

# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



## FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

“CONTROL MEDIANTE JOYSTICK DE TARJETA AVR BUTTERFLY  
USANDO MÓDULOS DE RADIO FRECUENCIA PARA COMUNICACIÓN  
CON TARJETA LPCXPRESSO CONTROLADORA DE MOTOR BLDC Y  
PRESENTACIÓN EN DISPLAY DE MENSAJES DE OPERACIÓN.”

### TESINA DE SEMINARIO

**Previa la obtención del Título de:**

INGENIERO EN ELECTRICIDAD  
ESPECIALIDAD EN ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL

**Presentado por:**

Ana Estefanía Jiménez Rodríguez

Xavier Roberto López Armijos

GUAYAQUIL – ECUADOR

AÑO 2012

## **AGRADECIMIENTO**

En primer lugar quiero agradecer a Dios por bendecirme para llegar hasta donde he llegado.

A mis padres y mis hermanos por ser siempre un pilar fundamental en toda mi etapa estudiantil.

A mis amigos por compartir momentos y circunstancias que dejaron enseñanzas en mí, a mi compañero de tesis por la paciencia y el apoyo constante.

Al Ing. Carlos Valdivieso por sus constantes aportes para la culminación de este proyecto.

**Ana Estefanía Jiménez Rodríguez.**

## **AGRADECIMIENTO**

A Dios por ser la guía perfecta que nos brinda sabiduría e inteligencia en todas las acciones y decisiones en mi vida.

A mis padres por ser ese apoyo incondicional en los momentos más importantes de mi vida.

A mis amigos que son la fuente de buenos ánimos y compartieron conmigo este largo trayecto.

Al Ing. Carlos Valdivieso por haber compartido sus conocimientos y tiempo en la realización de este proyecto.

**Xavier Roberto López Armijos.**

## **DEDICATORIA**

A mi madre, que siempre creyó en mí y ha sido el motor que me ha permitido seguir adelante día a día por su constante entrega y apoyo, que pueda observar culminada una meta.

**Ana Estefanía Jiménez Rodríguez.**

A mis padres, por ser ejemplo de mi vida, haber dado todo su apoyo en esta etapa estudiantil y hacer posible haberla culminado y a mis mejores amigos por estar conmigo y darme muchos consejos de vida.

**Xavier Roberto López Armijos.**

## **TRIBUNAL DE SUSTENTACIÓN**

---

Ing. Carlos Valdivieso A.

**PROFESOR DEL SEMINARIO DE GRADUACIÓN**

---

Ing. Patricia Chávez B.

**PROFESORA DELEGADA 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)

---

**Ana Estefanía Jiménez Rodríguez**

---

**Xavier Roberto López Armijos**

## RESUMEN

El motivo principal del presente trabajo es describir el funcionamiento de la comunicación UART por radiofrecuencia, para ello se hace uso de un transmisor que consta del kit de desarrollo AVR Butterfly, un módulo inalámbrico HM-TR el cual transmite los datos obtenidos por el puerto UART de la AVR Butterfly cuando se opera la funciones del joystick de la misma. Para la recepción de los datos se hace uso de un módulo similar al del transmisor el cual convierte la señal modulada en bits que son enviados a la tarjeta LPCXpresso1769 y estos son procesados por el CORTEX-M3 a las salidas que son las instrucciones que recibe la tarjeta controladora del motor LPCXpresso1114 y que finalmente se ve reflejado en las operaciones programadas para el control del motor.

El Kit AVR Butterfly, consta de un microcontrolador ATmega 169, en el cual se realiza la programación con el entorno AVR STUDIO 4 con el compilador GCC, y se graban las instrucciones del joystick en la memoria flash de la tarjeta; los módulos HM-TR para la transmisión recepción de los datos; la tarjeta de desarrollo LPCXpresso 1769 que incluye un microcontrolador CORTEX-M3; la tarjeta controladora y “Motor Control Kit” que ejecutan las instrucciones al motor BLDC.

Una vista general de cada capítulo de este trabajo es el siguiente, en el capítulo 1 muestra una reseña de las comunicaciones UART y la importancia de la radiofrecuencia en las comunicaciones actuales, sus aplicaciones y proyectos similares al expuesto en este trabajo, el capítulo 2 se hace una introducción a los motores BLDC y los fundamentos de radiofrecuencia, se hace énfasis en las herramientas de software y hardware necesarias para realizar la implementación del proyecto. El capítulo 3 corresponde a los ejercicios realizados para la familiarización con los kits AVR Butterfly y LPCXpresso y se enlista los componentes empleados para el desarrollo de cada ejercicio y la descripción de cada uno de ellos (el código fuente de cada ejercicio se enlista en los anexos). Por último, el capítulo 4 se enfoca en la respectiva simulación de cada ejercicio del capítulo 3, en software y su implementación física de ser posible.

## ÍNDICE GENERAL

RESUMEN.....	VII
ÍNDICE GENERAL .....	IX
ÍNDICE DE GRAFICOS .....	XIII
ÍNDICE DE TABLAS .....	XV
ÍNDICE DE ABREVIATURAS .....	XVI
INTRODUCCIÓN.....	XVIII
<b>CAPÍTULO 1: Descripción General del Proyecto</b>	
1.1 Motivación.....	2
1.2 Justificación .....	4
1.3 Descripción del proyecto.....	5
1.4 Aplicaciones.....	5
1.5 Proyectos similares.....	6
1.5.1 Circuito integrado para Control de Ventilador de Refrigeración.....	6
1.5.2 Controlador de motor OTP de medio puente .....	8
<b>CAPÍTULO 2: Fundamentos Teóricos</b>	
2.1 Introducción a los motores BLDC .....	11

2.1.1 Características fundamentales .....	13
2.1.2 Control a un motor BLDC.....	15
2.2 Propagación por Radiofrecuencia.....	18
2.2.1 Característica de la radiofrecuencia.....	18
2.2.2 Modulación FSK.....	20
2.2.3 Receptor FSK .....	21
2.2.4 Transmisor FSK.....	22
2.3 Herramientas de Hardware .....	23
2.3.1 Microprocesadores ATMEL .....	24
2.3.2 UART de la AVR Butterfly.....	24
2.3.3 Asignación de los pines para la comunicación con la PC.....	27
2.3.4 Microcontrolador ATmega169.....	28
2.3.5 UART en la LPCXpresso1769 .....	30
2.3.6 Microprocesador ARM CORTEX-M3 .....	30
2.3.7 Módulos inalámbricos HM-TR232.....	33
2.4 Herramientas de Software .....	35
2.4.1 Software AVR Studio 4 .....	35
2.4.2 PROTEUS .....	38
2.4.3 LPCXpresso IDE (Eclipse).....	40

**CAPÍTULO 3: Desarrollo e Implementación del Proyecto**

3.1 Diseño y desarrollo de la estación de trabajo .....	44
3.2 Transmisor: tarjeta AVR Butterfly.....	45
3.3 Receptor: LPCXpresso1769 y Motor control kit .....	46
3.4 Ejercicios a desarrollar e implementar .....	48
3.4.1 Ejercicio 1: Comunicación UART .....	48
3.4.2 Ejercicio 2: Comunicación UART LPCXpresso.....	51
3.4.3 Ejercicio 3: Control de un motor BLDC .....	54

**CAPÍTULO 4: Desarrollo de los Ejercicios Propuestos**

4.1 Especificación .....	62
4.1.1 Simulaciones .....	63

CONCLUSIONES

RECOMENDACIONES

ANEXOS

ANEXO I Código Fuente del Ejercicio 1: Comunicación UART

ANEXO II Código Fuente del Ejercicio 2: Comunicación UART LPCXpresso

ANEXO III Código Fuente del Ejercicio 3: Control de un motor BLDC

ANEXO IV Manual de usuario

ANEXO V Diagrama de bloques de la UART en la tarjeta AVR Butterfly

ANEXO VII Diagrama de bloques de la tarjeta LPCXpresso 1769

ANEXO VIII Diagrama eléctrico de las conexiones en la LPCXpresso

BIBLIOGRAFÍA

## ÍNDICE DE FIGURAS

Figura 1.1 Ventilador con motor BLDC .....	7
Figura 1.2 Controlador de un motor OTP de medio puente. ....	9
Figura 2.1 Vista lateral de un motor BLDC. ....	12
Figura 2.2 Esquema en bloques de un controlador con conmutación trapezoidal. ....	16
Figura 2.3 Esquema en bloques de un controlador con conmutación senoidal. ....	17
Figura 2.4 Movimientos de una partícula en el espacio .....	19
Figura 2.5 Receptor de FSK .....	22
Figura 2.6 Transmisor de FSK.....	23
Figura 2.7 Conectores del AVR Butterfly. ....	26
Figura 2.8 Conexiones para comunicación UART del AVR Butterfly.....	27
Figura 2.9 Módulo inalámbrico HM-TR. ....	34
Figura 2.10 Ventana Principal IDE.....	37
Figura 2.11 Ventana de Trabajo de Proteus .....	39
Figura 2.12 Entorno LPCXpresso IDE .....	40
Figura 2.13 Proceso de programar en C en LPCXpresso IDE.....	41
Figura 3.1 Estación de trabajo .....	44

Figura 3.2	Transmisor: Tarjeta AVR Butterfly y módulo HM-TR232 .....	45
Figura 3.3	Detalle de componentes del transmisor.....	46
Figura 3.4	Receptor: Tarjeta LPCXpresso, MKC y módulo HM-TR232 .....	47
Figura 3.5	Detalle de componentes del receptor .....	47
Figura 3.6	Diagrama de Flujo del ejercicio 1.....	49
Figura 3.7	Diagrama de Bloques del ejercicio 1 .....	51
Figura 3.8	Diagrama de Flujo del ejercicio 2.....	53
Figura 3.9	Diagrama de Bloques del ejercicio 2 .....	54
Figura 3.10	Diagrama de Flujo del ejercicio 3 transmisor .....	56
Figura 3.11	Diagrama de Flujo del ejercicio 3 receptor .....	57
Figura 3.12	Diagrama de Bloques del ejercicio 3 .....	59
Figura 4.1	Nombre de los integrantes del proyecto .....	64
Figura 4.2	Mensajes de cada uno de los botones del joystick.....	65

