alimentación del Motor BLDC [10].....	24
Figura 2.12 Esquema de un motor BLDC con sensor [10].....	24
Figura 2.13 Sensores de Efecto Hall [11] .....	25
Figura 2.14 Motor BLDC con sensores de efecto Hall [13].....	26
Figura 2.15 Motor BLDC sin sensores [15] .....	28
Figura 3.1 Diagrama de Bloques Encendido Rotatorio de Leds .....	33
Figura 3.2 Diagrama de Flujo Encendido Rotatorio de Leds .....	34
Figura 3.3 Diagrama de Bloques Control de Motor BLDC mediante la tarjeta LPC1114.....	37

Figura 3.4 Diagrama de Flujo Control de Motor BLDC mediante la tarjeta LPC1114.....	38
Figura 3.5 Diagrama de Flujo programa controlador de movimiento del motor BLDC handlers.....	48
Figura 3.6 Diagrama de bloques control de motor BLDC mediante las tarjetas LPC1114 y LPC1769. ....	54
Figura 3.7 Diagrama de flujo del programa controlador de botoneras en la LPC1769 INTERFAZ.C.....	55
Figura 4.1 Protoboard. [18].....	59
Figura 4.2 Patrón de disposición de las láminas de material conductor en un Protoboard [19].....	60
Figura 4.3 Lpcxpresso motor control kit [6].....	61
Figura 4.4 Desplazamiento rotatorio de Leds .....	62
Figura 4.5 Conexión del control de motor BLDC mediante la tarjeta LPC1114 .	63
Figura 4.6 Control de motor BLDC mediante la tarjeta LPC1114 .....	65
Figura 4.7 Conexión del control de motor BLDC mediante la tarjeta LPC1114 .	66
Figura 4.8 Control de motor BLDC usando tarjetas LPC1114 y LPC1769.....	68
Figura 4.9 Conexión del control de motor BLDC usando tarjetas LPC1114 y LPC1769.....	69

## ÍNDICE DE TABLAS

Tabla 1.1 Motor BLDC con sensores de efecto Hall. [24] .....	30
--	----

## ÍNDICE DE ABREVIATURAS

MC	Microcontrolador
CC	Corriente Continua
CA	Corriente Alterna
PWM	Modulación de ancho de pulso
BLDC	Motor CC sin escobillas
TM	Par motor
I	Intensidad De Inducido
W	Velocidad Angular

## INTRODUCCIÓN

Nuestro proyecto tiene como objetivo el uso de técnicas utilizadas en el control de motores BLDC sin sensores usando el microcontrolador ARM Cortex3 con 32 bits de LPCXpresso poniendo en práctica los conocimientos aprendidos durante el seminario de graduación de Microcontroladores Avanzados. Se usara el software LPCXpresso V2.0\_264 con lenguaje de programación "C" para el desarrollo de los programas que controlen nuestro proyecto.

En el capítulo 1 se dará una descripción general del proyecto, exponiendo los antecedentes, motivación, objetivos y limitaciones. El capítulo 2 cubrirá los aspectos principales de la fundamentación teórica y de las herramientas usadas para el desarrollo de nuestro proyecto. En el capítulo 3 se explicará el diseño preliminar del proyecto así mismo como las actividades desarrolladas para su realización. Y finalmente las pruebas, simulaciones y análisis de resultados serán cubiertos en el capítulo 4.

## **CAPÍTULO 1**

### **DESCRIPCIÓN GENERAL DEL PROYECTO**

En el siguiente capítulo detallaremos las características generales de nuestro proyecto, así como los objetivos planteados, motivos para la implementación de sistemas embebidos, explicaremos de que manera interactuamos con los proyectos de nuestros compañeros, los diferentes problemas que se presentan en cuanto al control de motores se refiere, así como las limitaciones a los que nos vemos sometidos.

## 1.1 ANTECEDENTES

Uno de los primeros sistemas embebidos reconocidos fue el sistema de guía de Apolo desarrollado por el laboratorio de desarrollo del MIT para las misiones Apolo hacia la luna. Cada vuelo hacia la luna tenía dos de estos sistemas. La función era manejar el sistema de guía inercial de los módulos de excursión lunar. En un comienzo fue considerado como el elemento que más riesgo presentaba en el proyecto Apolo. Este sistema de cómputo fue el primero en utilizar circuitos integrados y utilizaba una memoria RAM magnética, con un tamaño de palabra de 16 bits. El software fue escrito en el lenguaje ensamblador propio y constituía en el sistema operativo básico, pero capaz de soportar hasta ocho tareas simultáneas. Actualmente los sistemas embebidos los encontramos a cada comento de nuestra vida en el celular, televisores, microondas, equipos de audios, aviones, etc. Los sistemas embebidos a pesar de no ser muy conocidos están en muchas partes de nuestro diario vivir, en realidad, es difícil encontrar algún dispositivo en cuyo interior no esté basado en algún sistema embebido, desde aviones hasta teléfonos celulares e incluso en algunos electrodomésticos comunes como refrigeradoras y hornos microondas. Estos sistemas suelen tener en una de sus partes una computadora con características especiales conocida como micro controlador que viene a ser el cerebro del sistema [1].

Para aportar nuevas ideas al desarrollo de sistemas embebidos vamos a usar en este proyecto la tarjeta LPC1769 la cual es un sistema embebido y

que ha sido desarrollada en conjunto por las compañías NXP, LPCXpresso y Code\_Red. Usaremos esta tarjeta para el control de un motor BLDC sin sensores.

## **1.2 MOTIVOS PARA LA IMPLEMENTACIÓN DEL CONTROL DEL MOTOR BLDC**

Una de las principales motivaciones es la de implementar un control para un motor BLDC que se caracteriza por su mayor eficiencia en comparación con los motores DC convencionales, cabe recalcar que este trabajo lo realizamos conjuntamente con nuestros compañeros de manera grupal en el que cada uno de los grupos realiza una tarea específica persiguiendo un objetivo en común.

El mantener control sobre la operación del motor BLDC es una ventaja muy significativa, puesto que esto nos permite realizar ajustes a ciertos parámetros tales como sentido de rotación del eje del motor y la velocidad del mismo, lo que nos ayudará a encontrar diferentes usos y aplicaciones en las que podríamos incursionar tales como mantener control en un determinado proceso.

Uno de los motivos para la utilización de motores BLDC es que poseen un mayor tiempo de vida útil frente a los motores con escobillas debido a que en los motores BLDC en ausencia de escobillas no presentan el desgaste que es característico y muy común en los motores con escobillas, lo que implica

también la disminución de ruido y calor generado por el frecuente rozamiento.

Cabe recalcar que los motores BLDC al no poseer una limitación mecánica en su funcionamiento (debido a que la conmutación se basa en un circuito electrónico) tienen un rango de velocidad elevado [2].

### **1.3 IDENTIFICACIÓN DEL PROBLEMA**

La realización de nuestro proyecto se basa en mantener el control sobre un motor BLDC en parámetros tales como velocidad y sentido de rotación, utilizando dispositivos tales como el Motor Control Board y la tarjeta LPC1114. Nuestro principal problema se basa en como nuestros compañeros podrán interconectar sus prototipos con los nuestros ya que en su caso utilizan una tarjeta LPC1769 en conjunto con otros dispositivos para de esta manera mantener el control sobre el motor BLDC, por lo que será necesaria la implementación de algún sistema que nos permita la interconexión y a su vez la interacción entre ambos prototipos.

Generalizando un poco más, podemos ver que con el pasar del tiempo el control de procesos se realiza con una mayor rigurosidad y eficiencia en diferentes ámbitos, esto es comprensible ya que el avance de la tecnología que se realiza a pasos agigantados nos permite la disminución de pérdidas y ahorro de recursos que significan un aumento en la productividad y por ende en el crecimiento económico de cualquier entidad. En el día a día sin darnos

cuenta muchas veces interactuamos con sistemas de control de motores, sin embargo no nos percatamos de lo difícil que serían nuestras vidas en ausencia de estos aparatos. En la actualidad el uso de los motores va en aumento debido a las grandes necesidades del ser humano.

Cabe recalcar que en muchos de los casos el mantener control permanente sobre un motor es muy difícil, debido a que estos son aparatos eléctricos diseñados para mantener un trabajo constante con parámetros fijos determinados (tales como velocidad y sentido de rotación), es por esto que impera la necesidad de crear sistemas que nos permitan el control permanente de motores para de esta manera realizar procesos con una mayor eficiencia que se vea reflejado en diferentes aspectos como disminución del costo de mantenimiento, disminución de consumo de energía eléctrica, aumento de la vida útil de los aparatos.

En la actualidad este criterio ha sido tomado en cuenta en diferentes sectores industriales siendo un ejemplo palpable el mercado de la climatización y refrigeración el cual ya ha puesto en el mercado equipos de acondicionamiento de aire de alta eficiencia (ahorran hasta un 60% de energía eléctrica en comparación con los equipos normales) denominados Inverter los cuales utilizan motores BLDC controlados electrónicamente, de esta manera el potencial comprador se ve decantado por las bondades que ofrece dicho producto no solo en el aspecto económico sino también en el aspecto ecológico preservando el cuidado del planeta.

## 1.4 OBJETIVOS

El principal objetivo de nuestro trabajo es implementar los conocimientos adquiridos durante nuestra vida académica en lo que a microcontroladores se refiere, teniendo en cuenta varios aspectos tales como sus características, lenguaje de programación utilizado, alcances, dispositivos afines a su funcionamiento, ventajas que nos ofrece y demás aspectos indispensables que deben ser tomados en cuenta al usar un determinado microcontrolador.

Desarrollaremos un sistema de control de un motor BLDC con sensores utilizando los microcontroladores ARM Cortex3 con 32 bits de LPCXpresso LPC1114 y LPC1769 que conjuntamente trabajarán con el LPCXPRESSO MOTOR CONTROL BOARD el cual es una plataforma universal para control de motores en microcontroladores desarrollados por NXP.

Lograr comprender y utilizar de una manera eficiente el LPCXPRESSO MOTOR CONTROL BOARD, lo que implica el aprendizaje de las partes que lo componen así como sus bloques de funcionamiento, interfaces de comunicación disponibles y las diferentes ventajas que nos ofrece al utilizar o adherir en dicha tarjeta el microcontrolador LPC1114.

Tener control en el funcionamiento de un motor BLDC con sensores, con el fin de realizar un trabajo en específico, mediante un dispositivo de control podremos ordenar al motor que comience a trabajar, disminuya la velocidad,

aumente la velocidad, invierta el sentido de rotación y además que detenga su funcionamiento.

Lograr entender la utilidad y la importancia que tienen los sistemas embebidos así como las grandes ventajas que nos ofrecen en nuestra vida diaria, además de obtener gran cantidad de conocimientos acerca del funcionamiento de este tipo de sistemas y lo determinantes que pueden ser para el desarrollo de la tecnología.

Al desarrollar este tipo de proyectos uno de nuestros objetivos es tomar iniciativa para la elaboración de este tipo de sistemas que de alguna manera contribuyan al desarrollo de la tecnología en nuestro país incentivando al crecimiento económico del mismo.

## **1.5 LIMITACIONES**

Para el desarrollo de nuestro proyecto tendremos algunas limitaciones en aspectos de interés tales como trabajar por primera vez con los controladores de la familia ARM CORTEX de LPCXPRESSO lo que implica el aprendizaje de los bloques, componentes, dispositivos y arquitectura que comprenden cada uno de las tarjetas, tanto de la LPC1114 como de la LPC1769.

El trabajar con un dispositivo como el motor BLDC representa en cierto modo una limitación, ya que es un aparato que además de caracterizarse por su baja potencia de salida requiere del desarrollo de algún tipo de software para

su funcionamiento, por lo que es indispensable que dicho software sea correctamente elaborado, caso contrario el motor no funcionará y si lo hace no será de una manera adecuada.

Las tarjetas LPCXPRESSO contienen una serie de atributos o características, por ejemplo en el caso de la tarjeta LPC1114 es limitada en el número de pines disponibles para la interacción con el MOTOR CONTROL BOARD, es por esto que nos vemos con cierta limitación de acuerdo al tipo de tarjeta que utilicemos para el desarrollo de nuestro proyecto.

## **CAPÍTULO 2**

### **SUSTENTACION TEÓRICA**

#### **2.1 RESUMEN**

En el presente capítulo describiremos cada uno de los elementos que serán utilizados para el desarrollo de nuestro proyecto. La herramienta de Software será LPCXpresso 2.0\_2.64 y entre las herramientas de hardware tenemos: La tarjeta LPC1769, el Motor Control Kit y el motor BLDC con sensores las cuales serán interconectadas entre si de tal manera que la tarjeta LPC1769 envíe las señales de activación respectivas hacia el Motor Control Kit el cual

procesará dichas señales y logrará causar alguna respuesta en el Motor BLDC.

## **2.2 HERRAMIENTA DE SOFTWARE**

### **2.2.1 LPCXpresso 2.0\_2.64**

Es una herramienta de desarrollo para programar tarjetas creadas por la compañía LPCXpresso en la cual la programación se la realiza en el lenguaje C/C++ entre sus características principales podemos mencionar que posea un compilador paso a paso que nos permite ir observando línea por línea el desarrollo del programa.

El LPCXpresso IDE ha sido desarrollado por CodeRed junto a NXP. El mismo incluye un entorno de Eclipse específicamente adaptado para interactuar con la tarjeta.

Eclipse utiliza algunos conceptos comunes a otros programas de desarrollo por lo que vamos a ver algunos detalles.

Workspace es donde se encuentran nuestros proyectos. Estos proyectos pueden ser aplicaciones y/o bibliotecas. También almacena todas las configuraciones del entorno por lo que se puede mover muy fácilmente de computadora en computadora con solo almacenar una copia del el workspace.

El Proyecto puede ser de dos tipos. Biblioteca estática o una aplicación ejecutable. Aquí se contiene archivos de código fuente (.c), encabezados (.h) y cualquier otro archivo que se desee.

El workspace se encuentra ubicado en la parte encerrada en el cuadro de color rojo y el área encerrada en el rectángulo en azul corresponde a los proyectos [3].

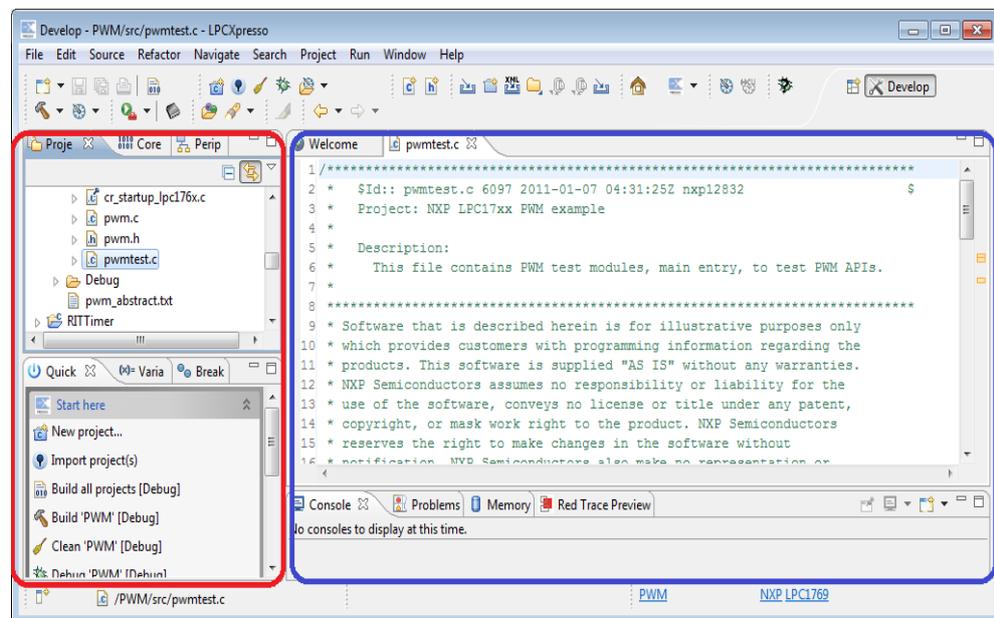


Figura 2.1 Vista del programa LPCXpresso 2.0\_2.64 [4]

## 2.3 HERRAMIENTAS DE HARDWARE.

### 2.3.1 INTRODUCCIÓN A LA TARJETA LPC1769

La tarjeta LPC1769 es un ARM Cortex M3 basado en microcontroladores para sistemas embebidos con un alto nivel de integración. El ARM Cortex M3 es la siguiente generación de procesadores usados en un sinnúmero de aplicaciones.