## ÍNDICE DE TABLAS

Tabla I Comparación entre un motor BLDC y un motor con escobillas. ....	14
Tabla II Asignación de pines, AVR Butterfly Vs. PC. ....	28
Tabla III Pines del módulo inalámbrico HM-TR.....	35

## ÍNDICE DE ABREVIATURAS

<b>AC</b>	Corriente alterna
<b>ADC</b>	Convertidor Analógico-Digital
<b>AHB</b>	Acceso a bus de alto rendimiento
<b>APB</b>	Bus de periféricos avanzados
<b>ASCII</b>	Código Estándar Estadounidense para el Intercambio de Información
<b>AVR</b>	Familia de microcontroladores de ATMEL
<b>BLDC</b>	Motor de corriente continua sin escobillas
<b>CAD</b>	Diseño asistido por computadora
<b>CPU</b>	Unidad central de procesamiento
<b>DC</b>	Corriente continua
<b>DMA</b>	Acceso directo a la memoria
<b>DSC</b>	Controlador de señal digital
<b>EMF</b>	Fuerza contra electromotriz
<b>FDMA</b>	Acceso múltiple por división de frecuencia
<b>FOC</b>	Control orientado por campo
<b>FSK</b>	Modulación por desplazamiento de frecuencia
<b>GND</b>	Tierra (Referencia)
<b>HCVA</b>	Ventilación, calefacción y aire acondicionado
<b>IDE</b>	Entorno de desarrollo integrado
<b>IEEE</b>	Instituto de ingenieros eléctricos y electrónicos

<b>ISM</b>	Banda industrial, científica y médica
<b>LCD</b>	Pantalla de cristal líquido
<b>MCK</b>	Motor Control Kit
<b>MHz</b>	Mega Hertz (un millón de Hertz)
<b>MOSFET</b>	Transistor de efecto de campo metal-óxido- semiconductor
<b>OTP</b>	Programables una sola vez
<b>PCB</b>	Placa de circuito impreso
<b>PLL</b>	Circuito de lazo cerrado
<b>PWM</b>	Modulación por ancho de pulso
<b>RC</b>	Filtro resistivo capacitivo
<b>RPM</b>	Revoluciones por minuto
<b>SCADA</b>	Supervisión, Control y adquisición de datos
<b>SPI</b>	Interfaz periférica serial
<b>TTL</b>	Lógica transistor-transistor
<b>UART</b>	Transmisor-Receptor Asíncrono Universal
<b>USB</b>	Bus universal en serie

## INTRODUCCIÓN

El proyecto presentado tiene como objetivo principal demostrar la importancia y funcionamiento de la comunicación UART, a través de radio frecuencia haciendo uso de módulos inalámbricos, usando el principio maestro-esclavo. La función de maestro la realiza el kit de desarrollo AVR Butterfly, las instrucciones ejecutadas por el joystick de la tarjeta son enviadas a la tarjeta de desarrollo LPCXpresso 1769 que es la encargada de procesar las instrucciones del maestro y mostrarlas por las salidas que posee esta tarjeta de desarrollo, a la salida estas instrucciones se convierten en las funciones programadas para el control del motor acoplado a la salida de la tarjeta controladora.

Para la implementación del proyecto se utilizaron herramientas de software para la programación y configuración de parámetros de las tarjetas y módulos inalámbricos, entre los que están AVR STUDIO 4, software utilizado para la programación en C de la AVR Butterfly, LPCXpresso 4.026, software de construcción y depuración del código en C para la tarjeta de desarrollo LPCXpresso y PROTEUS, software empleado para la simulación de algunos circuitos empleados en este proyecto.

# **CAPÍTULO 1**

## **ANTECEDENTES**

La importancia de las comunicaciones seriales, su aplicación en el campo de la industria, ha expandido de una manera acelerada el desarrollo e implementación de dispositivos capaces de transmitir y recibir datos cada más rápido y sin distorsión. Uno de los medios actualmente mas utilizados es la radiofrecuencia, debido a su versatilidad y facilidad para la instalación de los equipos de transmisión.

La descripción teórica de la comunicación serial no es complicada. La información se envía y recibe un bit a la vez a través del puerto serial y cuando esto es más lento que la comunicación en paralelo. Es un método de comunicación simple y puede alcanzar mayores distancias ya que permite la transmisión de un byte completo por vez. La especificación IEEE 488 determina que la longitud del cable para la comunicación serial puede alcanzar la distancia máxima de 1200 metros, mientras para la comunicación

paralelo la longitud no puede exceder los 20 metros, con no más de 2 metros entre dos dispositivos.

El formato para transmitir datos en la comunicación serial es el código ASCII. La comunicación requiere de 3 líneas de transmisión: (1) Tierra (o referencia), (2) Transmitir, (3) Recibir. Los datos pueden ser enviados por una línea y recibidos por otra, esto se debe a que la transmisión es asincrónica. Entre las características más importantes de la comunicación serial está la velocidad de transmisión, los bits de datos, los bits de parada, y la paridad. Para que dos puertos se puedan comunicar, es indispensable que estas características sean iguales para evitar conflictos en el envío y recepción de datos. Existen otras líneas disponibles para realizar "handshaking", o intercambio de pulsos de sincronización, pero no son requeridas.

## **1.1 Motivación**

La meta en este proyecto de graduación es aportar una base teórica sobre el funcionamiento de la comunicación UART mediante el uso de la radiofrecuencia y los métodos de control ya existentes. También contribuir con el desarrollo del área de Microcontroladores en la universidad y en el país. Finalmente aportar con un sistema de control

mediante joystick de tarjeta AVR BUTTERFLY usando módulos de radio frecuencia.

La limitación que producen las conexiones físicas (cables) para la implementación de sistemas de control de motores en sus diversas aplicaciones en la parte industrial, se encuentran con contratiempos, por ejemplo su movilidad, operación debido a su limitante física, cuando se realiza una instalación de este tipo de sistemas se debe tomar en cuenta también el riesgo de operación en el ambiente donde se lo va a utilizar.

El control remoto de vehículos que efectúan actividades de alto riesgo para un ser humano, en la industria el control inalámbrico de los periféricos conectados a un sistema SCADA, en robots móviles para que realicen actividades repetitivas sin la complejidad del cableado extenso; actividades en las que un cableado excesivo ocasiona problemas de instalación y operatividad.

Una solución viable a este problema es la utilización de un sistema de comunicación inalámbrico. El cual brindará flexibilidad en cuanto a la operación del sistema de control antes mencionado, además de proporcionar una ventaja de su implementación, costos, alcance y

movilidad en comparación con un sistema basado en un cableado sin perder ninguna característica en la operatividad de nuestra aplicación.

## **1.2 Justificación**

El principal problema del proyecto presentado en este trabajo radica en la comunicación UART entre las tarjetas de desarrollo AVR Butterfly y LPCXpresso empleando la radiofrecuencia para la transmisión y recepción de datos. Se emplearon módulos inalámbricos que garantizarán que no se produzcan algún tipo de distorsión en los datos enviados, ya que dichos datos son las posiciones del joystick de la tarjeta AVR Butterfly y receptadas por la tarjeta LPCXpresso y las salidas de esta son las instrucciones procesadas por el “Motor Control Kit”, para el control del motor BLDC acoplado a esta tarjeta.

Para ello se analizaron los límites de los componentes empleados en la implementación del proyecto para una optimización de los recursos. Para resolver el problema expuesto, se propone construir un código computacional que sea capaz de realizar la comunicación UART entre las tarjetas de desarrollo AVR Butterfly y LPCXpresso. Para lograr la máxima transmisión y recepción de datos, minimizando la distorsión de datos, pero además que cumplan con la totalidad de requerimientos del proyecto.

### **1.3 Descripción del proyecto**

Para la realización del proyecto se utilizó las tarjetas de desarrollo AVR Butterfly, LPCXpresso 1769 y “Motor Control Kit”, las cuales servirán de y control para un motor BLDC con sensores.

La comunicación empleada entre ambas tarjetas de desarrollo es la UART, pero con una variación en la transmisión y recepción de datos; ya que se utiliza módulos inalámbricos de radiofrecuencia para dicha función.

La función de receptor la cumple la tarjeta AVR Butterfly, que por medio del joystick permite al usuario interactuar con las funciones de posición que son mostradas en la pantalla LCD que esta posee. Dichas funciones se transmiten por radiofrecuencia, y son recibidas por la tarjeta LPCXpresso, posteriormente son procesadas y enviadas a la tarjeta “Motor Control Kit” como instrucciones para el control al motor que esta tiene acoplado.

### **1.4 Aplicaciones**

Entre las aplicaciones prácticas para este proyecto tenemos: el control remoto de vehículos que efectúan actividades de alto riesgo para un ser humano, en la industria el control inalámbrico de los periféricos conectados a un sistema SCADA, en robots móviles para que realicen

actividades repetitivas sin la complejidad del cableado extenso; actividades en las que un cableado excesivo ocasiona problemas de instalación y operatividad.

## **1.5 Proyectos similares**

Se presentaran dos casos de estudio similares que emplean motores BLDC, el primero es un microcontrolador para el accionamiento de ventiladores de refrigeración sin escobillas de filamento simple y el segundo control para un motor OTP de medio puente. En ambos casos de estudio se detalla su funcionamiento, modo de operación y las aplicaciones para la cual están diseñados.

### **1.5.1 Control de Ventilador de Refrigeración**

El microcontrolador diseñado para accionamiento de ventiladores de refrigeración sin escobillas de filamento simple a velocidad controlada y ruido reducido. En la figura 1.1 se puede apreciar el ventilador de refrigeración. El voltaje de alimentación esta en el rango de 4,5V a 16 voltios.



Figura 1.1 Ventilador con motor BLDC. [8]

Este microcontrolador está especialmente dirigido a las aplicaciones de ventilador de refrigeración con alimentación de "12 voltios". Posee una etapa de potencia integrada, la corriente circulante en los ventiladores de refrigeración alcanza 550mA DC, los que son requeridos para muchas aplicaciones. El rango de temperatura de operación es de -40 a 150 grados Celsius, esto permite su uso en casi cualquier tipo de ambiente.

Emplea técnicas digitales y/o analógicas para dar una solución simple al control de la velocidad del ventilador. Se utilizan técnicas de modulación de ancho de pulso (PWM) de alta frecuencia para la refrigeración de CPU en la que el ciclo de trabajo aplicado es proporcional a la velocidad de refrigeración

deseada. El rango de frecuencias PWM empleada por este microcontrolador va de 100 Hz a 100 KHz. Esto permite que acoplarse a sistema que utilicen frecuencia inferiores a los 25 KHz. Se utilizan el sensor de temperatura como señal analógica e inclusive el control del sensor de temperatura.

Para una refrigeración energéticamente eficiente el objetivo principal de los diseños es controlar y optimizar la velocidad de rotación del ventilador según la carga térmica del sistema.

### **1.5.2 Controlador de motor OTP de medio puente**

El microcontrolador MLX80151 para el control del motor OTP de medio puente fue diseñado para la automatización de motores DC de elevada corriente como ventiladores de refrigeración de motores, calefactores HVAC y bombas de inyección de combustible. El convertidor analógico-digital de auto secuencia permiten un control del motor en tiempo real y funciones de diagnóstico al mismo tiempo que minimiza la carga de interrupción en los microcontroladores de 16bit.

Si interfaz PWM permite realizar y se puede configurar para un solo cable, además de para aplicaciones con dos cables. Para

evitar el uso de un resonador cerámico, se emplea un oscilador RC de última generación con una tolerancia  $\pm 1,5\%$  en una gama de temperatura completa. En la figura 1.2 se muestra el esquemático del integrado conectado al motor DC y el rectificador de medio puente.

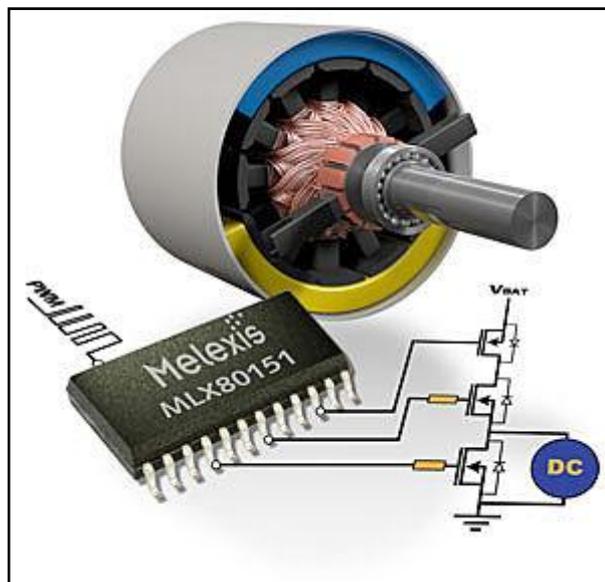


Figura 1.2 Controlador de un motor OTP de medio puente. [9]

## **CAPÍTULO 2**

### **FUNDAMENTOS TEÓRICOS**

En este capítulo se introducen los motores BLDC y su control y los recursos de hardware y software utilizados en el desarrollo del proyecto de graduación, se enuncia una breve descripción de los fabricantes de los microprocesadores en cada una de las tarjetas de desarrollo, el software empleado para la programación, construcción y depuración de los códigos para cada una de ellas, herramientas CAD de simulación de circuitos y el modo de operación de los módulos inalámbricos.

## 2.1 Introducción a los motores BLDC

Los motores de corriente continua sin escobillas (BLDC) son uno de los tipos de motores que más popularidad ha ganado en los últimos años. Actualmente, los motores BLDC se emplean en sectores industriales tales como: Automóvil, Aeroespacial, Médico, equipos de automatización e instrumentación.

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.

Estos motores carecen de colector y escobillas o carbones, 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 continua pulsante o bien una continúa con mucho componente de AC sin embargo se los clasifica como de DC porque al igual que los motores comunes tienen imanes permanentes. En la figura 2.1 se muestra una vista lateral de un motor BLDC.

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 mas rápido, simplemente hacemos girar el campo magnético secuencial a mayor velocidad, o aumentar la frecuencia de los pulsos.

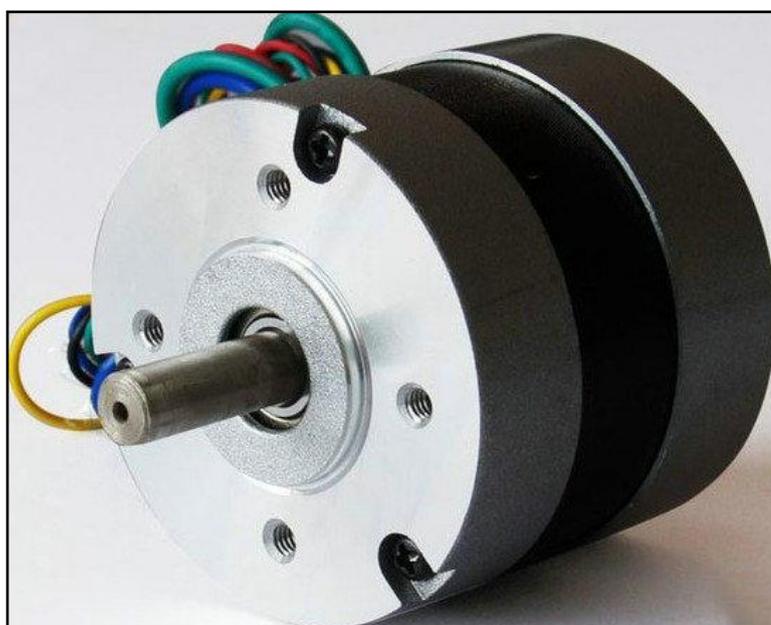


Figura 2.1 Vista lateral de un motor BLDC. [10]

Si tenemos un motor eléctrico común y le aumentamos la tensión en los terminales del mismo, veremos que aumenta la velocidad pero también aumenta la corriente de consumo. En un motor brushless la corriente y la velocidad son en cierto punto independientes; es el circuito regulador de velocidad el que se encarga de suministrar solo la corriente necesaria

para un determinado valor de RPM. Si alimentamos con mucha corriente, significa que estamos desperdiciando la capacidad de la batería, si en cambio, alimentamos con poca corriente, el motor eventualmente se detendrá.

El regulador controla la forma del pulso contra electromotriz (EMF) y así ajusta a la corriente de salida. La otra forma, la cual es la más común es que el motor inyecta un cierto valor de corriente y luego mide cual es la velocidad del motor al medir el EMF, de esta forma se auto ajusta.

Los reguladores de velocidad son los encargados de hacer andar el motor, en el caso del radiocontrol, miden el tiempo de 1ms a 2ms como un servo estándar y lo transforman en un máximo y mínimo de RPM's para el motor.

### **2.1.1 Características fundamentales**

Los motores BLDC tienen muchas ventajas frente a los motores DC con escobillas y frente a los motores de inducción. Algunas de estas ventajas que presentan son una mejor relación velocidad-par motor, la relación par motor-tamaño es mucho mayor, lo que implica que se puedan emplear en aplicaciones con una mayor respuesta dinámica, mayor eficiencia mejorando

la vida útil, reduciendo el ruido y permitiendo mayores rangos de velocidad. En la Tabla I se puede ver un resumen de las características y comparaciones básicas entre un motor BLDC y un motor con escobillas.

	<b>Motor BLDC</b>	<b>Motor con escobillas</b>
Conmutación	Conmutación electrónica basada en sensores de posición de efecto hall	Conmutación por escobillas
Mantenimiento	Mínimo	Periódico
Durabilidad	Mayor	Menor
Curva velocidad-par	Plana. Operación a todas las velocidades con carga definida	Moderada. A altas velocidades la fricción en las escobillas aumenta.
Potencia de salida-tamaño	Alta. Tiene una mejor disipación del calor.	Baja. El calor producido en la armadura es disipado en el interior.
Inercia del motor	Baja. Debido a los imanes permanentes del rotor.	Alta. Limita las características dinámicas.
Eficiencia	Alta. Sin caída de tensión por las escobillas	Moderada
Rango de velocidad	Alto. Sin limitaciones mecánicas	Bajo. Las escobillas son limitantes de la velocidad
Coste de construcción	Alto. Debido a imanes permanentes.	Bajo
Control	Complejo y caro	Simple y barato
Requisitos del control	El controlador es requerido para mantener el motor funcionando	No requiere control

Tabla I Comparación entre un motor BLDC y un motor con escobillas. [10]

Por otra parte, los motores BLDC poseen ciertas desventajas, tienen un mayor costo de adquisición, además requieren un control mucho más complejo, alto costo en su construcción, requiere una manufactura más elaborada.

### **2.1.2 Control a un motor BLDC**

Existe una gran variedad de técnicas para el arranque y control de velocidad de motores BLDC, entre las mas importantes están: la conmutación trapezoidal (ver figura 2.2); la de conmutación sinusoidal, (ver figura 2.3) y el método conocido como “Control Vectorial” (FOC – Field Oriented Control), este control efectúa sobre el trabajo del motor en funcionamiento. Este método es preciso, pero requiere de una enorme potencia de velocidad de cálculos y procesamientos de datos para proporcionar la posición exacta y el comportamiento del rotor del motor. Para una simplificación en los cálculos y procesamiento de datos se emplean los métodos de control trapezoidal y senoidal.