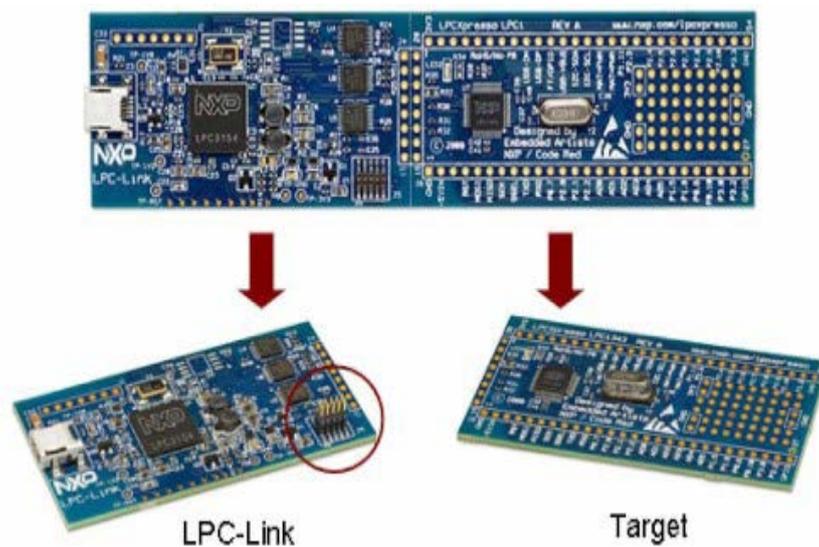


Figura 2.2 Tarjeta LPC1769 [5]

### 2.3.2 CARACTERÍSTICAS DE LA TARJETA LPC1769

Entre sus características principales podemos mencionar:

- Posee un procesador ARM Cortex M3 operando a velocidades de hasta 120Mhz.

- La memoria flash tiene una capacidad de 512 Kb.
- 64Kb de memoria SRAM.
- Posee interfaces de comunicaciones: SPI, RS232,I2C,Ethernet.
- Cuenta con comunicación USB OTG (Onthego) [4].

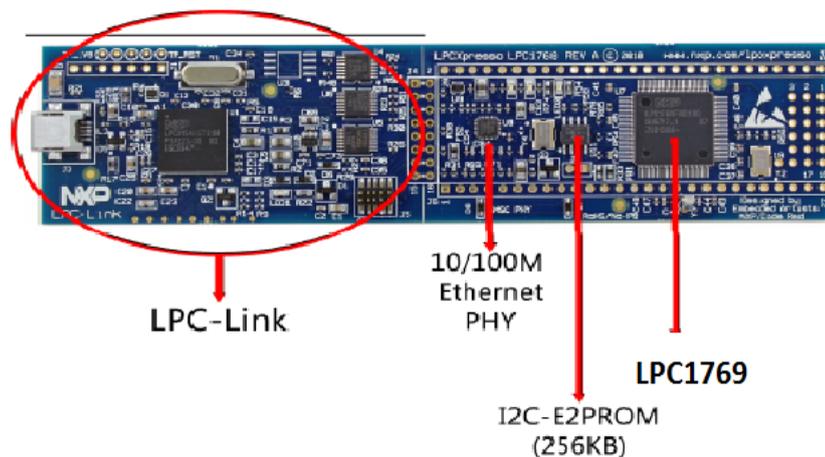


Figura 2.3 Características de la tarjeta LPC1769 [4]

### 2.3.3 INTRODUCCIÓN A LA TARJETA LPC1114

El LPC1114 es un procesador basado en ARM Cortex-M0, de bajo costo de 32-bit, diseñado para aplicaciones de microcontrolador 8/16-bit, que ofrece buen rendimiento, bajo consumo de potencia, simple conjunto de instrucciones y fácil direccionamiento de memoria, junto con el tamaño de código reducido en comparación con arquitecturas existentes 8/16-bit. El LPC1114 opera en frecuencias de la CPU de hasta 50MHz. El

complemento periférico de la LPC1114 incluye un máximo de 32kB de memoria flash, hasta 8 KB de memoria de datos, un modo de interfaz I2C-bus, una conexión para RS-485/EIA-485 UART, y posee dos interfaces SPI con las características del SSP, cuatro contadores / temporizadores de propósito general, un ADC de 10 bits, y 42 entradas o salidas I/O pins.

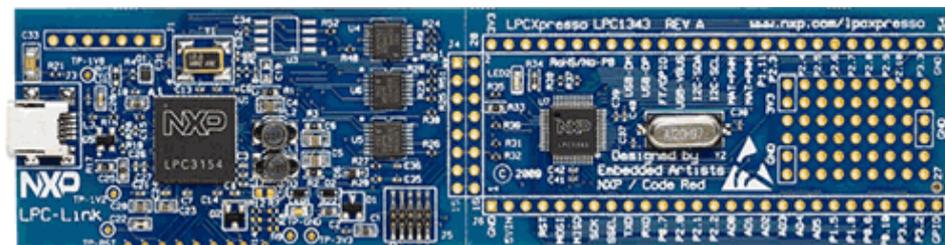


Figura 2.4 Tarjeta LPC1114 [5]

### 2.3.4 CARACTERÍSTICAS PRINCIPALES DE TARJETA LPC1114

Entre las principales características de la tarjeta LPC1114 podemos mencionar que posee:

- Microcontrolador NXP LPC1114 Cortex M-0 de hasta 50Mhz
- Posee 32Kb de memoria flash
- Y 8Kbde SRAM
- Tiene un I2C de alta velocidad
- Dos conexiones SSP y SPI
- Cuenta con 4 timer/Contadores
- Un convertidor ADC de 10 bits
- Y posee 42 pines de entrada/ salidas de propósito general [5].

### 2.3.5 MOTOR CONTROL

Se trata de una plataforma universal para el control de baja tensión del motor basado en microcontroladores de NXP. Con esta plataforma es posible controlar BLDC, BLAC, motores de paso y motores de corriente continua. El motor control LPCXpresso tiene una estructura tal como se indica en la imagen siguiente. El lado derecho contiene la electrónica de potencia para la conducción de las fases del motor.

El lado izquierdo es el lado de control con zócalos para tarjetas de LPCXpresso diferentes, así como una toma de control del procesador PLCC434. El lado izquierdo contiene también un OLED y una palanca de mando que puede servir como una interfaz de usuario para el sistema [6].

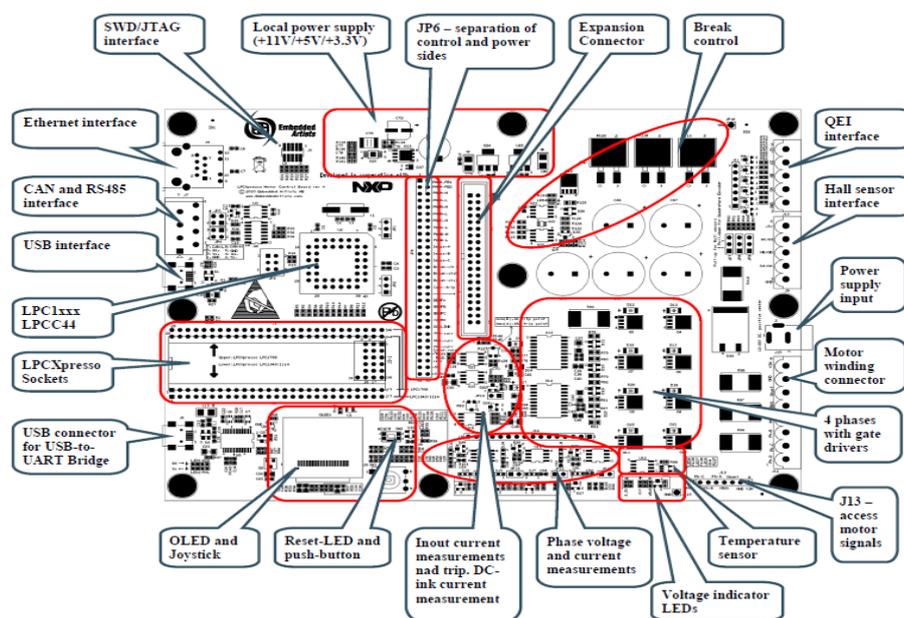


Figura 2.5 Vista de los principales componentes del Motor Control Board [6]

### **2.3.6 MOTORES BRUSHLESS DC (BLDC)**

Un motor DC sin escobillas, conocido más comúnmente como motores BLDC por sus abreviaturas en inglés: BRUSHLESS DIRECT CURRENT. En los últimos años estos motores tienen una gran demanda por sus beneficios.

La palabra brushless se puede traducir como "sin escobillas", las escobillas son los elementos que hacen contacto en el colector de un motor común. En los motores de DC más pequeños, son de una aleación de cobre y en motores más grandes son de un compuesto a base de carbón. Los motores brushless han derivado de los motores de los CD ROM, los DISCOS RIGIDOS y los ventiladores de computación, son motores trifásicos de alto rendimiento y bajo peso.

Estos motores carecen de colector y escobillas o carbones, entonces en vez de funcionar en DC funcionan en AC, la mayoría se alimentan con una señal trifásica, esta señal idealmente debería ser sinusoidal, pero en la práctica son pulsos, haciendo que la señal sea un continuo pulsante o bien un continuo con mucho componente de AC sin embargo se los clasifica como de DC porque al igual que los motores comunes tienen imanes permanentes.

Estos imanes son atraídos por la polaridad de un campo magnético generado en las bobinas, las cuales como decíamos reciben pulsos en un patrón específico. Si queremos que el motor gire más rápido, simplemente hacemos

girar el campo magnético secuencial a mayor velocidad. O lo que sería lo mismo a aumentar la frecuencia de los pulsos.

Estos motores sin escobillas permiten que consigamos un gran rendimiento y una gran potencia a cambio de un gran consumo. Debido a esto, tenemos que utilizar baterías con poca densidad de energía en comparación con otras pero con una característica única, pueden entregar muchísima potencia ya que los motores brushless necesitan del orden de 10 A a 11 v.



**Figura 2.6 Motor BLDC [7]**

La eliminación de las escobillas conlleva la necesidad de un circuito electrónico encargado de alimentar las distintas fases en función de la posición del eje y la de un sistema de sensores, tres por lo general, para detectar la posición del mismo. Estos sensores suelen ser del tipo Hall, sensibles al campo magnético, colocados en el devanado del estator y cerca de los imanes del rotor. La existencia de estos elementos presenta algunos inconvenientes, como son la menor fiabilidad y la complejidad de montaje de los sensores y de su cableado, lo que encarece la fabricación del motor. Por todo ello, en los últimos años se han desarrollados sistemas de detección

“sin sensores” para aquellas aplicaciones donde no se requiere par motor a muy baja velocidad. Las técnicas se basan en determinar la posición rotórica a partir de las tensiones inducidas en los devanados del motor. Como quiera que en la fase de arranque no se dispone de tensiones generadas por el movimiento, se recurre a distintas técnicas de puesta en marcha dependiendo de la aplicación; todas ellas basadas en microcontroladores o en circuitos integrados específicos.

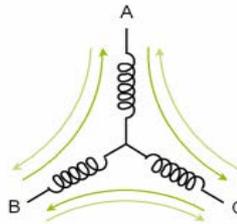
Los motores eléctricos solían tener un colector de delgas o un par de anillos rozantes. Estos sistemas, que producen rozamiento, disminuyen el rendimiento, desprenden calor y ruido, requieren mucho mantenimiento y pueden producir partículas de carbón que manchan el motor de un polvo que, además, puede ser conductor.

Existen dos tipos de motores brushless: el motor brushless trapezoidal y el motor brushless senoidal, el principio de funcionamiento es el mismo, la diferencia radica en la forma de alimentar los devanados del estator.

#### **2.3.6.1 MOTOR BRUSHLESS DE CONMUTACIÓN TRAPEZOIDAL**

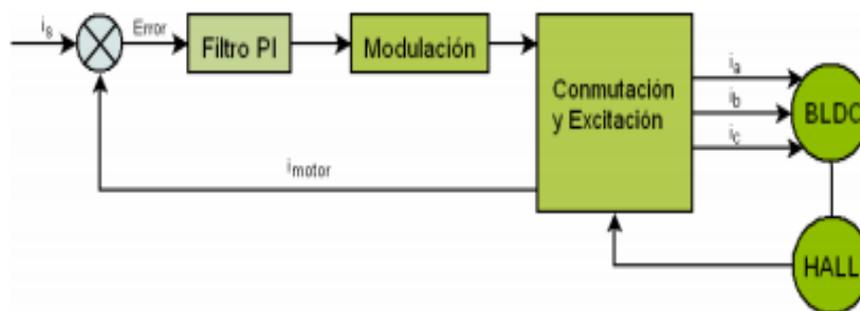
Uno de los métodos más simples de control de motores brushless es el llamado conmutación trapezoidal. En este esquema se controla la corriente que circula por los terminales del motor, excitando un par simultáneamente y manteniendo el tercer terminal desconectado. Sucesivamente se va

alternando el par de terminales a excitar hasta completar las seis combinaciones posibles.



**Figura 2.7 Bobinado de un motor BLDC conexión en estrella BLDC [8]**

Tres sensores de efecto hall situados en el motor son utilizados para proporcionar la posición aproximada del rotor al controlador y que éste pueda determinar el próximo par de terminales a excitar. La siguiente figura muestra el diagrama de bloques de un controlador trapezoidal típico con lazo cerrado de corriente.



**Figura 2.8 Diagrama de bloques de controlador con conmutación trapezoidal [8]**

La corriente que circula por el par de terminales activos es comparada con la corriente deseada y el error resultante es aplicado a un Filtro PI (Proporcional - Integrador). La salida de este filtro intenta corregir la desviación y por tanto minimizar el error. Con esta técnica se consigue mantener constante la corriente que circula por cualquiera de los bobinados del motor.

La conmutación trapezoidal proporciona una primera aproximación al control de motores brushless. Gracias a su sencilla implementación y a los pocos recursos utilizados es ampliamente usada en aplicaciones de muy bajo coste. No obstante debido a su alto rizado de par en todo el espectro frecuencial la hace desaconsejable para cualquier aplicación que demande una mínima precisión o eficiencia. La conmutación sinusoidal soluciona el problema del rizado del par a cambio de aumentar la complejidad del control y de incorporar un sensor de mayor precisión. Sin embargo, debido a que trabaja en el espacio variante del tiempo presenta una limitación de control a altas velocidades.

Existen distintas técnicas de modulación orientadas a la generación de señales de excitación para motores Brushless mediante las cuales, se puede aumentar la eficiencia del sistema.

Debido a que en todo momento las corrientes de dos bobinados son iguales en magnitud y la tercera siempre es nula, el vector de corrientes del estator o



En cambio para la construcción del motor brushless senoidales diferente del trapezoidal en que el flujo del entrehierro es senoidal y está generado por los imanes del rotor que tiene una forma especial, también los devanados tienen una distribución senoidal, mientras que en el trapezoidal los devanados están concentrados [8].



**Figura 2.10 Bobinado del Rotor de un Motor BLDC [9]**

### **2.3.6.3 FÓRMULAS MATEMÁTICAS PARA LOS MOTORES BRUSHLESS**

En un motor de corriente continua con escobillas, se obtiene par motor gracias a la interacción del campo magnético inductor, estacionario, y la intensidad del arrollamiento inducido giratorio. Campo y corriente eléctrica se mantienen siempre en la misma posición relativa gracias al mecanismo de conmutación formado por el colector de delgas y las escobillas. En motores de pequeña potencia suele obtenerse la excitación mediante imanes

permanentes. En este caso, solo se dispone de dos terminales para el control y la alimentación del motor. Las relaciones básicas electromecánicas son en este caso las siguientes:

$$T_m = K \cdot i$$

$$E = K \cdot \Omega$$

Siendo,  $T_m$ : Par motor;  $i$ : intensidad de inducido;  $E$ : tensión inducida;  $\Omega$ : velocidad angular

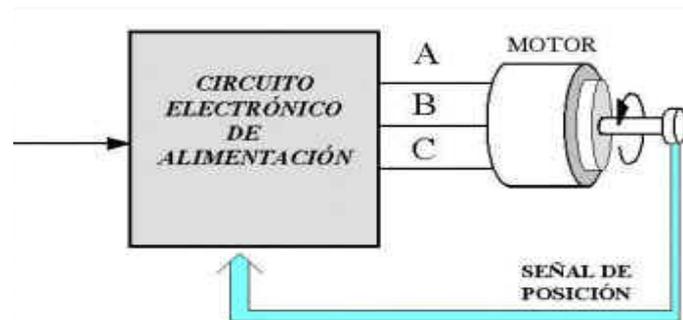
El hecho de tener control directo sobre el par mediante la intensidad de inducido, y sobre la velocidad a través de la tensión, convierte a este motor en el modelo de referencia para la regulación de velocidad. No obstante, la alimentación del inducido a través de las escobillas y el colector presenta muchos inconvenientes, hasta el punto que en algunos casos se hace inviable su utilización.

El motor que nos ocupa es similar al de corriente continua con escobillas, con las siguientes salvedades:

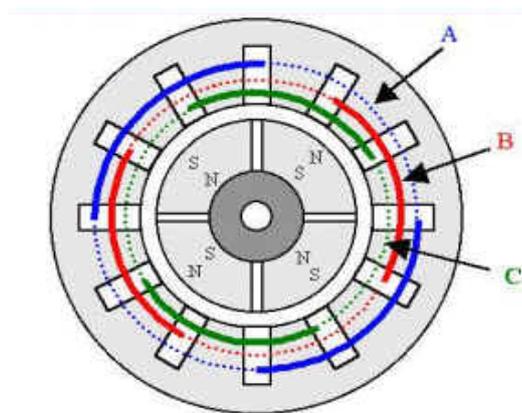
- a) La conmutación se realiza de forma electrónica en lugar de mecánica.
- b) Los imanes permanentes van alojados en el rotor en lugar de en el estator.

c) Las bobinas van alojadas en el estator, constituyendo un devanado monofásico o polifásico.

Su funcionamiento se basa en la alimentación secuencial de cada una de las fases del estator de forma sincronizada con el movimiento del rotor. De esta forma, los imanes permanentes siguen el movimiento del campo magnético estático, cuyo desplazamiento depende a su vez del giro del rotor.



**Figura 2.11 Circuito de Alimentación del Motor BLDC [10]**



**Figura 2.12 Esquema de un motor BLDC con sensor [10]**

El auge de estos motores en los últimos años no se debe tanto al desarrollo de nuevas técnicas como al abaratamiento de los componentes electrónicos, tanto de control como de potencia. También, se han beneficiado de la reducción de precios de los imanes permanentes de altas prestaciones a base de tierras raras como el Neodimio o el Samario (Nd-Fe-B, y Sm-Co). Las características magnéticas de estos imanes permiten reducir de forma significativa el peso de los motores en comparación con los que incorporan los imanes convencionales de ferrita [10].

### 2.3.7 LOS SENSORES EFECTO HALL

El efecto Hall consiste en la aparición de un campo eléctrico en un conductor cuando es travesado por un campo magnético. A este campo eléctrico se le llama campo Hall. Este efecto fue descubierto en 1879 por el físico estadounidense Edwin Herbert Hall.

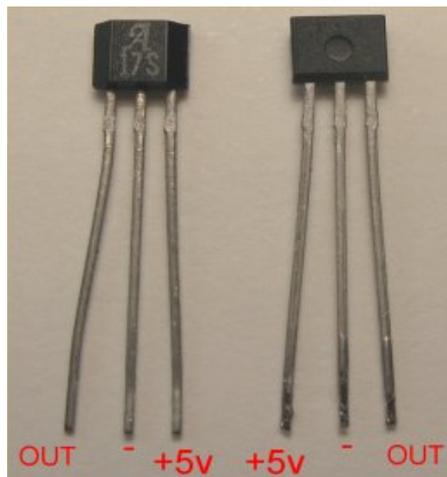


Figura 2.13 Sensores de Efecto Hall [11]

Los sensores basados en efecto Hall constan de un elemento conductor o semiconductor y un imán. Cuando un objeto ferromagnético se aproxima al sensor, el campo que provoca el imán en el elemento se debilita. Así se puede determinar la proximidad de un objeto, siempre que sea ferromagnético [12].

### 2.3.8 MOTORES BLDC CON SENSORES DE EFECTO HALL

Gracias a este principio, mediante un disco magnético acoplado al eje del motor brushless podemos sensar la posición del rotor. Principalmente, estos sensores se usan para que la electrónica pueda conmutar las tres bobinas del motor de acuerdo a la posición de los polos del imán del rotor. Así por ejemplo un motor brushless de dos polos con 3 sensores hall (a  $120^\circ$ ), tiene una resolución en posición de 6 pulsos por vuelta ( $60^\circ$  de conmutación).

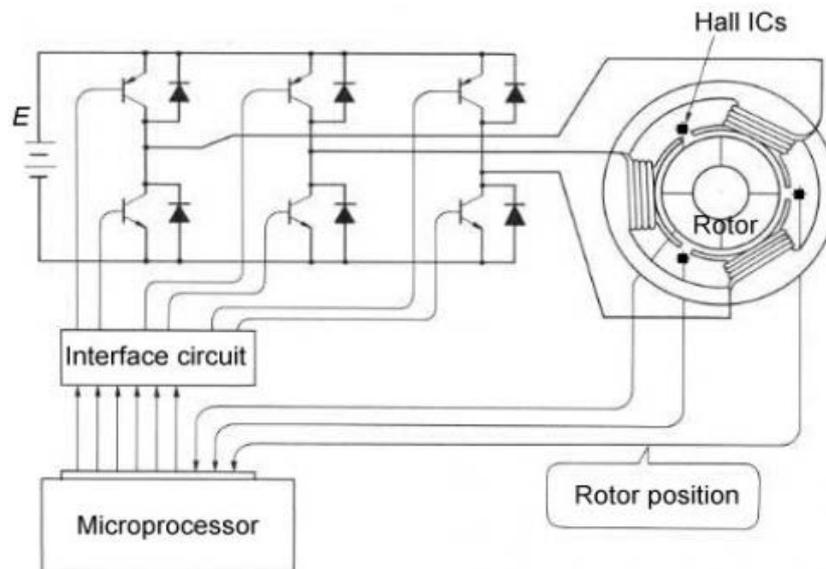


Figura 2.14 Motor BLDC con sensores de efecto Hall [13]

En caso de los motores brushless multipolares, esta resolución aumenta.

Ocasionalmente, se pueden accionar los motores brushless sin sensores Hall para determinadas aplicaciones muy sencillas, como por ejemplo ventiladores y bombas [14].

### **2.3.9 MOTORES BLCD SIN SENSORES DE EFECTO HALL.**

En los motores BLDC se miden la fuerza contra electromotriz FEM en las bobinas para deducir la posición del rotor, eliminando la necesidad de sensores separados de efecto Hall, y por lo tanto a menudo se llaman los reguladores “sensorless”. Como un motor de CA, el voltaje en las bobinas es sinusoidal, pero sobre una conmutación entera la salida aparece trapezoidal debido a la salida de la C.C. del regulador.

El regulador contiene 3 bidireccionales para conducir potencia de CC. De gran intensidad, que son controladas por un circuito lógico. Los reguladores simples emplean comparadores para determinarse cuando la fase de salida debe ser de avanzada, mientras que reguladores más avanzados emplean a microcontroladores para manejar la aceleración, la velocidad del control y eficiencia. Los reguladores que detectan la posición del rotor basada en la back FEM tienen desafíos del suplemento en iniciar el movimiento porque no se produce ningún back-FEM cuando el rotor es inmóvil. Esto es lograda generalmente comenzando la rotación a partir de una fase arbitraria, y después saltando a la fase correcta si se encuentra para ser incorrecto. Esto

puede hacer el motor funcionar brevemente al revés, agregando aún más complejidad a la secuencia de lanzamiento.

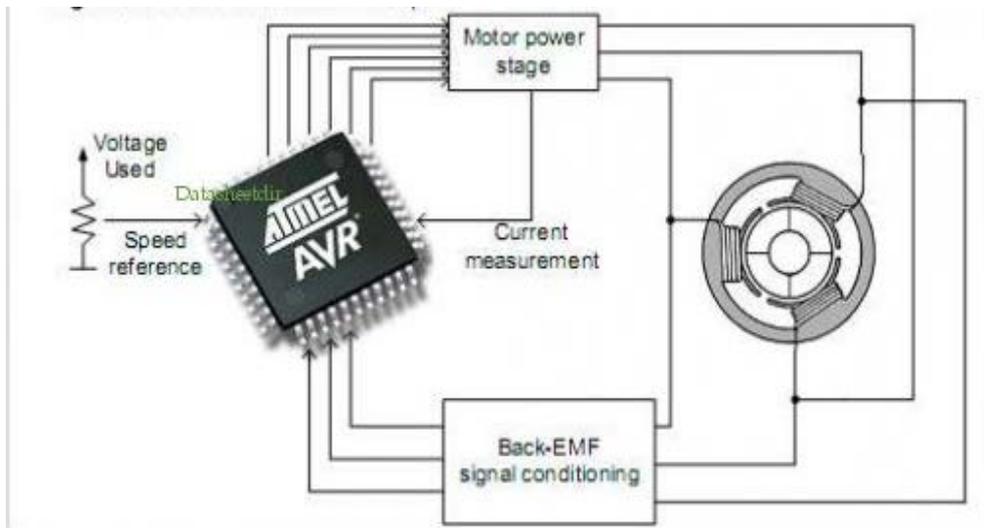


Figura 2.15 Motor BLDC sin sensores [15]

### 2.3.10 VENTAJAS DE USAR MOTORES BLDC SIN SENSORES

Entre las principales ventajas podemos mencionar:

- Fiabilidad, especialmente la industria aeroespacial y militar
- Menor espacio físico
- Costos de fabricación bajo especialmente en sistemas de baja potencia.
- Mide el tiempo en las transiciones cada 60 grados.
- La conmutación se produce después de los 30 grados del flanco ascendente o descendente de la BFEM.
- Estos motores funcionan mejor cuando no sean necesarios los cambios de velocidad rápidos

### 2.3.11 PRINCIPALES DIFERENCIAS ENTRE MOTORES SIN ESCOBILLAS Y MOTORES CON ESCOBILLAS

A continuación se presenta una tabla comparativa entre los motores DC con escobillas y los motores DC sin escobillas.

MOTOR BRUSHLESS(sin escobillas)	MOTOR NORMAL(con escobillas)
<ul style="list-style-type: none"> <li>• <b>Conmutación</b></li> </ul>	
Conmutación electrónica basada en sensores de efecto Hall	Conmutación por escobillas
<ul style="list-style-type: none"> <li>• <b>Mantenimiento</b></li> </ul>	
Mínimo	Periódico
<ul style="list-style-type: none"> <li>• <b>Durabilidad</b></li> </ul>	
Mayor	Menor
<ul style="list-style-type: none"> <li>• <b>Curva velocidad/par</b></li> </ul>	
Plana Operación a todas las velocidades con la carga	Moderada A altas velocidades la fricción definida de las escobillas se incrementa reduciendo el par
<ul style="list-style-type: none"> <li>• <b>Eficiencia</b></li> </ul>	
Alta sin caída de tensión por las escobillas	Moderada
<ul style="list-style-type: none"> <li>• <b>Potencia de salida/tamaño</b></li> </ul>	
Alta Menor tamaño y debido a sus mejores características térmicas porque los bobinados están en el estator, que al estar en la carcasa tiene una mejor disipación de calor	Bajo El calor producido en la armadura es disipado en el interior aumentando la temperatura y limitando las características
<ul style="list-style-type: none"> <li>• <b>Inercia del rotor</b></li> </ul>	
Baja Debido a los imanes permanentes en el rotor	Alta Limita las características dinámicas
<ul style="list-style-type: none"> <li>• <b>Rango de velocidad</b></li> </ul>	
Alto Sin limitaciones mecánicas	Bajo El límite lo imponen

impuestas por escobillas/conmutador.	principalmente las escobillas
• <b>Ruido eléctrico generado</b>	
Bajo	Arcos en las escobillas
• <b>Coste de construcción</b>	
Alto debido a los imanes permanentes	Bajo
• <b>Control</b>	
Complejo y caro	Simple y barato

**Tabla 1.1 Motor BLDC con sensores de efecto Hall. [16]**

### **2.3.12 VENTAJAS DEL USO DE LOS MOTORES BRUSHLESS**

Los motores brushless tienen muchas ventajas por sobre los motores brushed (con escobillas) entre ellas son:

- Mayor rendimiento (mayor duración de las baterías para la misma potencia)
- Mayor eficiencia (menos pérdida por calor)
- Menor peso para la misma potencia
- Conmutación electrónica basada en sensores de posición de efecto Hall
- Requieren menos mantenimiento al no tener escobillas
- Relación velocidad/par motor es casi una constante
- Mayor potencia para el mismo tamaño
- Mejor disipación de calor
- Rango de velocidad elevado al no tener limitación mecánica.
- Menor ruido electrónico (menos interferencias en otros circuitos) [17].

### **2.3.13 DESVENTAJAS DEL USO DE LOS MOTORES BRUSHLESS**

- Mayor costo de construcción
- El control es mediante un circuito caro y complejo
- Siempre hace falta un control electrónico para que funcione, que a veces duplica el costo

## **CAPÍTULO 3**

### **DESARROLLO DEL PROYECTO**

En el siguiente capítulo detallaremos los ejercicios que hemos desarrollado para lograr obtener el control del movimiento del motor BLDC, utilizando la tarjeta LPC1114 y LPC1769, así como detallaremos los algoritmos y hardware empleados.

### 3.1 ENCENDIDO ROTATORIO DE LEDS

#### 3.1.1 DESCRIPCIÓN GENERAL

Nuestro dispositivo principal es la tarjeta LPC1769 quien enviará las diferentes señales de voltaje a través del puerto dos para de esta manera encender cada uno de los diodos led en el orden indicado. Además de esto una botonera será conectada a la tarjeta para que esta detecte el momento en que el sentido de rotación de los diodos led debe ser cambiado.