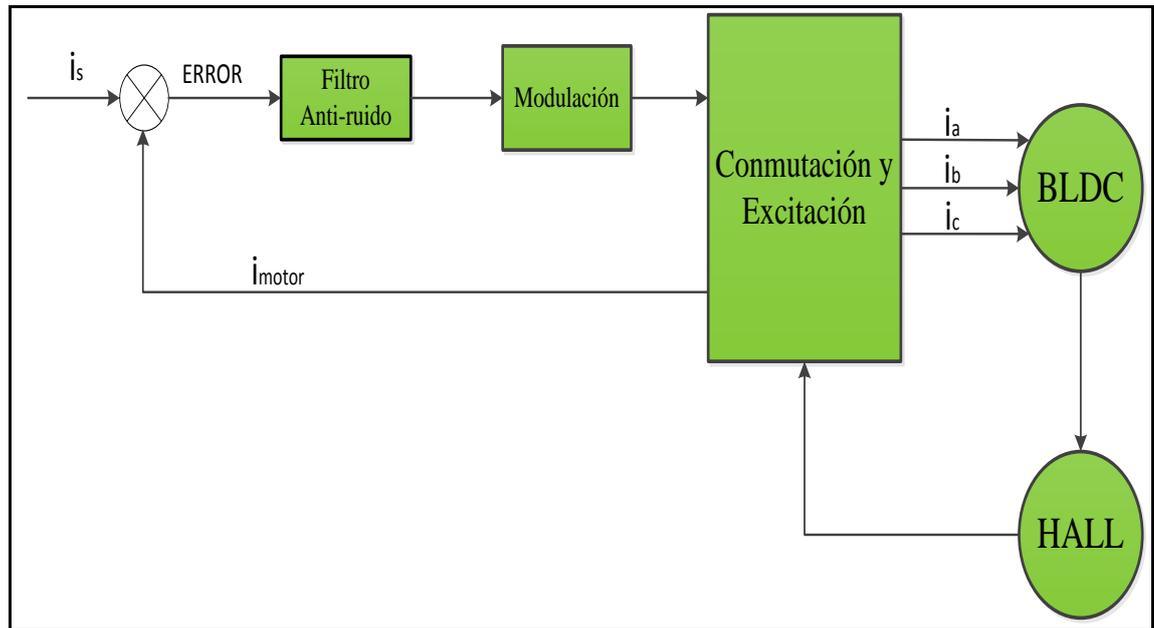


Figura 2.2 Esquema en bloques de un controlador con conmutación trapezoidal. [12]

El control por conmutación senoidal ha sufrido limitaciones en su desarrollo debido a los costos adicionales y los complejos algoritmos de control, sin embargo en aplicaciones que requieren un funcionamiento silencioso y de bajos niveles de vibración mecánica, su movimiento de rotación suave lo hace ideal para dichas aplicaciones.

Las diversas técnicas de control de posición del rotor de motores BLDC, pueden incluir o no, sensores de efecto Hall, o también llamados detectores de campos eléctricos en conductores atravesados por campos magnéticos, comúnmente usados en sistemas de control de conmutación trapezoidal.

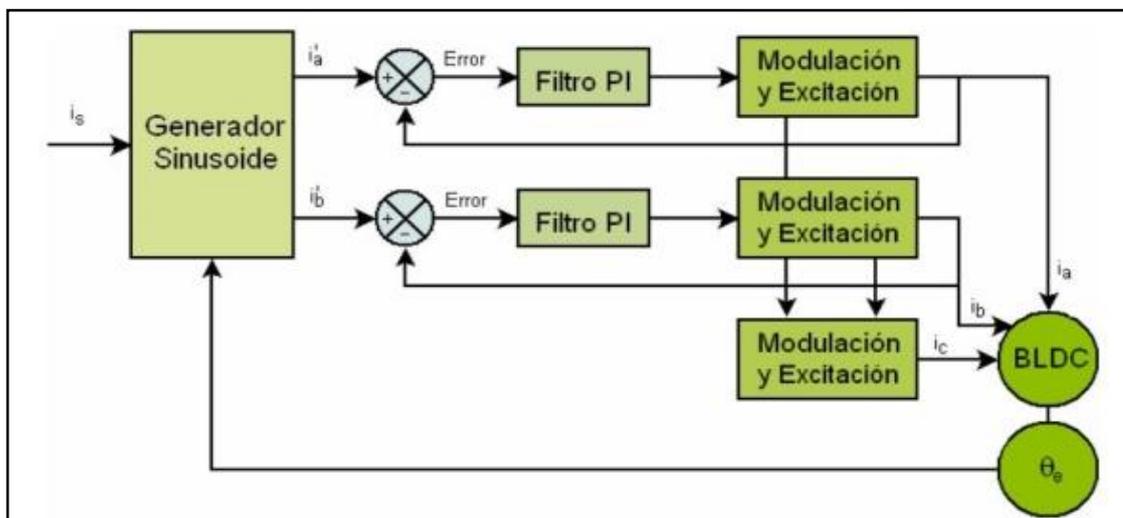


Figura 2.3 Esquema en bloques de un controlador con conmutación senoidal. [12]

Los cuatro bloques fundamentales de un sistema de control de un motor BLDC, la etapa de potencia para accionar el motor, generalmente se implementa con transistores MOSFET de potencia, los lazos de retroalimentación para detectar la posición del rotor y para realizar la detección de corriente por cada rama activada.

Los reguladores de tensión de alimentación para energizar los circuitos digitales, analógicos y de accionamiento de los transistores MOSFET.

En las aplicaciones actuales, muchos fabricantes de microcontroladores ofrecen periféricos para aplicaciones de

control de motores BLDC para optimizar el rendimiento del sistema. Los circuitos integrados capaces de controlar las tres fases de un motor BLDC reducen la complejidad del sistema y ayudan a cumplir con los requisitos de mínimo espacio dentro de un PCB.

## **2.2 Propagación por radiofrecuencia**

Las ondas electromagnéticas son aquellas que son capaces de viajar a través del vacío, a diferencia de las ondas mecánicas que necesitan un medio material para poder propagarse. Como su nombre indica, la radiación electromagnética consta de carga eléctrica y magnética. Existe un enorme rango de ondas electromagnéticas y se diferencian por su frecuencia.

### **2.2.1 Características de la radiofrecuencia**

Dos características importantes de las radiaciones electromagnéticas son: la energía fotónica y la velocidad de propagación.

La velocidad de transmisión de estas radiaciones es siempre la misma, 300.000 Km/s. (velocidad de la luz), sin importar su frecuencia o energía. La radiación electromagnética es un

transporte de energía mediante partículas (cuantos) a través del espacio. Estas partículas se desplazan siguiendo un movimiento ondulatorio que da lugar a una onda y son portadoras de energía eléctrica y energía magnética. Los fenómenos electromagnéticos se describen más fácilmente si se asocian no a ondas sino a partículas elementales o fotones (figura 2.4).

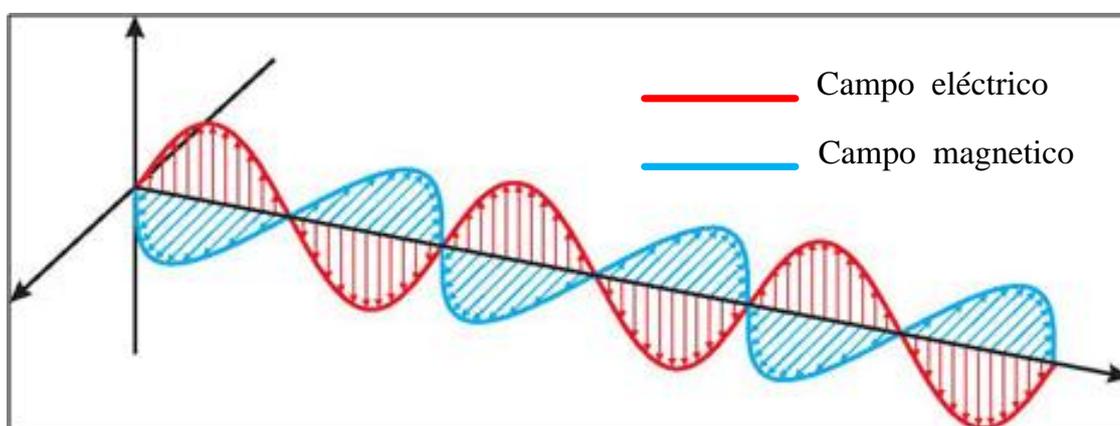


Figura 2.4 Movimientos de una partícula en el espacio. [15]

Cuanto mayor es la frecuencia de una onda electromagnética, mayor es la energía de la partícula asociada a ella. Los fotones asociados a los Rayos X y los Rayos Gamma tienen un alto contenido energético, mientras que los asociados a radiaciones extremadamente bajas tienen un bajísimo contenido energético.

### 2.2.2 Modulación FSK

La transmisión digital es la transmisión de pulsos digitales, entre dos o más puntos, de un sistema de comunicación. El radio digital es la transmisión de portadoras analógicas moduladas, en forma digital, entre dos o más puntos de un sistema de comunicación. Los sistemas de transmisión digital requieren de un elemento físico, entre el transmisor y el receptor, como un par de cables metálicos, un cable coaxial, o un cable de fibra óptica. En los sistemas de radio digital, el medio de transmisión es el espacio libre o la atmósfera de la Tierra.

La modulación por desplazamiento de frecuencia (FSK) es una forma de modulación angular de amplitud constante, similar a la modulación en frecuencia convencional, excepto que la señal modulante es un flujo de pulsos binarios que varía, entre dos niveles de voltaje discreto, en lugar de una forma de onda analógica que cambia de manera continua. La expresión general para una señal FSK binaria es:

$$v(t) = V_c * \text{Cos} \left[ \left( \omega_c + \frac{v_m(t) * D_m}{2} \right) t \right] \quad (2.1)$$

donde:

$V_c$  = amplitud pico de la portadora no modulada

$w_c$  = frecuencia de la portadora en radianes

$v_m(t)$  = señal modulante digital binaria

$D_m$  = cambio en frecuencia de salida en radianes

De la ecuación (2.1) puede verse que esta modulación, la amplitud de la portadora se mantiene constante con la modulación. Sin embargo, la frecuencia en radianes de la portadora de salida cambia. El cambio de frecuencia es proporcional a la amplitud y polaridad de la señal de entrada binaria.

### **2.2.3 Receptor FSK**

El circuito que más se utiliza para desmodular las señales de FSK binarias es el circuito de fase cerrada (PLL) véase figura 2.5, conforme cambia la entrada de PLL entre las frecuencias de marca y espacio, el voltaje de error de corriente continúa a la salida del comparador de fase sigue el desplazamiento de frecuencia.

Debido a que sólo hay dos frecuencias de entrada, también hay sólo dos voltajes de error de salida: uno que representa un 1 lógico y otro un 0 lógico. Por lo regular, la frecuencia natural del

PLL se hace igual a la frecuencia central del modulador de FSK. Los cambios en el voltaje de error cc, siguen a los cambios en la frecuencia de entrada analógica y son simétricos alrededor de 0V.