#### 3.1.2 DIAGRAMA DE BLOQUES

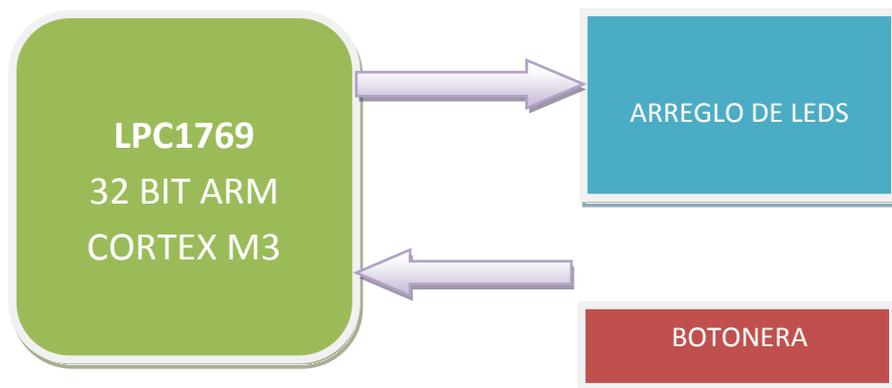


Figura 3.1 Diagrama de Bloques Encendido Rotatorio de Leds

### 3.1.3 DIAGRAMA DE FLUJO

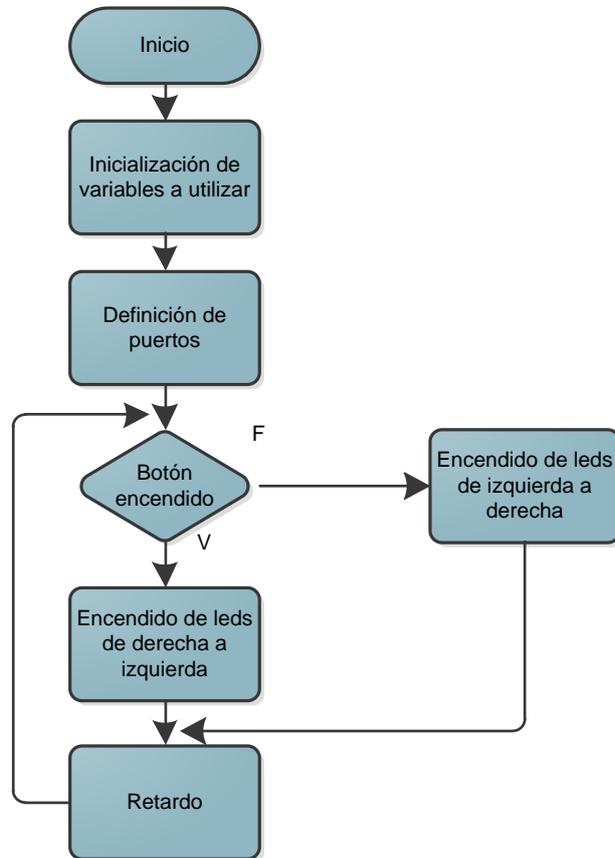


Figura 3.2 Diagrama de Flujo Encendido Rotatorio de Leds

### 3.1.4 DESCRIPCIÓN DEL ALGORITMO

- 1) Inicialización de variables.
- 2) Definición de puertos a utilizar declarándolos ya sea como entradas o salidas en cada uno de sus respectivos pines.
- 3) Seteo de los valores en cada pin.
- 4) Ingreso en un lazo infinito.

- 5) Pregunta si la botonera que está conectada en el pin 8 ha sido activada.
- 6) Si la botonera no ha sido activada los bits se encienden de derecha a izquierda.
- 7) Si la botonera ha sido activada los bits en la próxima secuencia se encenderán de izquierda a derecha.
- 8) Los pasos 5,6 y 7 se repiten indefinidamente ya que se encuentran en el lazo infinito.

### 3.1.5 PROGRAMA PRINCIPAL

```

/*****
 * Autores: Emerson Chang / Oscar Saigua
 * Nombre: Rotación de LEDS
 *
 *****/
Descripción
 * El siguiente código fuente nos permite mostrar un arreglo de 8 diodos
 * mediante el Puerto 2, de esta manera el encendido de los LEDS será en
 * un determinado sentido. Cuando el usuario presiona la botonera, el
 * encendido de los LEDS cambia de sentido.
 *****/

#include<cr_section_macros.h>
#include <NXP/crp.h>

// Variable to store CRP value in. Will be placed automatically
// by the linker when "Enable Code Read Protect" selected.
// See crp.h header for more information
__CRP const unsigned int CRP_WORD = CRP_NO_CRP ;
#include "lpc17xx.h"
#include "type.h"

int main (void)
{
  uint32_t i, j;

```

```

/* SystemClockUpdate() updates the SystemFrequency variable */
SystemClockUpdate();
LPC_GPIO2->FIODIR = 0xFFFFFEFF;          /* P2.xx defined as
Outputs */
LPC_GPIO2->FIOMASK= 0x00000000;
LPC_GPIO2->FIOCLR = 0xFFFFFFFF;          /* turn off all the LEDs */
LPC_GPIO2->FIOCLR = 0xFFFFFFFF;
while(1)
{
if (LPC_GPIO2->FIOPIN==0x00000000)
    {
        for(i = 0; i < 8; i++)
        {
            LPC_GPIO2->FIOSET = 1 <<i;
            for(j = 1000000; j > 0; j--);
        }
        LPC_GPIO2->FIOCLR = 0xFFFFFFFF;
        for(j = 1000000; j > 0; j--);
    }

if (LPC_GPIO2->FIOPIN==0x00000100)
    {
        for(i = 8; i > 0; i--)
        {
            LPC_GPIO2->FIOSET = 1 << (i-1);
            for(j = 1000000; j > 0; j--);
        }
        LPC_GPIO2->FIOCLR = 0xFFFFFFFF;
        for(j = 1000000; j > 0; j--);
    }
}
}
}

```

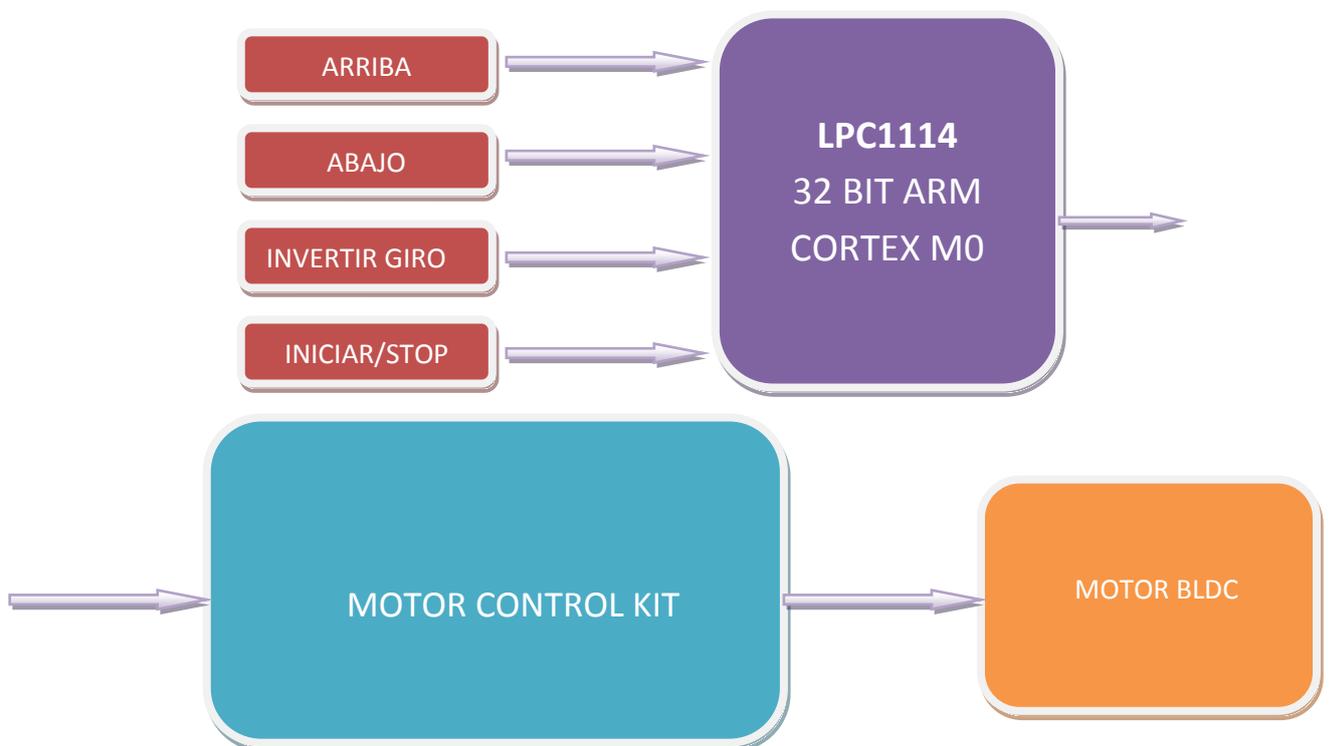
## 3.2 CONTROL DE MOTOR BLDC MEDIANTE LA TARJETA LPC1114

### 3.2.1 DESCRIPCIÓN GENERAL

En este caso la tarjeta LPC1114 es el dispositivo que se encarga de controlar el motor BLDC, ya sea para controlar su velocidad, sentido de giro, para iniciar o detener su movimiento, este control es logrado gracias a un conjunto

de botoneras que nos permiten obtener una respuesta inmediata en el motor BLDC. El Motor Control Board trabaja conjuntamente con la tarjeta LPC1114 para mantener un control constante sobre el motor, controlando parámetros básicos como la velocidad máxima, velocidad mínima, temperatura, etc.

### 3.2.2 DIAGRAMA DE BLOQUES



**Figura 3.3 Diagrama de Bloques Control de Motor BLDC mediante la tarjeta LPC1114**

### 3.2.3 DIAGRAMA DE FLUJO PROGRAMA PRINCIPAL MAIN.C

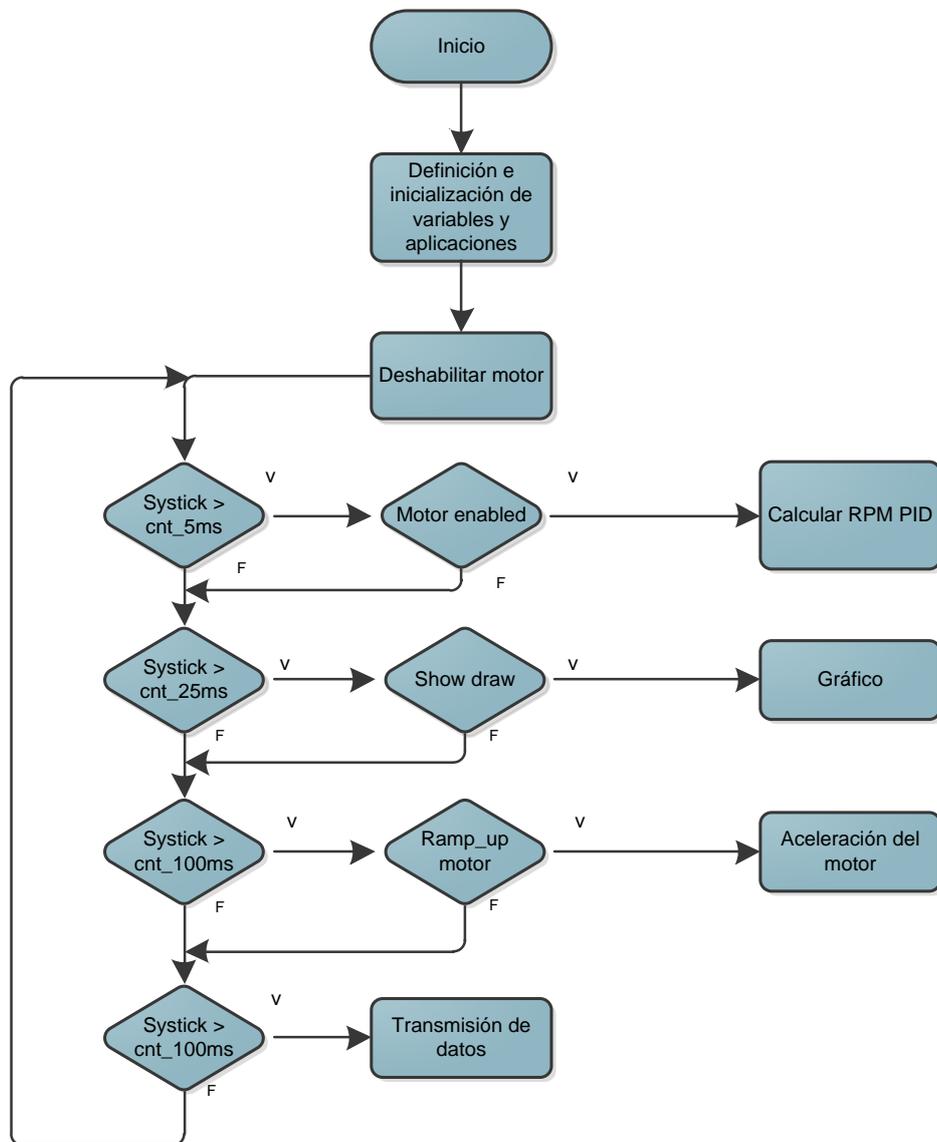


Figura 3.4 Diagrama de Flujo Control de Motor BLDC mediante la tarjeta LPC1114

### 3.2.4 DESCRIPCIÓN DEL ALGORITMO

- 1) Definición de variables.
- 2) Definición de puertos a utilizar.
- 3) Al cumplirse la primera condición de que la variable `systick_cntr` sea mayor a la variable `cnt_5ms` el programa consulta si el motor se encuentra habilitado y sin movimiento para posteriormente realizar el proceso PID.
- 4) En el caso de que `systick_cntr` es mayor a la variable `cnt_25ms`, el programa nos permite mostrar un pixel en la pequeña pantalla denominada Oled, esto de acuerdo a la velocidad que posee el motor, conforme se va formando la gráfica.
- 5) Si `systick_cntr` es mayor a `cnt_100ms`, el motor se encuentra habilitado y en movimiento, el programa permite aumentar gradualmente la velocidad del motor hasta el límite dado por el setpoint.
- 6) En el caso de que `systick_cntr` sea mayor a `cnt_1s`, se envía la estructura tipo motor.
- 7) Los pasos 3 hasta el 6 se repiten indefinidamente ya que se encuentran en un lazo infinito.

### 3.2.5 PROGRAMA PRINCIPAL

```

/*****
* $Id::BLDC.c 3871 2010-07-16 11:50:22Z gerritdewaard      $
*
* Nombre: Main
* Autor: Code Red Technologies
* Description:
* Programa principal que nos permite manejar en un lazo infinito el
* motor BLDC utilizando determinados procedimientos con la ayuda
* del motor control board, además nos permite mostrar en pantalla
* la gráfica de la velocidad variable del motor.
*****/

#include "application.h"

#define GRAPH_LIMIT    64-8
/*****
* External global variables
*****/
externvolatile uint32_t SysTick_cntr;
externvolatileMOTOR_TypeDef Motor;
//extern volatile RX_PACKET RxPackets[RX_MAX_PACKETS];
/*****
* Local variables
*****/
volatile SYNC_STRUCT sync;
volatile BLDC_STRUCT bldc;
volatile uint8_t blink = 0;
volatile uint8_t TXBUF[0xFF];
volatile uint8_t RXBUF[0xFF];
uint16_t connTimer=0;           // Timer for connection timeout
volatile uint8_t oled_X = 0;
volatile uint8_t oled_Y = 0;
volatile uint8_t oled_Y_PixBUF[96];

volatile uint8_t show_graph = FALSE;
volatile uint8_t show_temperature = FALSE;
volatile uint8_t clear_screen = TRUE;
volatile uint8_t sec5 = 5;

/*****

```

```

* Local Functions
*****/
voidMemCopyBYTE(uint8_t *src, uint8_t *dst, uint32_t count);
voidMemCopyDWORD(uint32_t *src, uint32_t *dst, uint32_t count);
voidInitSyncPacket (void);
voidProcessRxPacket(uint8_t index);

/*****
** Functionname: Call_5ms
** Descriptions: procedimiento que realiza el cálculo PID
** Parámetros: ninguno
** Valor a Retornar: ninguno
*****/
voidCall_5ms(void)
{
    if (Motor.Enable&& (Motor.RampingUp == FALSE))
    {
        /* Do the PID calculations */
        vPID_RPM(&Motor);
    }
}

/*****
** Functionname: Call_25ms
** Descriptions: procedimiento que muestra un punto de la grafica de velocidad del motor
** Parámetros: ninguno
** Valor a Retornar: ninguno
*****/
voidCall_25ms(void)
{
#if MC_BOARD_ENABLE_OLED == 1
    if (show_graph == TRUE)
    {
        /* Down scale the actual RPM value to screen size */
        oled_Y = (Motor.RPM/80);

        /* Clip the output */
        if (oled_Y>= GRAPH_LIMIT) oled_Y = GRAPH_LIMIT;

        /* Clear the previous pixel */
        oled_putPixel(oled_X, oled_Y_PixBUF[oled_X] ,
OLED_COLOR_BLACK);

```

```

        oled_Y = 64 - oled_Y;
        oled_putPixel(oled_X, oled_Y , OLED_COLOR_WHITE);

        oled_Y_PixBUF[oled_X] = oled_Y;
        oled_X++;

        if (oled_X == 96) oled_X = 0;
    }
#endif
}
/*****
** Functionname: Call_100ms
** Descriptions: procedimiento que aumenta o disminuya la velocidad del
** motor.
** Parámetros: ninguno
** Valor a Retornar: ninguno
*****/
voidCall_100ms(void)
{
    if (Motor.Enable&&Motor.RampingUp)
    {
        if (clear_screen == TRUE)
        {
            /* Clear the OLED screen */
            oled_clearScreen(OLED_COLOR_BLACK);
            clear_screen = FALSE;
        }
        /* Ramp-up the motor.. */
        vBLDC_RampUp (&Motor, Motor.sp);
    }
}
/*****
** Functionname: Call_1s
** Descriptions: procedimiento que nos muestra la temperature en
** pantalla.
** Parámetros: ninguno
** Valor a Retornar: ninguno
*****/
voidCall_1s(void)
{
    /* */
    vCOMMS_send(&Motor);
}

```

```

#if (MC_BOARD_ENABLE_OLED == 1)
#if (MC_BOARD_ENABLE_LM75 == 1)
static uint8_t firstTime = 1;
    uint8_t buffer[13] = "Temp: xx.x C";
    uint16_t temperature;

    if (show_temperature == TRUE)
    {
        if (firstTime)
        {
            /* Clear the OLED screen */
            oled_clearScreen(OLED_COLOR_BLACK);
            oled_putString(1, 1, (char *)buffer,
OLED_COLOR_WHITE, OLED_COLOR_BLACK);
            firstTime = 0;
        }

        temperature = (uint16_t)lm75a_readTemp();
        temperature = (temperature + 5) / 10; //round to 0.1 degree
        buffer[6] = ((temperature / 100) % 10) + '0';
        buffer[7] = ((temperature / 10) % 10) + '0';
        buffer[8] = '\0';
        oled_putString((6*6), 1, (char *)&buffer[6],
OLED_COLOR_WHITE, OLED_COLOR_BLACK);
        buffer[9] = (temperature % 10) + '0';
        buffer[10] = '\0';
        oled_putString((9*6), 1, (char *)&buffer[9],
OLED_COLOR_WHITE, OLED_COLOR_BLACK);
    }
    elsefirstTime = 1;

#endif //MC_BOARD_ENABLE_OLED
#endif //MC_BOARD_ENABLE_LM75
}

```

```

/*****
** Nombre de la funcion: Appl_Init
** Descripción: Función que permite la inicialización de diversas
** aplicaciones.
** Parametros:Ninguno
** Valor a retornar:Ninguno
*****/
voidAppl_Init (void)

```

```

{
  /******
  /* SYSTICK
    */
  /******
  SysTick_Config (SysTick_VALUE);

  /******
  /* FREERUNNING COUNTER
    */
  /******
  /* Setup Timer16 1 as free running counter for e.g. RPM calc. */
  /* Enable the clock to Timer16 1 */
  LPC_SYSCON->SYSAHBCLKCTRL |= (1<<8);
  /* Set the prescaler to get never get an overflow */
  LPC_TMR16B1->PR = 55;
  /* RESET timer16 1 */
  LPC_TMR16B1->TCR = 1<<1;
  /* ENABLE timer16 1 */
  LPC_TMR16B1->TCR = 1;
  /******
  /* BLDC MOTOR INIT
    */
  /******
  vBLDC_Init(&Motor);

  /******
  /* HALL sensors INIT
    */
  /******
  GPIOInit();
  GPIOSetDir( LED_PORT, LED_BIT, 1 );
  GPIOSetValue( LED_PORT, LED_BIT, 0 );
  /* Set the direction to input */
  GPIOSetDir(HALL_A_PORT,HALL_A_PIN,0); // HALL A
  GPIOSetDir(HALL_B_PORT,HALL_B_PIN,0); // HALL B
  GPIOSetDir(HALL_C_PORT,HALL_C_PIN,0); // HALL C

  /* Setup the interrupt, seq: portNum, bitPosi, sense = edge, single =
  both, event*/
  GPIOSetInterrupt(HALL_A_PORT,HALL_A_PIN,0,1,1);
  GPIOSetInterrupt(HALL_B_PORT,HALL_B_PIN,0,1,1);
  GPIOSetInterrupt(HALL_C_PORT,HALL_C_PIN,0,1,1);

```

```

    /* Enable the interrupts */
    GPIOIntEnable(HALL_A_PORT,HALL_A_PIN);      // HALL A
    GPIOIntEnable(HALL_B_PORT,HALL_B_PIN);      // HALL B
    GPIOIntEnable(HALL_C_PORT,HALL_C_PIN);      // HALL C

    NVIC_EnableIRQ(EINT2_IRQn);
    /*******/
    /* I2C INIT
    */
    /*******/
I2CInit( (uint32_t)I2CMaster);

    /*******/
    /* OLED INIT
    */
    /*******/
#if MC_BOARD_ENABLE_OLED == 1
    /* Initialize the OLED */
    oled_init();
    /* Clear the OLED screen */
    oled_clearScreen(OLED_COLOR_BLACK);
#endif //MC_BOARD_ENABLE_OLED

#if USE_STARTUP_GUI == 1
    /*******/
    /* Startup message on OLED
    */
    /*******/
    vGUI_StartupMessage();
#endif //USE_STARTUP_GUI

    /*******/
    /* JOYSTICK INIT
    */
    /*******/
#if MC_BOARD_ENABLE_JOYSTICK == 1
    joystick_init();
#endif

    /* CAN ROM DRIVER INIT
    */
    /*******/
#if (USE_CAN == 1)
    /* Initialize the CAN ROM driver handlers */

```

```

        vCAN_initRomHandlers();
#endif
        /******
        /* UART INIT
        /******
        /* Initialize the UART */
#if (USE_UART == 1)
        /* Push UART_RS485 high for UART -> USB mode */
        GPIOSetDir(3,4,1);
        GPIOSetValue(3,4,1);
        /* Initialize UART */
        UARTInit(115200);
        /* Add a 5 sec delay before sending startup message to UART. */
        /* Delay added to allow for USB enumeration on PC after reset. */
        SysTick_cntr = 0;
        while (SysTick_cntr > 5000);
        SysTick_cntr = 0;

        UARTSend((uint8_t *)"\n\r/*****",
50);
        UARTSend((uint8_t *)"\n\r/*", 50);
        UARTSend((uint8_t *)"\n\r/* BLDC Motor Control Demo Application on
*/", 50);
        UARTSend((uint8_t *)"\n\r/* LPCXpresso Motor Control Board
*/", 50);
        UARTSend((uint8_t *)"\n\r/* Date: 2011-02-14, rev A",
50);
        UARTSend((uint8_t *)"\n\r/*", 50);
        UARTSend((uint8_t *)"\n\r/*****",
50);
        UARTSend((uint8_t *)"\n\rRPM\tMV\tActive\tDirection\n\r", 27);
#if (USE_NXP_GUI == 1)
        /******
        /* PACKET INITIALIZATION
        */
        /******
        InitSyncPacket();
#endif
#endif // (USE_UART == 1)
}
void MemCopyBYTE(uint8_t *src, uint8_t *dst, uint32_t count)
{
    while(count--)*dst++ = *src++;
}

```

```

voidMemCopyDWORD(uint32_t *src, uint32_t *dst, uint32_t count)
{
    while(count--)*dst++ = *src++;
}
/*****
* MAIN
*****/
intmain (void) {

    uint32_t cnt_5s = 0, cnt_1s=0, cnt_5ms=0, cnt_25ms=0, cnt_100ms=0;
//    uint8_t packet_cnt, index;

    /* Initialize the application */
    Appl_Init ();

    /* Disable the Motor */
    Motor.Enable = 0;

    /* Loop forever */
    while (1)
    {
        if ( SysTick_cntr>= cnt_5ms )           {cnt_5ms =
SysTick_cntr+10;          Call_5ms();}
        if ( SysTick_cntr>= cnt_25ms )         {cnt_25ms =
SysTick_cntr+25;  Call_25ms();}
        if ( SysTick_cntr>= cnt_100ms ) {cnt_100ms =SysTick_cntr
+100; Call_100ms();}
        if ( SysTick_cntr>= cnt_1s )           {cnt_1s =
SysTick_cntr+1000;Call_1s();}
        if ( SysTick_cntr>= cnt_5s )           {cnt_5s =
SysTick_cntr+5000;Call_5s();}

#if (USE_NXP_GUI == 1)
        /* Process Rx Packets */
        packet_cnt = GetRxPacketCnt();

        if(packet_cnt> 0)
        {
            index = GetNextRxPacketIndex();
            ProcessRxPacket(index);
        }
#endif
    }
}

```

### 3.2.6 DIAGRAMA DE FLUJO PROGRAMA CONTROLADOR DE MOVIMIENTO DEL MOTOR BLDC HANDLERS.C

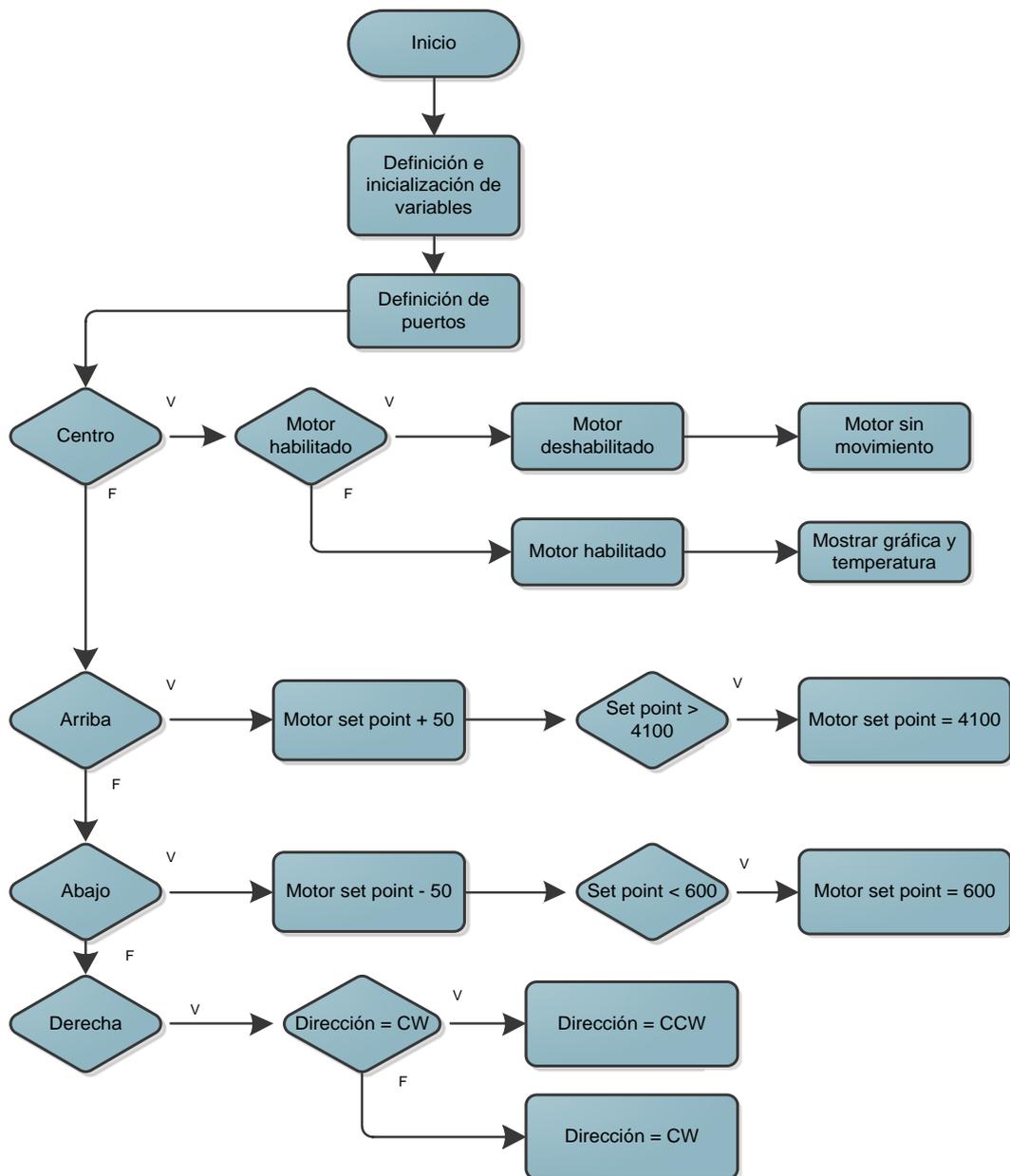


Figura 3.5 Diagrama de Flujo programa controlador de movimiento del motor BLDC handlers

### 3.2.7 DESCRIPCIÓN DEL ALGORITMO

- 1) Inicialización de variables.
- 2) Definición de variables a utilizar en el programa.
- 3) Definición de puertos a utilizar como entradas y salidas, en este caso hemos seleccionado el puerto dos y el puerto tres.
- 4) Si el usuario presiona el botón correspondiente al centro, entonces si el motor se encuentra habilitado el programa lo deshabilita inmediatamente y lo inmoviliza. En el caso de que el motor se encuentre deshabilitado, se lo habilita automáticamente y se lo pone en movimiento.
- 5) En el caso de que el usuario presione el botón arriba, el motor incrementa su velocidad, en el caso de alcanzar su máxima velocidad el motor se estabiliza.
- 6) Si el usuario presiona el botón abajo, el motor decrementa su velocidad hasta alcanzar el mínimo en cuyo caso se estabiliza.
- 7) Cuando el usuario presiona el botón de la derecha, el motor invierte el sentido de giro.
- 8) Todo se repite cada vez que se genere una interrupción la cual activa la función void PIOINT2\_IRQHandler(void) que contiene todo el procedimiento detallado.