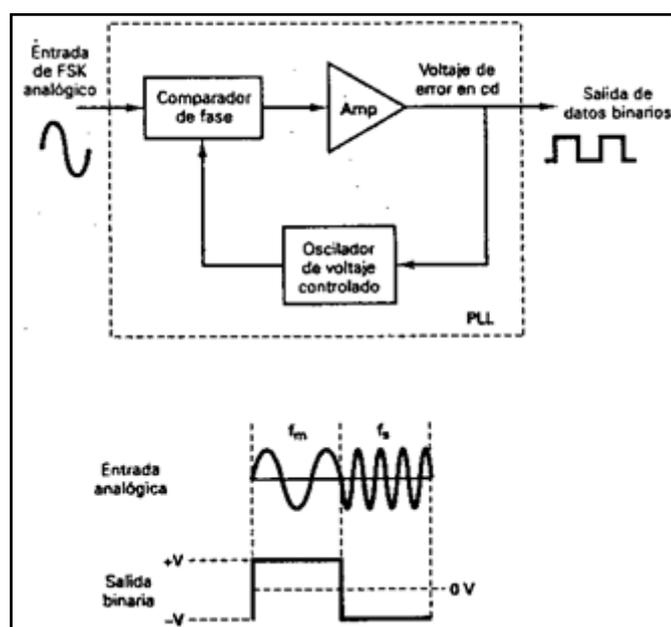


Figura 2.5 Receptor FSK. [7]

#### 2.2.4 Transmisor FSK

La salida de un modulador de FSK binario, es una función escalón en el dominio del tiempo. Conforme cambia la señal de entrada binaria de 0 lógico a 1 lógico, y viceversa, la salida se desplaza entre dos frecuencias, hay un cambio en la frecuencia de salida, cada vez que la condición lógica de la señal de entrada

binaria cambia. Un transmisor de FSK binario sencillo se muestra en la figura 2.6.

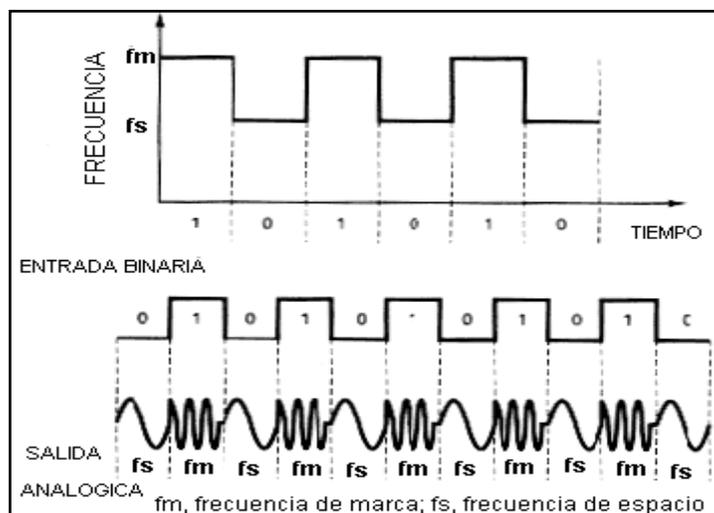


Figura 2.6 Transmisor FSK. [7]

## 2.3 Herramientas de Hardware

Se enunciará el hardware utilizado para la implementación de los ejercicios de prueba y el proyecto de graduación, las tarjetas AVR Butterfly, LPCXpresso y los módulos inalámbricos.

### 2.3.1 Microprocesadores ATMEL

Los AVR son una familia de microcontroladores del fabricante estadounidense Atmel. El diseño AVR de ATMEL difiere de los demás microcontroladores de 8 bits por tener mayor cantidad de registros (32) y un conjunto ortogonal de instrucciones. Tienen

diseño simple y son fáciles de programar. Se pueden dividir en los siguientes grupos:

**ATxmega:** procesadores muy potentes con memoria flash programable de 16 a 384 kB, encapsulados de 44, 64 y 100 pines (A4, A3, A1), capacidad de DMA, eventos, criptografía y amplio conjunto de periféricos con DACs.

**ATmega:** microcontroladores AVR grandes con memoria flash programable 4 a 256 kB, encapsulados de 28 a 100 pines, set de instrucciones extendido y amplio conjunto de periféricos.

**ATtiny:** pequeños microcontroladores AVR con memoria flash programable 0,5 a 8 kB, encapsulados de 6 a 20 pines y un limitado grupo de periféricos.

### 2.3.2 UART de la AVR Butterfly

El Transmisor/Receptor Universal Síncrono y Asíncrono (UART) es un dispositivo altamente flexible para comunicación serial. Las principales características son: Operación Full Duplex, Registros de transmisión y recepción independientes, Operación síncrona o asíncrona, Generador de tasa de baudios de alta resolución, Detección de error, Filtro de ruido, Modo de comunicación

multiproceso, Doble velocidad en modo de comunicación asíncrono.

Los registros de control son compartidos por todas las unidades. Los recuadros punteados en el diagrama de bloques muestran las tres partes principales del UART: Generador de reloj, transmisor y receptor. En el anexo V se muestra el diagrama de bloques de la UART en la tarjeta AVR Butterfly. La generación del reloj lógico consiste en la sincronización de entrada de reloj externo con la operación de esclavo síncrono; el pin XCLK (reloj de transferencia) solamente se usa para el modo de transferencia síncrona.

El transmisor consiste de un solo buffer de escritura, un registro de corrimiento serial, el generador de paridad y el control lógico para manejar los diferentes formatos de tramas seriales. El buffer de escritura permite una continua transferencia de datos sin ningún retardo entre tramas.

Algunos de los pines de I/O del microcontrolador ATmega169 están disponibles en los conectores del AVR Butterfly. Los conectores para comunicación son los UART, USI e ISP y los

conectores para la programación son: PORT B y PORT D y posee el conector ADC para la entrada al ADC del ATmega169, como se muestra en la figura 2.7.

El AVR Butterfly tiene incluido un convertidor de nivel para la interfaz RS-232, para reprogramar al AVR Butterfly utilizando la característica “auto-programación” del ATmega169.

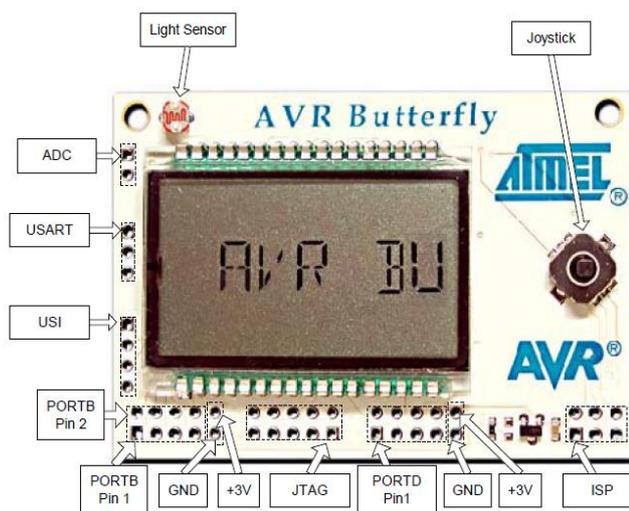


Figura 2.7 Conectores del AVR Butterfly. [1]

La AVR Butterfly tiene un pequeño joystick como entrada para el usuario. Este opera en cinco direcciones, incluyendo el centro. La línea común para todas las direcciones es GND. Esto significa que la resistencia de compensación interna deberá estar

habilitada en la ATmega 169 para detectar la entrada desde el joystick.

### 2.3.3 Asignación de los pines para la comunicación con la PC

La comunicación con la PC requiere de tres líneas: TXD, RXD y GND. TXD es la línea para transmitir datos desde la PC hacia el AVR Butterfly, RXD es la línea para recepción de datos enviados desde el AVR Butterfly hacia la PC y GND es la tierra común.

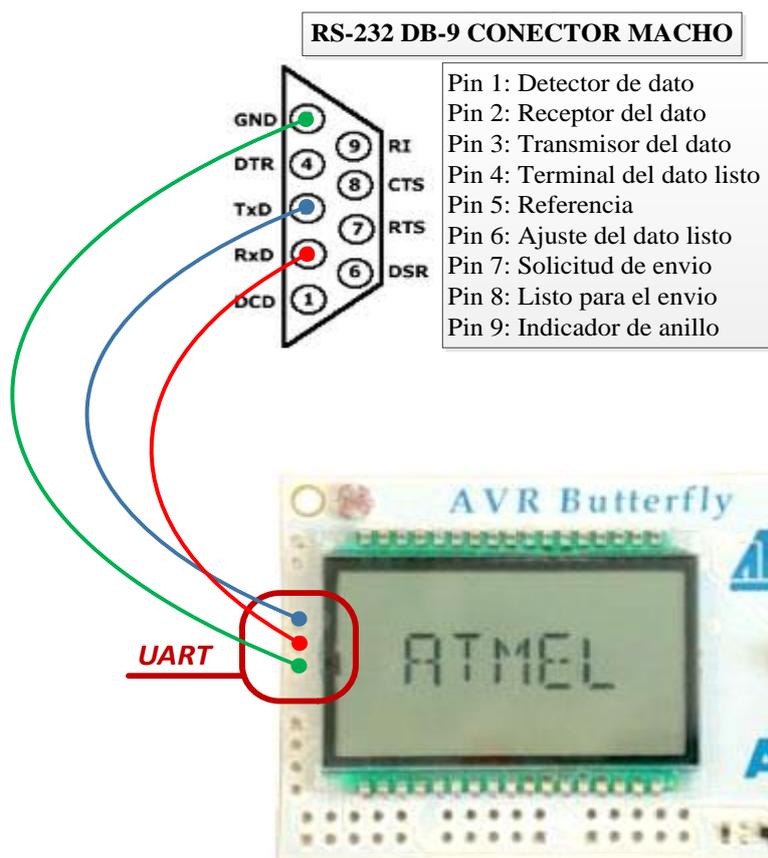


Figura 2.8 Conexiones para comunicación UART del AVR Butterfly. [1]

En la Figura 2.8 se observa cómo se debe realizar el cableado para la comunicación, a través de la interfaz serial RS-232, entre el AVR Butterfly y la PC. A la izquierda se aprecia un conector DB9 hembra soldado a los cables que se conectan a la interfaz UART del AVR Butterfly (derecha). En la tabla II se muestra la asignación de pines de la AVR Butterfly para la comunicación con la PC.