### 3.2.8 PROGRAMA CONTROLADOR DE MOVIMIENTO DEL MOTOR BLDC HANDLERS.C

```

/*****
* Nombre: Handlers.c
* Autor: OscarSaigua / Emerson Chang
* Descripcion:
* Programa que nos permite el control de la velocidad del motor BLDC
* por medio de los puertos 2 y 3, además nos permite cambiar el sentido
* de giro del eje del motor, detener y activar su funcionamiento.
*****/

#include"application.h"

externvolatileuint32_tbldc_Tick;
externvolatileuint32_tbldc_SySTickold;
externvolatileuint32_tbldc_SySTicknew;
externvolatileMOTOR_TypeDef Motor;
externvolatileuint8_tshow_graph;
externvolatileuint8_tshow_temperature;
volatileuint32_tSysTick_cntr = 0;

/*****
** Function name:          PIOINT0_IRQHandler
**
** Descriptions:          Use one GPIO pin(port0 pin1) as interrupt source
**
** parameters:            None
** Returned value:        None
**
*****/

#ifndef GPIO_GENERIC_INTS

voidPIOINT2_IRQHandler(void)
{
    uint32_tregVal;
    uint32_t regVal2;
    regVal = LPC_GPIO2->MIS;
    regVal2 = LPC_GPIO3->MIS;
}

```

```

switch (regVal)
{
case (1<<HALL_A_PIN):
    /* Calculate the actual RPM */
    vBLDC_CalcRPM(&Motor);
    /* Read the HALL sensor */
    vBLDC_ReadHall();
    break;
case ((1<<HALL_B_PIN)):
case ((1<<HALL_C_PIN)):
    vBLDC_ReadHall();
    break;
case (1<<JOY_C_PIN):
    if (Motor.Enable)
    {
        /* Disable the motor */
        Motor.Enable = 0;
        /* Set the RPM to 0 */
        Motor.RPM = 0;
        Motor.RampingUp = FALSE;
    }
    else
    {
        /* Enable the motor */
        Motor.Enable = 1;
        Motor.CMT_CNT = 0;
        Motor.RampingUp = TRUE;
        show_graph = TRUE;
        show_temperature = TRUE;
    }
    break;

default:
    break;
}

```

```

switch (regVal2){

case (1<<JOY_U_PIN):
    Motor.sp+=50;
    if (Motor.sp>= 4100)
        Motor.sp = 4100;

```

```
        break;
    case (1<<JOY_R_PIN):

        if (Motor.Direction == CW)
            Motor.Direction = CCW;
        else
            Motor.Direction = CW;
        break;
    case (1<<JOY_D_PIN):
        Motor.sp-=50;
        if (Motor.sp<= 0)
            Motor.sp = 0;
        break;
    default:
        break;
}

/* Clear all interrupt sources */
LPC_GPIO2->IC = 0xFFFF;
LPC_GPIO3->IC = 0xFFFF;
return;
}
#endif//GPIO_GENERIC_INTS

void SysTick_Handler (void)
{
    SysTick_cntr++;
    bldc_Tick++;
}
```

### **3.3 CONTROL DE MOTOR BLDC MEDIANTE LAS TARJETAS LPC1769YLPC1114**

#### **3.3.1 DESCRIPCIÓN GENERAL**

En este caso la tarjeta LPC1114 es el dispositivo que se encarga de controlar el motor BLDC, ya sea para controlar su velocidad, sentido de giro, para iniciar o detener su movimiento. La tarjeta LPC1769 es la aquella que se encarga de procesar las señales que llegan desde el exterior ya sea de un joystick o de un conjunto de botoneras, dicha tarjeta envía las señales de voltaje para de esta manera lograr alguna respuesta inmediata en el motor BLDC. El motor control board trabaja conjuntamente con la tarjeta LPC1114 para mantener un control constante sobre el motor, controlando parámetros básicos como la velocidad máxima, velocidad mínima, temperatura.

### 3.3.2 DIAGRAMA DE BLOQUES

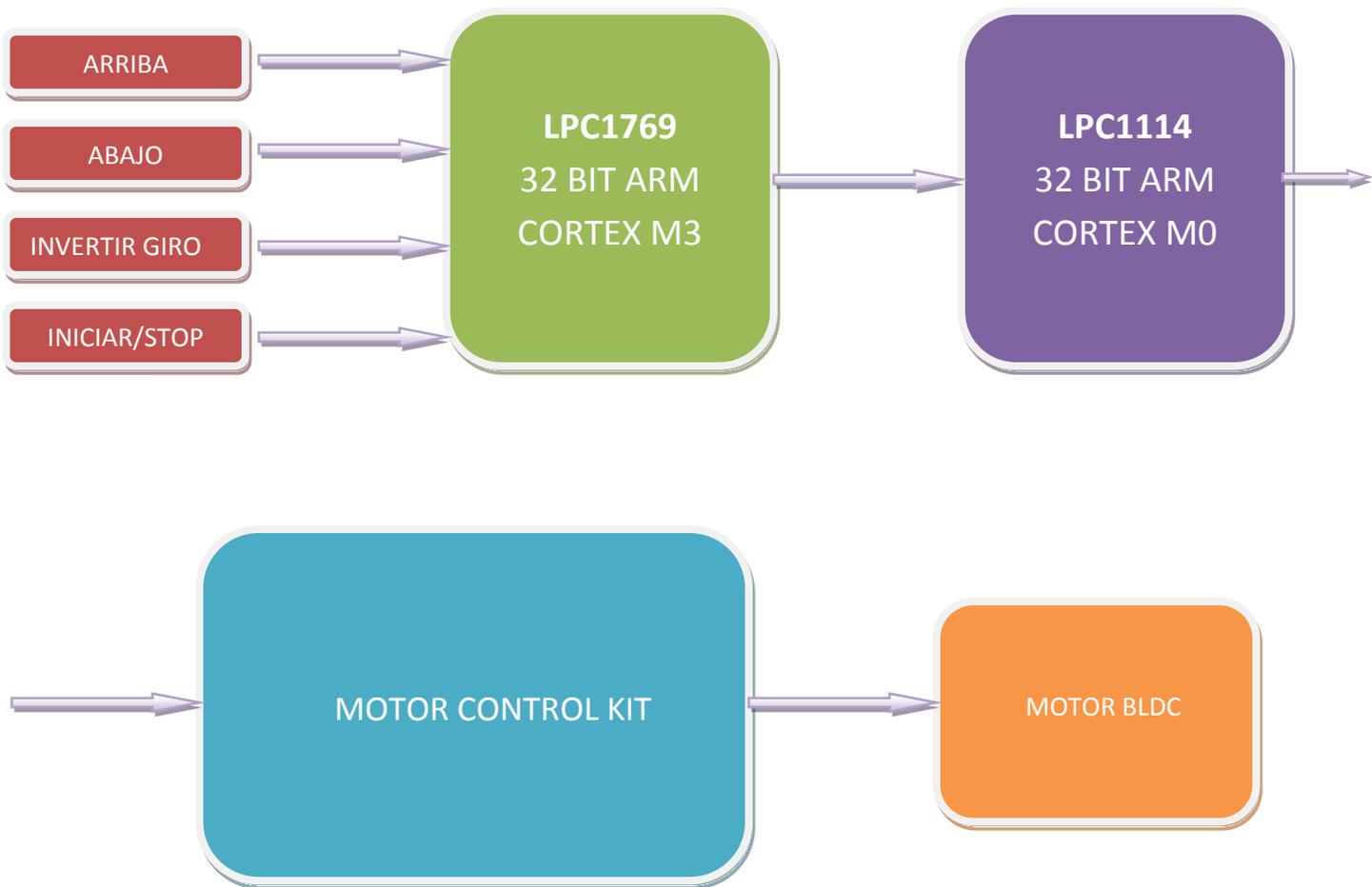


Figura 3.6 Diagrama de bloques control de motor BLDC mediante las tarjetas LPC1114 y LPC1769.

### 3.3.3 DIAGRAMA DE FLUJO DEL PROGRAMA CONTROLADOR DE BOTONERAS EN LA LPC1769 INTERFAZ.C

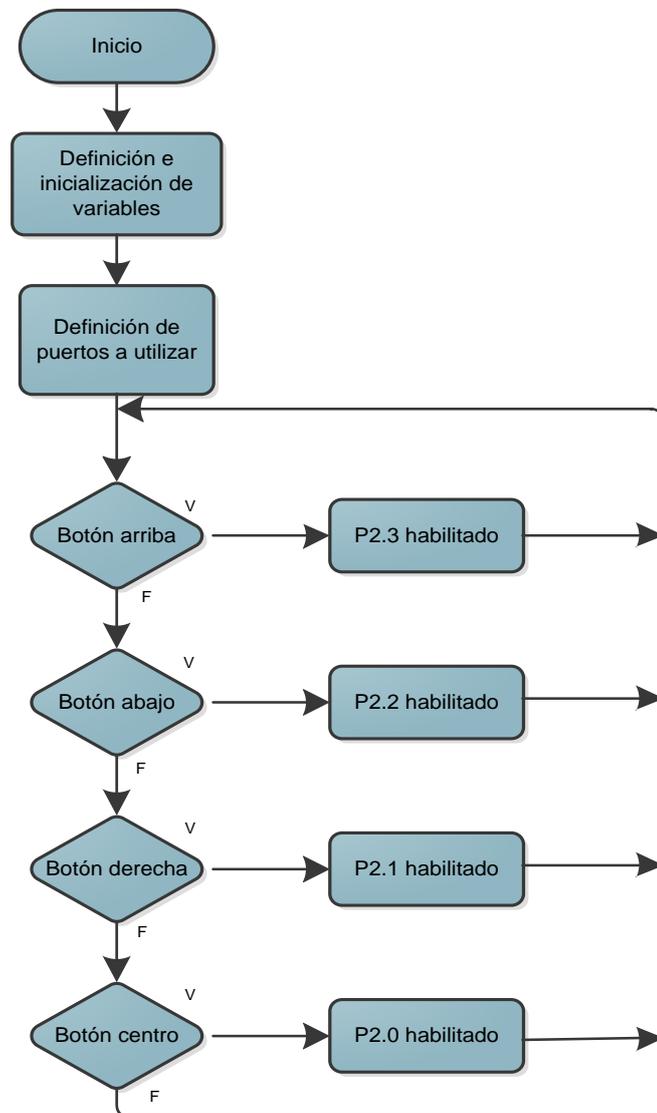


Figura 3.7 Diagrama de flujo del programa controlador de botoneras en la LPC1769 INTERFAZ.C.

### 3.3.4 DESCRIPCIÓN DEL ALGORITMO

- 1) Inicialización de variables.
- 2) Definición de puertos a utilizar como entradas y salidas, en este caso hemos seleccionado el puerto dos, los primeros cuatro pines serán definidos como salidas y los siguientes cuatro pines serán definidos como entradas, también es necesario asignarle un valor de '1' lógico a los primeros cuatro pines de salida, ya que estamos trabajando con lógica negativa.
- 3) Si el usuario presiona el botón asignado al pin P2.4, se activa con un '0' lógico el pin P2.0 lo que activa el motor o detiene su movimiento.
- 4) Si el usuario presiona el botón asignado al pin P2.5, se activa con un '0' lógico el pin P2.1 lo que ocasiona la inversión del sentido de giro.
- 5) Si el usuario presiona el botón asignado al pin P2.6, se activa con un '0' lógico el pin P2.2 lo que ocasiona que se incremente la velocidad de giro del motor.
- 6) Si el usuario presiona el botón asignado al pin P2.7, se activa con un '0' lógico el pin P2.3 lo que ocasiona que disminuya la velocidad de giro del motor.
- 7) Los pasos 3 al 6 se repiten indefinidamente ya que este procedimiento se encuentra en un lazo infinito.

### 3.3.5 PROGRAMA CONTROLADOR DE BOTONERAS EN LA LPC1769

#### INTERFAZ.C

```

/*****
 * Autores: Oscar Saigua / Emerson Chang
 * Nombre: Controlador de botoneras mediante tarjeta LPC1769
 * *****/
#include<cr_section_macros.h>
#include<NXP/crp.h>
__CRPconst unsigned int CRP_WORD = CRP_NO_CRP ;
#include"lpc17xx.h"
#include"type.h"
int main (void)
{
    uint32_t i,j;
    SystemClockUpdate();

    LPC_GPIO2->FIODIR = 0xFFFFFFFF;           /* P2.xx defined as
Outputs */
    LPC_GPIO2->FIOCLR = 0xFFFFFFFF;          /* turn off all the LEDs */
    LPC_GPIO2->FIOMASK= 0x00000000;
    while(1)
    {
        for(i = 0; i < 4; i++)
        {
            LPC_GPIO2->FIOSET = 1 <<i;
        }
        if(LPC_GPIO2->FIOPIN==0x000000EF){
            LPC_GPIO2->FIOCLR = 0x00000001;
        }
        if(LPC_GPIO2->FIOPIN==0x000000DF){
            LPC_GPIO2->FIOCLR = 0x00000002;
        }
        if(LPC_GPIO2->FIOPIN==0x000000BF){
            LPC_GPIO2->FIOCLR = 0x00000004;
        }
        if(LPC_GPIO2->FIOPIN==0x0000007F){
            LPC_GPIO2->FIOCLR = 0x00000008;
        }
        for(j = 1000000; j > 0; j--);
    }
}

```

## **CAPÍTULO 4**

### **PRUEBAS Y SIMULACIONES**

#### **4.1 RESUMEN**

En el siguiente capítulo detallaremos los diferentes ejercicios que hemos desarrollado, dando una descripción acerca del propósito de cada uno de ellos, especificando materiales y dispositivos que hemos utilizado y que son necesarios para dicha implementación.

A continuación mencionaremos los elementos cuyo uso es indispensable en todos los ejercicios desarrollados.

## 4.2 PROTOBOARD

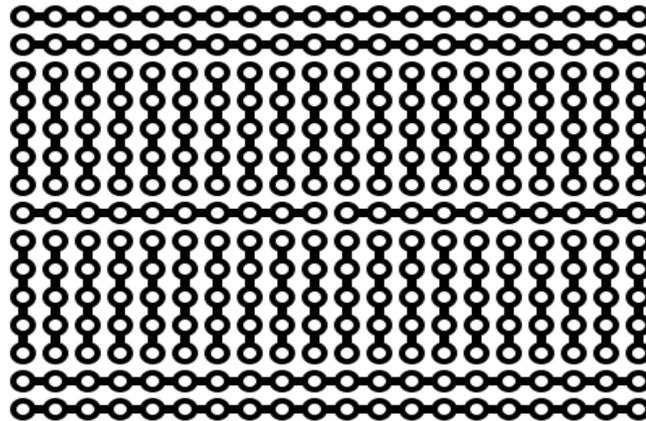
Es un dispositivo que nos permite ensamblar circuitos electrónicos de forma temporal, permite realizar circuitos sin necesidad usar el soldador, esto supone una gran ventaja a la hora de diseñar y probar nuestros proyectos. Esta herramienta está diseñada para poder ensayar prototipos sin necesidad de fijar los componentes. Los protoboards son fácilmente reconocibles por su forma de tableta perforada. En su interior, cuenta con unas láminas de metal que se encargan de sostener y conectar los terminales de los componentes y los hilos de conexión.



**Figura 4.1 Protoboard [18]**

Las filas superiores, al igual que las inferiores, suelen emplearse para insertar la de alimentación del circuito. Estas filas están unidas horizontalmente de extremo a extremo, así que todos los orificios que están en esas filas son una sola conexión. En la zona central vemos cuatro

regiones de perforaciones separadas por un canal. Las conexiones están dispuestas de forma vertical, por esta razón, siempre que necesitemos insertar un circuito integrado, lo haremos entre ambas regiones.



**Figura 4.2 Patrón de disposición de las láminas de material conductor en un Protoboard [19]**

### 4.3 MOTOR CONTROL KIT

El kit contiene el Motor Control Board y la tarjeta LPCXPRESSO LPC1114 Cortex M0 para el control del motor BLDC sin escobillas. El Motor Control Kit se trata de una plataforma universal para el control en baja tensión del motor, todo esto basado en microcontroladores de NXP. A continuación detallamos las principales características del Motor Control Board:

- Zócalo para LPCXpresso LPC1114, LPC11C24 and LPC1343.
- Zócalo para LPCXpresso LPC176x.

- 12-30V voltaje de entrada, 17A máxima corriente (max 300W salida).
- 15W fuente de alimentación (+11V, +5V, +3.3V).
- Joystick de 5 posiciones.
- Dimensiones de 200 x 150 mm.
- Pantalla OLED de 96x64 pixeles [6].



Figura 4.3 Lpcxpresso motor control kit [6]

## 4.4 ENCENDIDO ROTATORIO DE LEDS

### 4.4.1 DESCRIPCIÓN GENERAL

Nuestro dispositivo principal es la tarjeta LPC1769 quien enviará las diferentes señales de voltaje a través del puerto dos para de esta manera encender cada uno de los diodos led en el orden indicado. Además de esto

una botonera será conectada a la tarjeta para que esta detecte el momento en que el sentido de rotación de los diodos led debe ser cambiado.

#### 4.4.2 LISTADO DE MATERIALES

- ✓ Tarjeta LPC1769 NXP CORTEX M3.
- ✓ Ocho LEDS.
- ✓ Ocho resistencias 330  $\Omega$ .
- ✓ Botonera

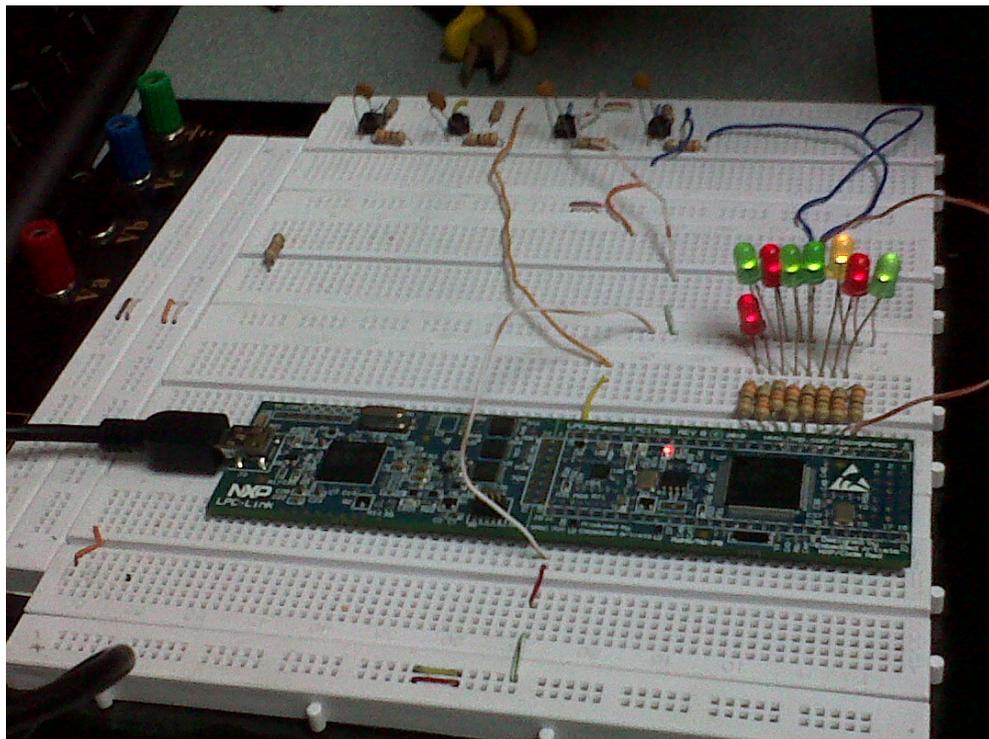


Figura 4.4 Desplazamiento rotatorio de Leds

### 4.4.3 DIAGRAMA DE CONEXIONES

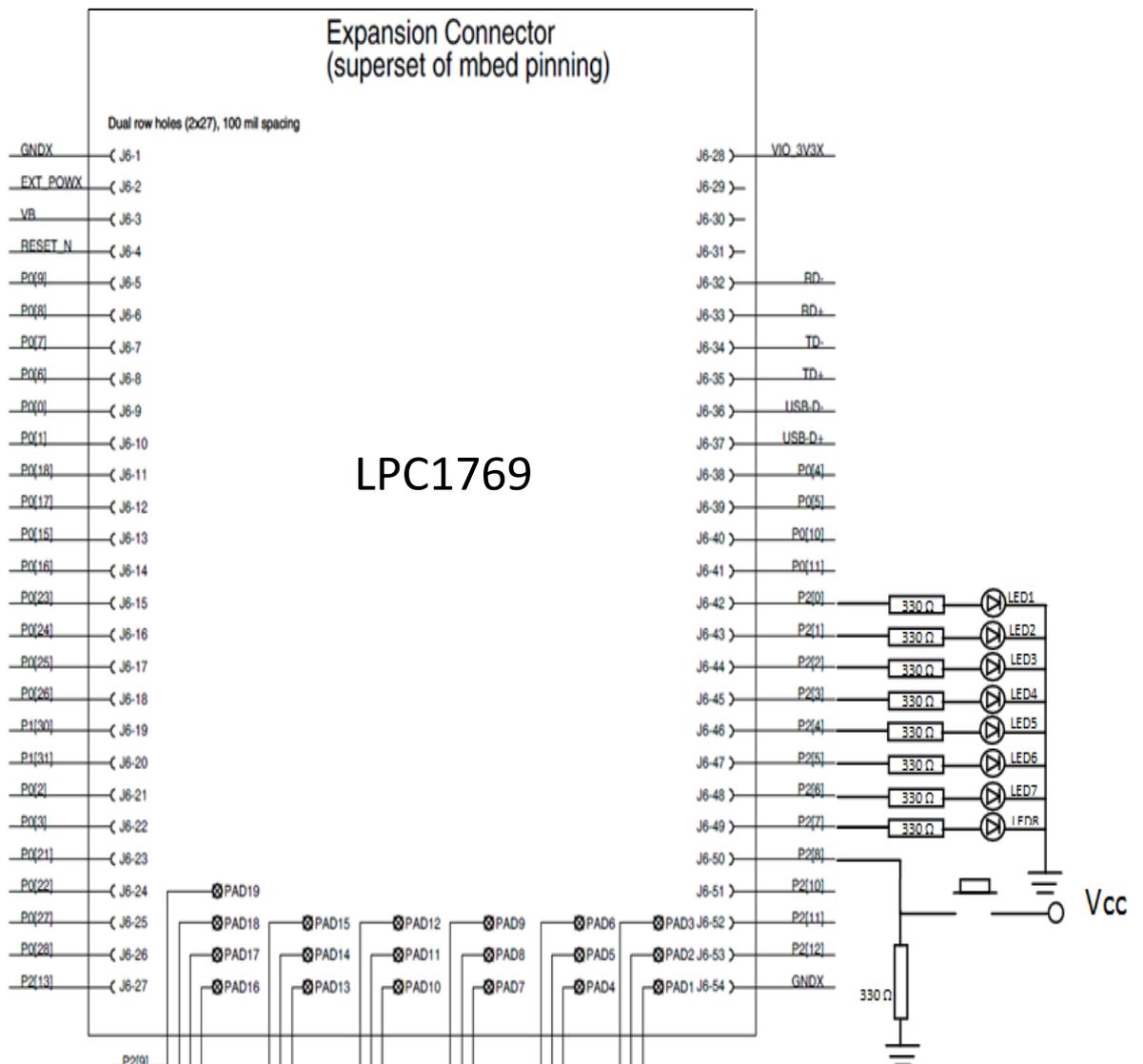


Figura 4.5 Conexión del control de motor BLDC mediante la tarjeta LPC1114

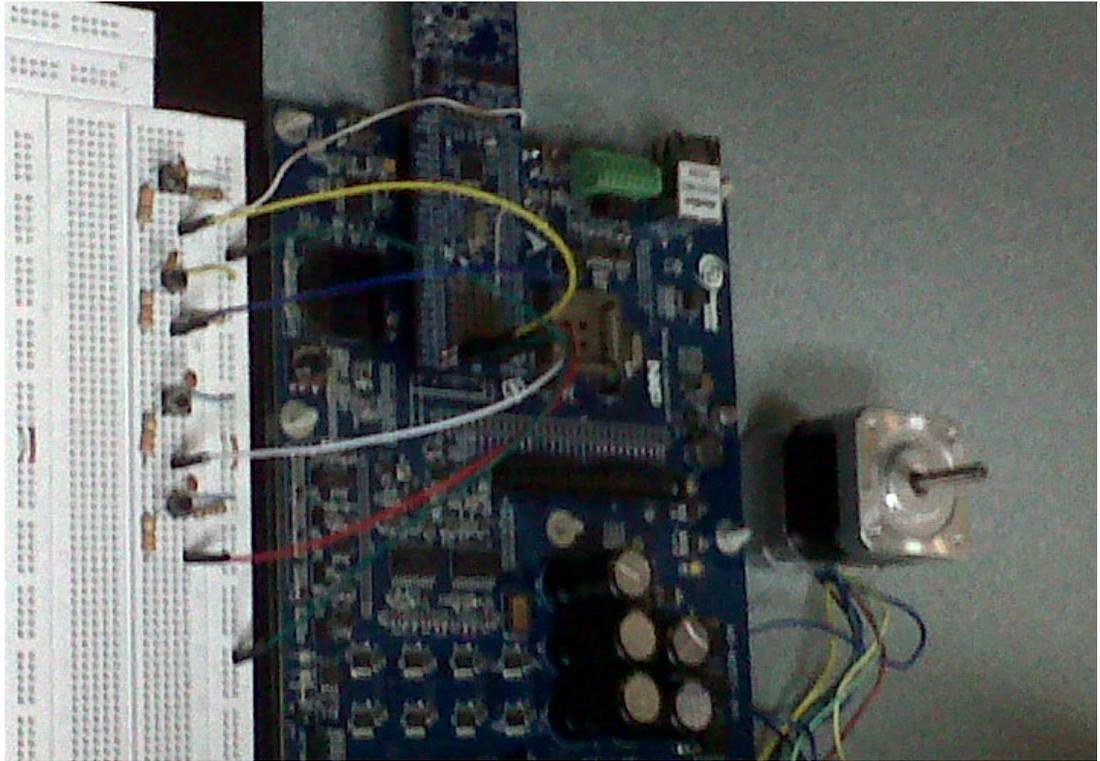
## **4.5 CONTROL DE MOTOR BLDC MEDIANTE LA TARJETA LPC1114**

### **4.5.1 DESCRIPCIÓN GENERAL**

La tarjeta LPC1114 es el dispositivo que se encarga de controlar el motor BLDC, ya sea para controlar su velocidad, sentido de giro, para iniciar o detener su movimiento, este control es logrado gracias a un conjunto de botoneras que nos permiten obtener una respuesta inmediata en el motor BLDC. El Motor Control Board trabaja conjuntamente con la tarjeta LPC1114 para mantener un control constante sobre el motor, controlando parámetros básicos como la velocidad máxima, velocidad mínima, temperatura, etc.

### **4.5.2 LISTADO DE MATERIALES**

- ✓ Tarjeta LPC1114 NXP CORTEX M0
- ✓ Motor Control Board NXP
- ✓ Cuatro Botoneras
- ✓ Cuatro Resistencias 270  $\Omega$  y 10 k $\Omega$
- ✓ Cuatro Resistencias.
- ✓ Cuatro capacitores de 100uf.