<b>Puertos en AVR Butterfly UART</b>	<b>Puerto en PC (RS232)</b>
Pin 1 (RXD)	Pin 3
Pin 2 (TXD)	Pin 2
Pin 3 (GND)	Pin 5

Tabla II Asignación de pines, AVR Butterfly Vs. PC.[1]

#### **2.3.4 Microcontrolador ATmega 169**

El ATmega169 es un microcontrolador de 8 bits de baja potencia basado en AVR y arquitectura RISC. Para ejecutar poderosas instrucciones en un solo ciclo de reloj, el ATmega 169 logra colocar MIPS por MHz permitiendo al diseñador del sistema optimizar el consumo de energía en función de la velocidad de procesamiento.

Las principales características del ATmega169 son las siguientes: alto desempeño, baja potencia, arquitectura RISC avanzada que incluyen, 120 instrucciones poderosas, la mayoría con ejecución de un solo ciclo de reloj, 32x8 registros de trabajo de propósito general y operación totalmente estática. Posee Memoria de Datos no volátiles con Memoria Flash Programable 2/4/8 Kbyte en el sistema, con duración: 10000 ciclos de escritura/borrado, 128/256/512 bytes de EEPROM programable en el sistema, con duración: 100000 ciclos de escritura/borrado. 128/256/512 bytes de SRAM interna.

Las características periféricas son: Contador/Temporizador de 8 bits con Prescaler y dos canales PWM. Contador/Temporizador de Alta Velocidad de 8 bits con Prescaler separado: dos Salidas PWM de Alta Frecuencia con Registros de Comparación de salida separados. Generador Programable de Tiempo Muerto. Interfaz Serie Universal con Detector de Condición de Comienzo. ADC de 10 bits: Cuatro Canales de Una Sola Salida. Dos Pares de Canales ADC Diferenciales con Ganancia Programable (1x, 20x). Temporizador Programable de Vigilancia con Oscilador separado dentro del integrado. Comparador Analógico dentro del integrado.

Las características especiales del microcontrolador son: Sistema de Depuración (Debug) WIRE dentro del integrado. Programable dentro del Sistema a través del Puerto SPI. Fuentes de Interrupción Externas e Internas. Modos de Descanso en Baja Potencia, de Reducción de Ruido de ADC, y de reducción de Potencia.

### **2.3.5 UART en la LPCXpresso 1769**

El LPCXpresso es un conjunto de herramientas completo para evaluación y desarrollo con microcontroladores de NXP. Esta compuesta por: LPCXpresso IDE y herramientas de desarrollo, IDE basado en Eclipse, Compiler y linker GNU, GDB debugger, LPCXpresso target board (stick), BaseBoard o hardware adicional (opcional).

### **2.3.6 Microprocesador ARM CORTEX-M3**

El ARM Cortex\_M3 es un microprocesador de 32 bits de propósito general, el cual ofrece un alto rendimiento y un bajo consumo de energía. El Cortex-M3 ofrece muchas nuevas características incluyendo un Thumb-2 instrucción de configuración, baja interrupción por latencia, hardware dividido, almacenamiento de instrucciones y carga de múltiples cargas

interrumpibles/continuo, autoguardado y restitución por interrupciones, un controlador integrado de interrupciones con un controlador de reactivación de instrucciones y múltiples núcleos de buses capaces con simultáneos accesos.

Las técnicas del microprocesador central son empleadas a todas las partes del procesamiento y sistemas de memoria que pueden operar de forma continua. Ocasionalmente, mientras una instrucción esta siendo ejecutada, su sucesora esta siendo decodificada, una tercera instrucción esta siendo buscada dentro de la memoria.

El procesador Cortex-M3 ofrece grandes beneficios a los desarrolladores, incluyendo un excelente rendimiento del procesamiento combinado con una manipulación de la interrupción, posee un mejorado sistema de depuración con vastos breakpoints y tramas, tiene un eficiente núcleo del procesador, sistema y memorias, una ventaja apreciable es su bajo consumo de energía con modos de hibernación integrados. Unidad de protección de la memoria (MPU).

El cortex-M3 incluye tres buses AHB-Lite, un sistema de bus y los buses, el I-code y el D-code, los cuales son más rápidos y son usados análogamente con las interfaces TCM: un bus está dedicado para la instrucción de búsqueda (I-code) y un bus para el acceso a datos (D-code). El uso de dos núcleos de buses permite operaciones simultáneas, si concurren operaciones de diferentes dispositivos.

La LPCX 1769 usa una matriz multi-capas AHB para conectar el Cortex-M3 y otros buses maestros para los periféricos en un método flexible que optimiza el rendimiento permitiendo que los periféricos de diferentes puertos esclavos de una matriz puedan ser accedidos simultáneamente por diferentes buses maestros.

En Anexo VI se muestra el diagrama de bloques de la LPCXpresso1769, donde se aprecia las matrices multi-capas AHB, el microcontrolador Cortex-M3 y demás puertos y memorias que componen la LPCXpresso1769.

Entre las principales características que posee la tarjeta de desarrollo LPCXpresso 1769 son las siguientes: Procesador Cortex-M3, funcionando en las frecuencias de hasta 100

megaciclos (LPCXpresso1768/67/66/65/64) o de hasta 120 megaciclos (LPCXpresso1769), una unidad de protección de memoria (MPU) que apoya ocho regiones es incluida. ARME Cortex-M3 incorporado jerarquizó el regulador de interrupción Vectored (NVIC).

### **2.3.7 Módulos inalámbricos HM-TR232**

El módulo de enlace inalámbrico de datos, series HM-TR es desarrollado por Hope microelectronics Co. Ltd., dedicado para aplicaciones que necesiten transmisión inalámbrica de datos. Esta característica permite una alta tasa de datos, una amplia distancia de transmisión.

Entre las características principales que posee los módulos inalámbricos HM-TR232 son: Tecnología FSK, modo half dúplex, robusto a la interferencia. Banda ISM, no requiere aplicar una licencia de frecuencia para su uso.

La tasa de datos puede ser seleccionado de un amplio rango. Posee un pin habilitador para el control del ciclo de trabajo para satisfacer los requerimientos de las diferentes aplicaciones. Alta sensibilidad, amplio rango de transmisión. Interfaz UART

estándar, nivel lógico TTL o RS232 seleccionable. Muy confiable, tamaño pequeño y de fácil montaje. En la tabla III se enlista los pines de módulo inalámbrico con su respectiva descripción de la función que cada uno de ellos realiza.

Las aplicaciones mas usas con este módulo son: Control remoto, sistema de medición remoto. Medición de enlace inalámbrico. Control de acceso. Selector de identidad. Almacenamiento datos. Monitoreo de recién nacidos. En la figura 2.9 se muestra el esquema de la asignación de cada uno de los pines del módulo inalámbrico HM-TR232.

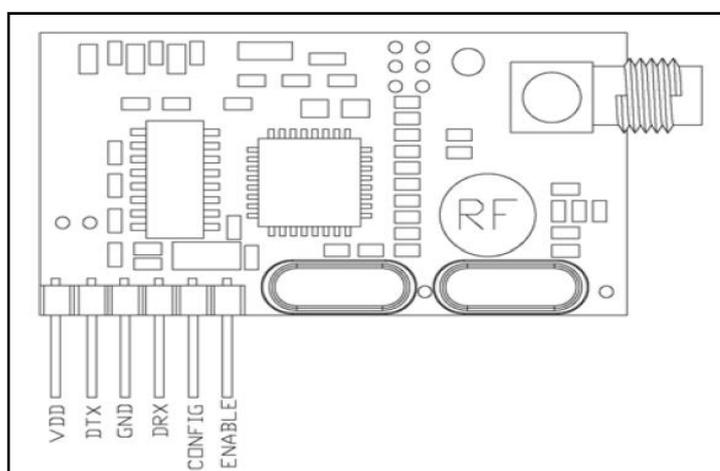


Figura 2.9 Modulo inalámbrico HM-TR. [5]

Pines	Función	Descripción
Vcc	Fuente de alimentación	+ 5V
DTX	Transmisión del dato	Modulo de transmisión del dato
DRX	Recepción del dato	Modulo de recepción del dato
CONFIG	Modo de configuración	Si el pin CONFIG esta en alto, el modulo entra en el modo de configuración para ajustar los parámetros de trabajo.
ENABLE	Función de trabajo	Si el pin CONFIG esta en bajo, el modulo entrará en modo normal para transmisión de datos.

Tabla III Pines del módulo inalámbrico HM-TR. [5]

## 2.4 Herramientas de Software

Se enunciara brevemente sobre los recursos de software utilizados en el presente trabajo, AVR Studio 4, PROTEUS y LPCXpresso Eclipse como los medios de programación de las tarjetas y simulación de los circuitos implementados.

### 2.4.1 Software AVR Studio 4

Adicionalmente, ATMEL también proporciona en línea el entorno de software AVR estudio 4 que permite editar, ensamblar y simular el código fuente. Una vez ensamblado y depurado el código fuente del programa, se transferirá el código máquina a la memoria flash del microcontrolador para esto se debe disponer

de otro entorno de desarrollo para programar en forma serial o paralelo la memoria flash.

Las características del AVR Studio 4 son: Un núcleo del lenguaje simple, con funcionalidades añadidas importantes, como funciones matemáticas y de manejo de archivos, proporcionadas por bibliotecas. Es un lenguaje muy flexible que permite programar con múltiples estilos. Uno de los más empleados es el estructurado. Usa un lenguaje de pre procesado, el preprocesador de C, para tareas múltiples archivos de código fuente.

Acceso a memoria de bajo nivel mediante el uso de punteros. Un conjunto reducido de palabras clave. Por defecto, el paso de parámetros a una función se realiza por valor. Plataforma de trabajo, Windows, Linux. Genera un código de maquina compatible con simuladores como Proteus.

AVR Studio 4 consiste de muchas ventanas y sub-módulos. Cada ventana apoya a las partes del trabajo que se intenta emprender. En la figura 2.10 se puede apreciar la ventana principal del IDE. AVR Studio 4 proporciona herramientas para la administración de

proyectos, edición de archivo fuente, simulación del chip e interfaz para emulación en circuito de la poderosa familia de microcontroladores AVR de 8 bit.