**Figura 4.6 Control de motor BLDC mediante la tarjeta LPC1114**

### 4.5.3 DIAGRAMA DE CONEXIONES

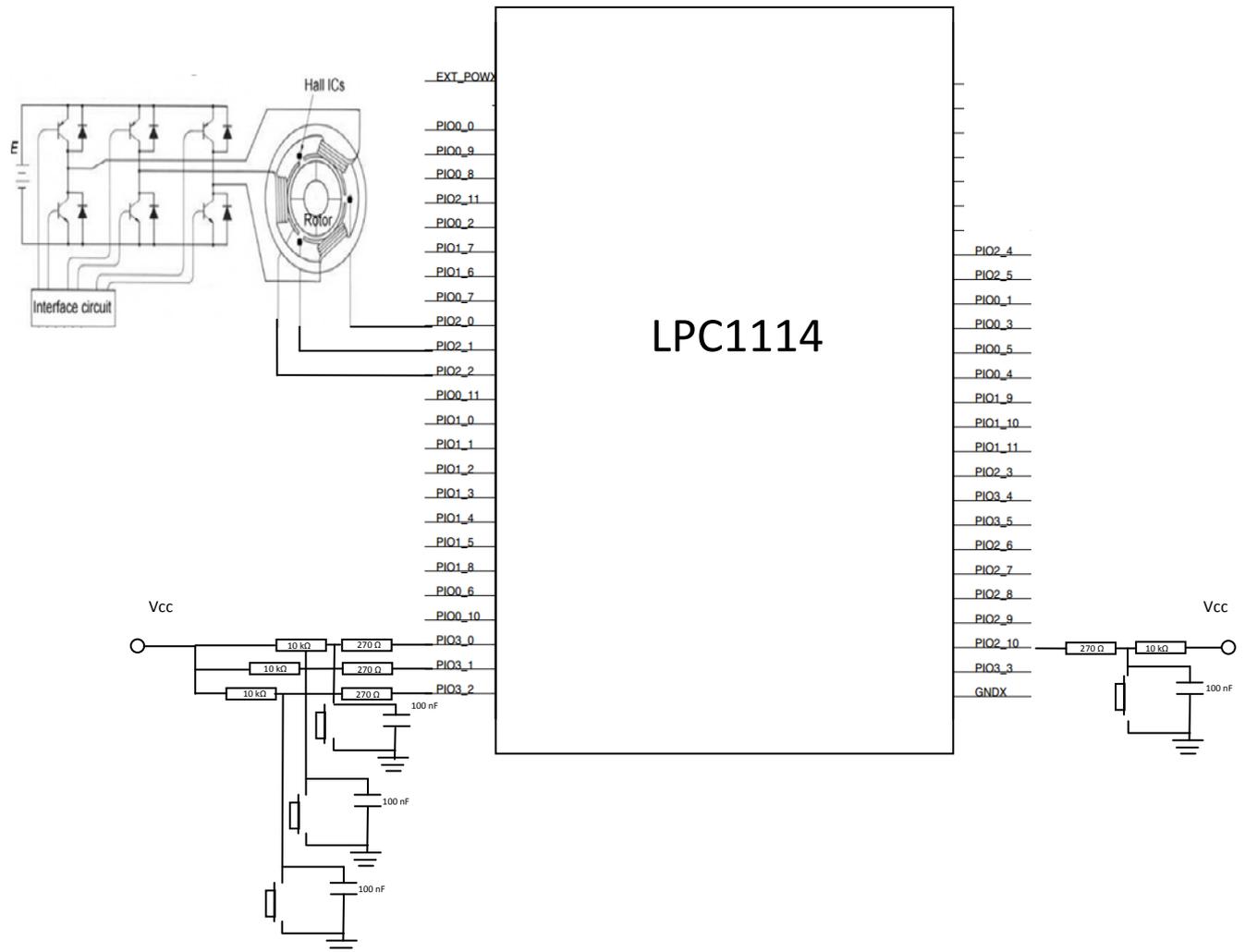


Figura 4.7 Conexión del control de motor BLDC mediante la tarjeta LPC1114

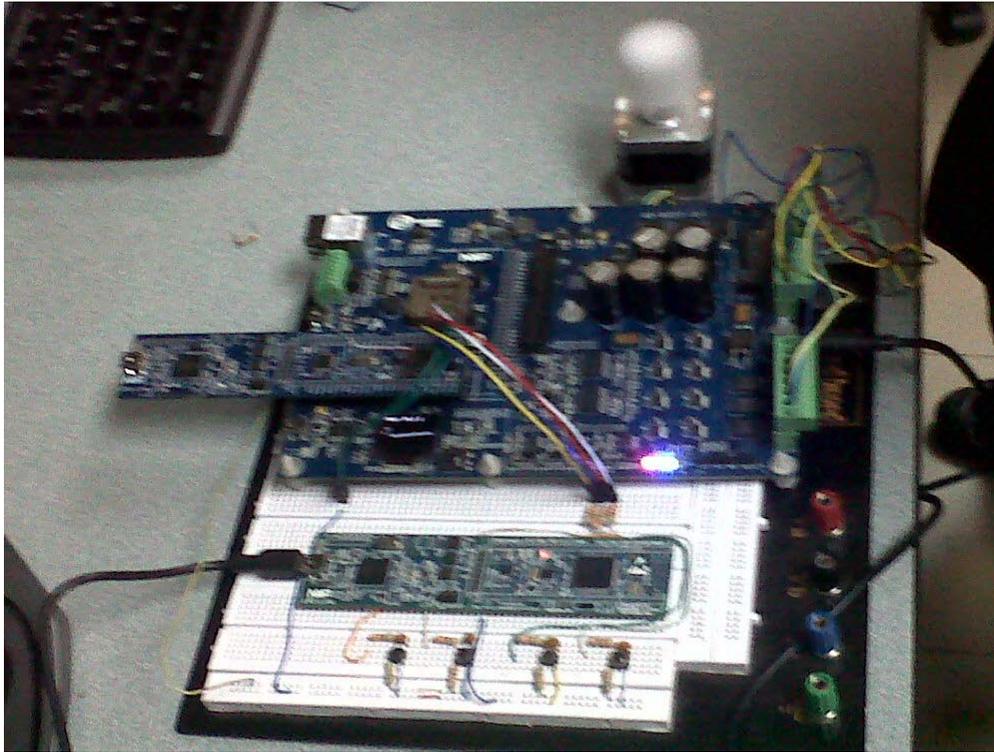
## **4.6 CONTROL DE MOTOR BLDC MEDIANTE LAS TARJETAS LPC1769 Y LPC1114**

### **4.6.1 DESCRIPCIÓN GENERAL**

En este caso la tarjeta LPC1114 es el dispositivo que se encarga de controlar el motor BLDC, ya sea para controlar su velocidad, sentido de giro, para iniciar o detener su movimiento. La tarjeta LPC1769 es la aquella que se encarga de procesar las señales que llegan desde el exterior ya sea de un joystick o de un conjunto de botoneras, dicha tarjeta envía las señales de voltaje para de esta manera lograr alguna respuesta inmediata en el motor BLDC. El motor control board trabaja conjuntamente con la tarjeta LPC1114 para mantener un control constante sobre el motor, controlando parámetros básicos como la velocidad máxima, velocidad mínima, temperatura.

### **4.6.2 LISTADO DE MATERIALES**

- ✓ Tarjeta LPC1114 NXP CORTEX M0
- ✓ Tarjeta LPC1769 NXP CORTEX M3
- ✓ Motor Control Board NXP
- ✓ Cuatro Botoneras
- ✓ Cuatro Resistencias 270  $\Omega$ .
- ✓ Cuatro Resistencias 10 k $\Omega$ .
- ✓ Cuatro capacitores de 100 nf



**Figura 4.8 Control de motor BLDC usando tarjetas LPC1114 y LPC1769**

### 4.6.3 DIAGRAMA DE CONEXIONES

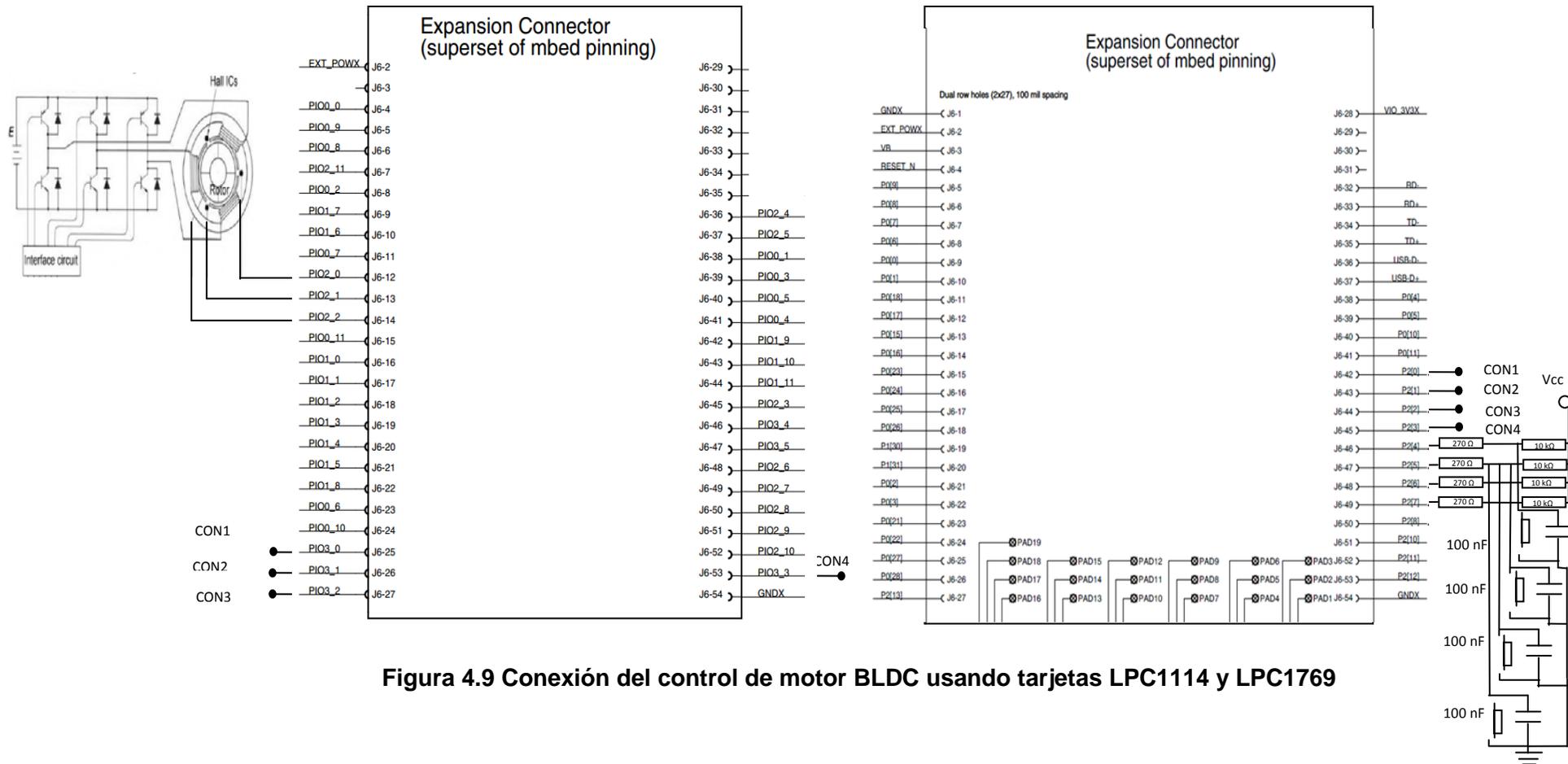


Figura 4.9 Conexión del control de motor BLDC usando tarjetas LPC1114 y LPC1769

## CONCLUSIONES

- 1) Luego de la realización de nuestro proyecto hemos podido completar la implementación de un control para motores BLDC usando sistemas embebidos provistos por LPCXpresso los cuales fueron las tarjetas LPC1114 CórteX M0 y LPC1769 CórteX M3 junto con el kit motor control y estos a su vez constituyeron el hardware. Y para el software usamos el programa LPCXpresso junto con las notas de aplicación de Microchip las cuales fueron de gran utilidad estas son: AN857a y AN957 obteniendo como resultado el control del motor BLDC cumpliendo con los objetivos propuestos como controlar la velocidad y el cambio de giro.
- 2) Se comprobó que el motor BLDC sin sensor está destinado principalmente al aeromodelismo y para aplicaciones en las cuales no se requieren un control a bajas velocidades ya que su control se torna crítico; así mismo para que el motor BLDC sin sensores opere debe existir una pequeña fuerza contra electromotriz a fin de ser medida y poder saber así la ubicación del eje del rotor para luego controlar los

transistores del puente H. En cambio los motores BLDC con sensores de efecto Hall han ganado mucho terreno en el ámbito industrial y actualmente se encuentran en gran cantidad de procesos industriales como a su vez los vemos en electrodomésticos. Los motores que incluyen sensores Hall generalmente manejan gran mayor de potencia que los de aerodelismo mientras que los BLDC sin sensores son motores pequeños y de poca potencia o sea bajo HP.

- 3) El funcionamiento del motor BLDC es controlado por las tarjetas LPC1114 Cortex M0 y LPC1769 Cortex M3. El kit del motor control fue de gran utilidad ya que dicho kit proveía la parte del circuito de fuerza necesario para hacer arrancar el motor, lo que nos ayudó a ahorrar tiempo y dedicarnos a la programación de las tarjetas para efectuar el control del motor.
- 4) Logramos controlar la velocidad del motor BLDC con una velocidad máxima de 4100rpm y una mínima de 600rpm. Disminuyendo o aumentando la velocidad desde una botonera que al presionarla varía la velocidad en 50rpm, lo que permite a la persona alcanzar una determinada velocidad, incluso poseemos una pantalla que nos permite observar la curva de velocidad del motor.

## **RECOMENDACIONES**

- 1) Al usar tarjetas embebidas como las utilizadas en este proyecto se recomienda sostenerlas por los lados procurando no tocar las partes electrónicas ya que una descarga electrostática podría quemar los componentes de la misma.
- 2) Al momento de cargar los programas en dichas tarjetas usando el cable micro USB verificar que este correctamente conectado el cable tanto a la tarjeta como la PC para una correcta programación de la tarjeta, al mismo tiempo al retirar el cable tomarlo por la base del mismo y no halarlo lo que podría dañar la tarjeta o el cable.
- 3) Al realizar el cambio de giro del motor es recomendable reducir las velocidad del motor, ya que aunque el proyecto es diseñado para hacer un cambio de giro instantáneo. Al realizar se produce un

movimiento brusco en el motor que podría afectar tanto al motor como a las conexiones realizadas al mismo.

- 4) Se debe conectar el GND de las tarjetas y el kit del motor control en un punto común para las tres referencias del GND.

## BIBLIOGRAFÍA

[1] Antonio Nadal Galiana Linares, Sistemas Embebidos, <http://server-die.alc.upv.es/asignaturas/PAEEES/2005-06/A07%20-%20Sistemas%20Embebidos.pdf>, fecha de consulta marzo 2012

[2] Instituto Balseiro, Maquinarias Eléctricas Modernas, <http://www.ib.cnea.gov.ar/nmayer/monografias/2008/Marcos%20Perez%20-%20M%C3%A1quinas%20El%C3%A9ctricas%20Modernas.pdf>, fecha de consulta marzo 2012

[3] NXP Semiconductors, Getting Started with NXP LPCXpresso, <http://ics.nxp.com/support/documents/microcontrollers/pdf/lpcxpresso.gettingstarted.pdf>, fecha de consulta marzo 2012

[4] NXP Semiconductors, LPC 1769/68/67/66/65/64/63 Product Data Sheet [http://www.nxp.com/documents/data\\_sheet/LPC1769\\_68\\_67\\_66\\_65\\_64\\_63.pdf](http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf), fecha de consulta marzo 2012

[5] NXP Semiconductors, LPC 1110/11/12/13/14/15 Product Data Sheet, [http://www.nxp.com/documents/data\\_sheet/LPC111X.pdf](http://www.nxp.com/documents/data_sheet/LPC111X.pdf), fecha de consulta marzo 2012

[6] NXP Semiconductors, LPCXpresso Motor Control Kit User's Guide, [http://www.embeddedartists.com/sites/default/files/support/xpr/motor\\_kit/LPCXpresso\\_Motor\\_Control\\_Users\\_Guide.pdf](http://www.embeddedartists.com/sites/default/files/support/xpr/motor_kit/LPCXpresso_Motor_Control_Users_Guide.pdf), fecha de consulta marzo 2012

[7] Johnson Electric, Motor EC (sin escobillas), <http://www.johnsonelectric.com/es/resources-for-engineers/automotive-applications/motion-technology/ec-motor-brushless.html>, fecha de consulta marzo 2012

[8] Ingenia-Cat Motion Control Department, Técnicas de Control para Motores Brushless, <http://www.ingeniamc.com/Es/-Control-techniques-for-brushless-motors.pdf>, fecha de consulta marzo 2012

[9] Howstuffworks, How does a Brushless Electric Motor Work?, <http://electronics.howstuffworks.com/brushless-motor.htm>, fecha de consulta marzo 2012.

[10] Gerardo Man Instructor de Aerodelismo, Motores Brushless Características, <http://www.brushless-rc-com.ar/Características.html>, fecha de consulta abril 2012

[11] Allegro Microsystems, Sensor de Efecto Hall Allegro 1321, <http://personales.ya.com/perropinto/robotidus/BRUJULA.htm>, fecha de consulta abril 2012.

[12] Centro de Radioastronomía y Astrofísica de la UNAM, Sensor de Efecto Hall, <http://www.crya.com.mx/docus/shall.pdf>, fecha de consulta abril 2012

[13] Let's Make Robots, Brushless DC Drivers, <http://letsmakerobots.com/node/2876>, fecha de consulta abril 2012

[14] MotorContinua, Tutorial del Motor DC, <http://es.cribd.com/doc/52141572/Tutorial-Interactivo-Del-Motor-Brushless>, fecha de consulta abril 2012

[15] Atmel Sensorless Control of BLDC Motors, <http://www.atmel.com/Images/doc8192.pdf>, fecha de consulta abril 2012

[16] Club R.C. Aznal Collar Aeromodelismo, Tabla Comparativa entre Motores Brushless y Brushed, <http://rc-aznalcollar.es/app/app/download/57779804422/Tabla+comparative+entre+motores+brushless+y+brushed.pdf>, fecha de consulta abril 2012

[17] Master Ingenieros S.A., Motores Brushless sin Escobillas Características Fundamentales, <http://www.masteringenieros.com/master/Ficheros/File/motor.pdf>, fecha de consulta abril 2012

[18] Electrónica y Servicio, Tabla Protoboard de Cuatro Bloques, <http://www.electronicayservicio.com/consumibles/consumibles/tpb107.htm>, fecha de consulta abril 2012.

[19] Wikipedia, Placa de Pruebas, [http://es.m.wikipedia.org/wiki/Placa\\_de\\_pruebas](http://es.m.wikipedia.org/wiki/Placa_de_pruebas), fecha de consulta abril 2012