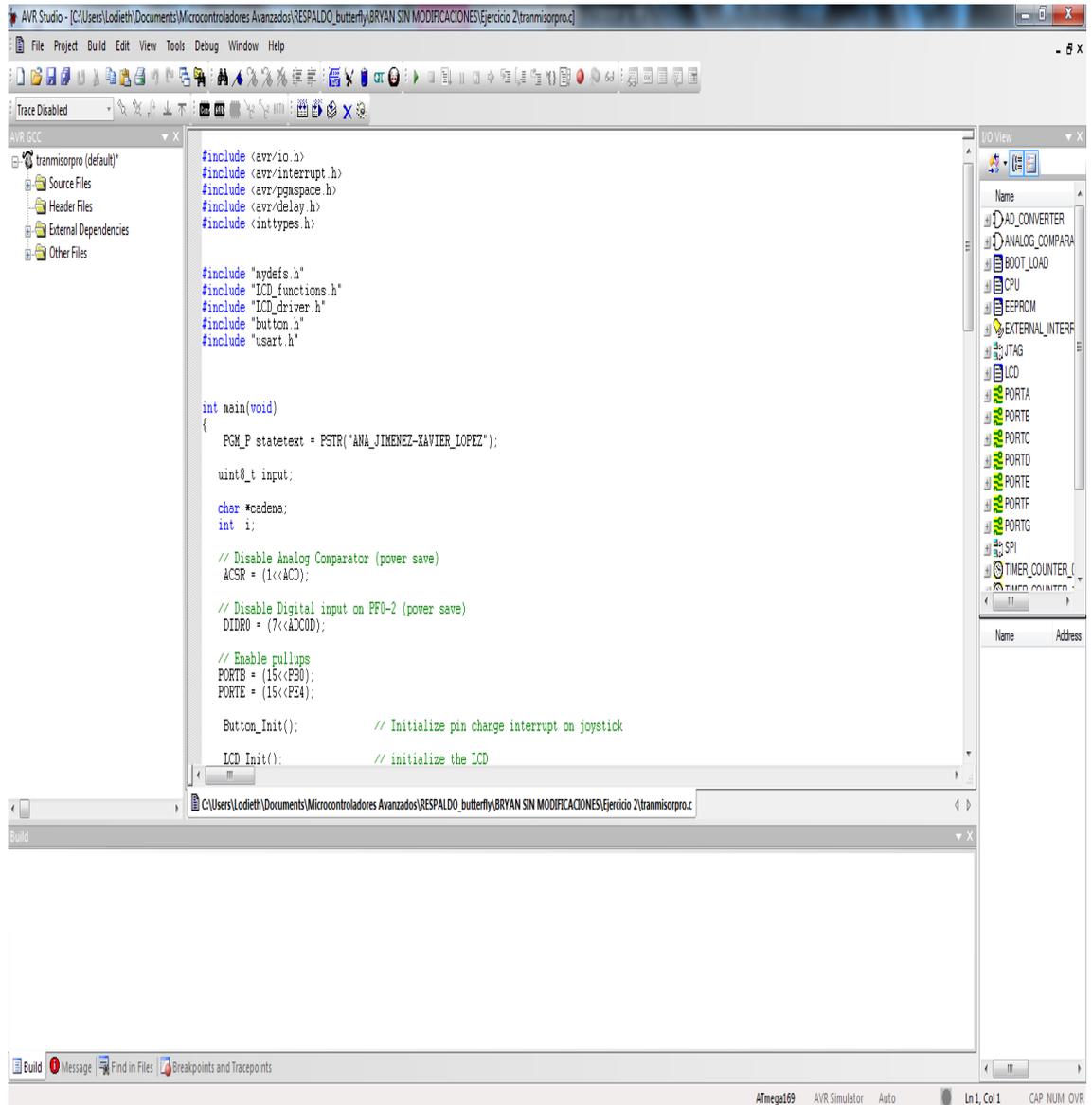


Figura 2.10 Ventana Principal IDE. [6]

## 2.4.2 PROTEUS

Proteus es una aplicación CAD, compuesta de tres módulos:

- ISIS (Intelligent schematic input system): Es el módulo de captura de esquemas.
- VSM (Virtual system modeling): Es el módulo de simulación incluyendo Prospice.
- ARES (Advanced routing modeling): Es el modulo para la realización de circuitos impresos (PCB).

El módulo ISIS es un programa que nos permite dibujar sobre un área de trabajo un circuito que posteriormente podremos simular. En la manipulación del software casi siempre existirán varias opciones para un mismo fin, normalmente podremos optar por seguir un menú, acceder a un icono o trabajar en el teclado.

El menú permite acceder a la mayor parte de opciones del programa, sin embargo algunas solo están disponibles en los iconos de las barras de herramientas. Las barras de herramientas son varias y se pueden colocar en cualquier parte de la pantalla. El área de trabajo se muestra en la figura 2.11.

PROTEUS pose la aplicación ISIS permite generar circuitos reales, y comprobar su funcionamiento en un PCB (printed circuit board) tarjeta de circuito impreso. Entorno de diseño gráfico de esquemas electrónicos fácil de utilizar e implementar al momento de realizar un circuito electrónico. Proteus cuenta con una gran cantidad de funciones para trabajar con circuitos electrónicos.

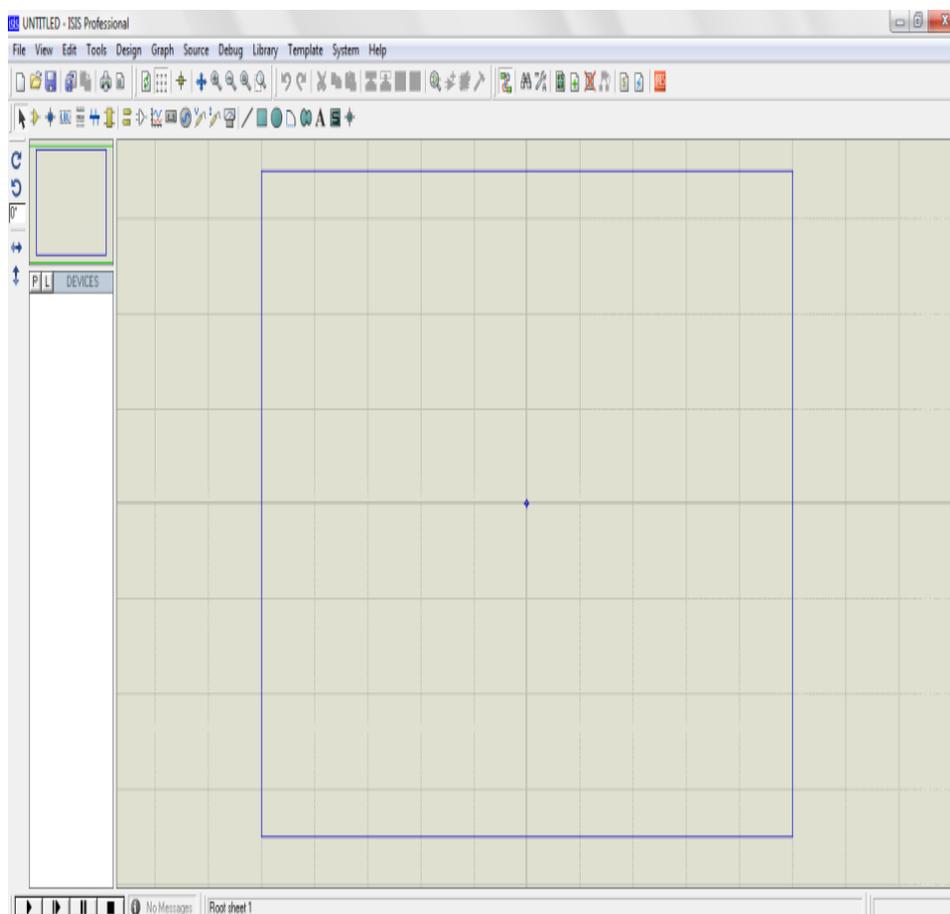


Figura 2.11 Ventana de Trabajo de Proteus

### 2.4.3 LPCXpresso IDE (Eclipse)

El LPCXpresso IDE fue desarrollado por “CodeRed” junto a NXP.

El mismo incluye un entorno de Eclipse específicamente adaptado para interactuar con el target board. En la figura 2.12 se muestra el entorno del LPCXpresso IDE.

El entorno Eclipse utiliza algunos conceptos que no siempre son comunes a otros entornos de desarrollo por lo que vamos a ver algunos de ellos.

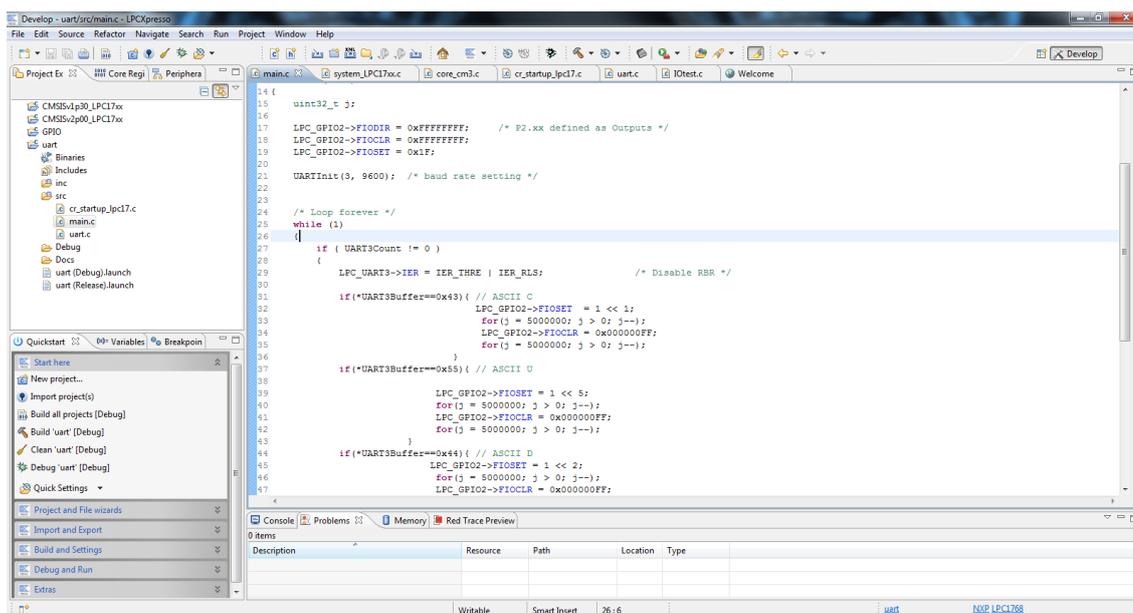


Figura 2.12 Entorno LPCXpresso IDE

**Workspace:** Es el contenedor de 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.

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

En general utilizaremos el workspace para intercambiar proyectos (en el sentido convencional de la palabra) ya que el mismo incluirá todas las bibliotecas necesarias.

Al programar en C, generalmente el proceso que se sigue es el siguiente mostrado en la figura 2.13.

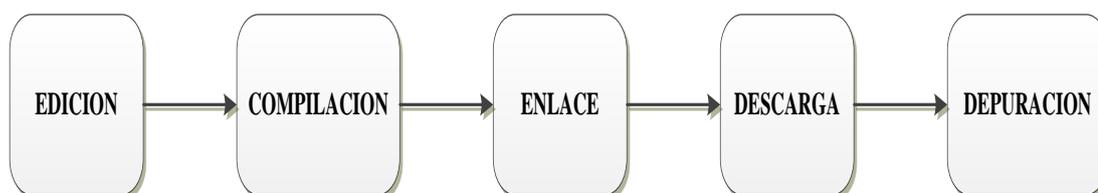


Figura 2.13 Proceso de programar en C en LPCXpresso IDE. [6]

Para generar un programa ejecutable. Primero necesitaremos ejecutar el compilador. Este antes de empezar a compilar, ejecuta el pre-procesador. Cuando se tienen todos los archivos .o, se puede proceder al enlace de los archivos para generar un

archivo ejecutable. La salida generalmente es un archivo .axf o .elf que contienen el código ejecutable. Luego se puede pasar a un archivo .hex.

## **CAPÍTULO 3**

### **DESARROLLO DEL PROYECTO Y EJERCICIOS DE PRUEBA**

Para el diseño, desarrollo e implementación este proyecto se hizo uso de las tarjetas de desarrollo AVR Butterfly, LPCXpresso 1769 y “Motor Control Kit”, las que en conjunto forman la estación de trabajo, la cual se utilizará para realizar la comunicación UART por medio de radiofrecuencia. La estación de trabajo consta de dos partes principales, la transmisora y la receptora. Los ejercicios 1 y 2, propuesto en este capítulo son una introducción al uso de las tarjetas y como estas realizan la comunicación UART. El ejercicio 3, es una breve explicación del funcionamiento del proyecto de graduación.

### 3.1 Diseño y desarrollo de la estación de trabajo

Es un conjunto de tarjetas, componentes electrónicos y fuentes de alimentación DC. Su implementación física se muestra en la figura 3.1. La estación de trabajo consta de dos partes: La transmisión que el encargada de enviar el dato, la parte receptora que consta de dos componentes las tarjetas LPCXpresso 1769 y “Motor Control Kit”.

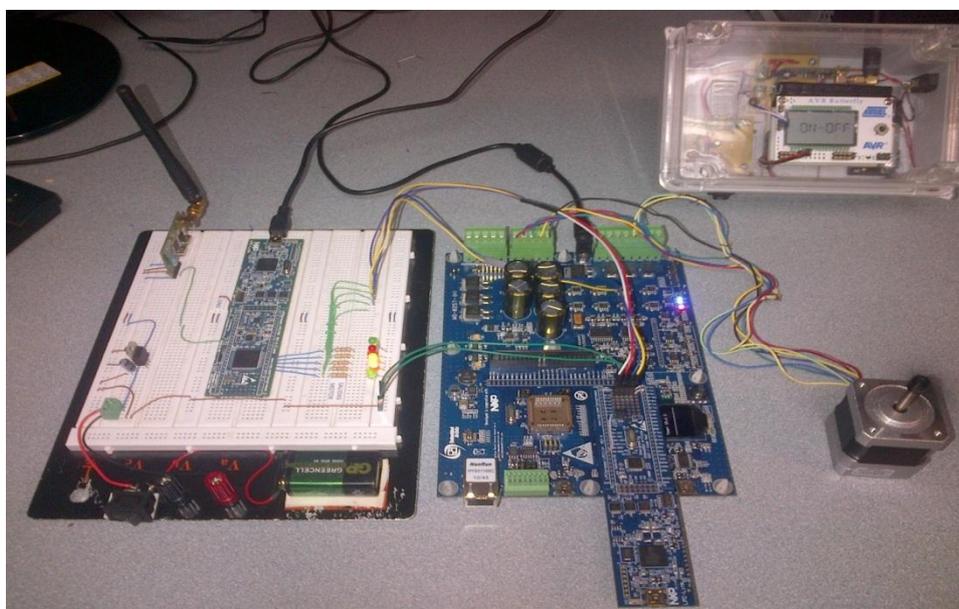


Figura 3.1 Estación de trabajo

La estación de trabajo es modular, evitando así un cableado excesivo y proporcionando un alcance máximo de 330 metros entre el transmisor y receptor, sin que exista algún tipo de distorsión en la señal. En la figura 3.1 se puede apreciar la implementación física de la estación de trabajo para el desarrollo del proyecto de graduación.

### 3.2 Transmisor: tarjeta AVR Butterfly

La transmisión del dato es realizada por la tarjeta AVR Butterfly, la fuente de alimentación DC para energizar el modulo wireless HM-TR232, un inversor de señal, que es el encargado de invertir la señal, para que esta llegue con lógica positiva al pin TX del módulo inalámbrico HM-TR232. Cuando esta señal llega al pin TX, el modulo la convierte en una modulación angular de amplitud constante (modulación FSK). En la figura 3.2 se muestra la implementación del circuito transmisor, la tarjeta AVR Butterfly, fuente de alimentación para del módulo HM-TR232 y el inversor de señal. En la figura 3.3 se puede ver el detalle de los componentes del transmisor.

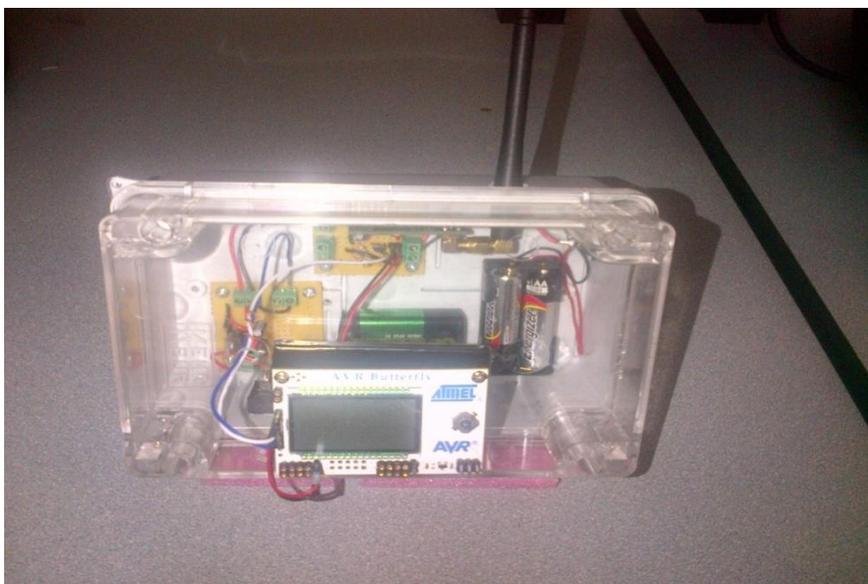


Figura 3.2 Transmisor: Tarjeta AVR Butterfly y módulo HM-TR232

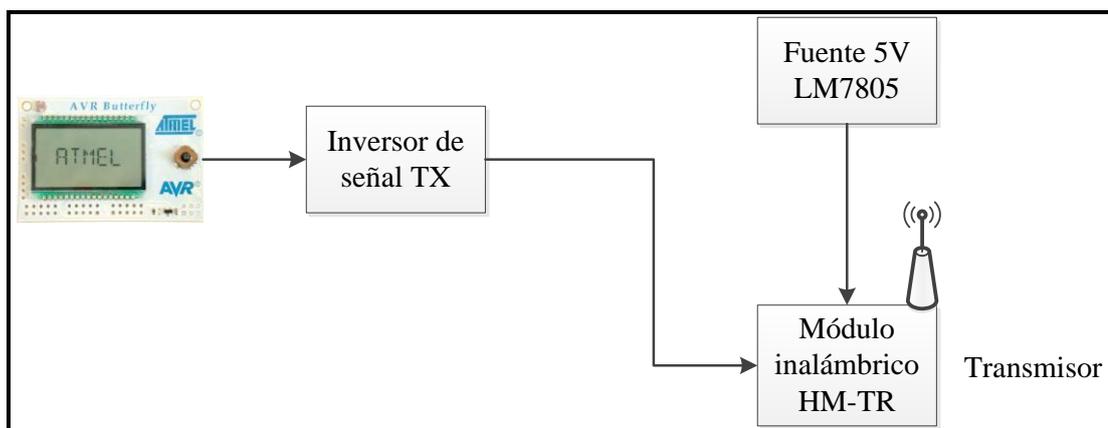


Figura 3.3 Detalle de componentes del transmisor.

### 3.3 Receptor: LPCXpresso1769 y “Motor Control Kit”

El receptor está constituido de la tarjeta de desarrollo LPCXpresso1769, una fuente de alimentación DC de 5V para energizar el módulo HM-TR232, similar al usado en el receptor, el módulo inalámbrico antes mencionado y el “Motor Control Kit” (MCK). La recepción del dato a través del módulo inalámbrico, el cual realiza la desmodulación de la señal transmitida, que consiste en saber distinguir las dos diferentes frecuencias de entrada para poder generar la onda original. Una vez que la señal es como la enviada por la AVR Butterfly, el módulo inalámbrico lo envía al puerto UART3 de la tarjeta LPCXpresso 1769 por el pin TX3, los datos son procesados y se convierten en las instrucciones para el control del motor, que a su vez son las entradas para el “Motor Control Kit” efectúe las operaciones previamente programadas.

En la figura 3.4 se muestra la implementación del circuito receptor, la tarjeta de desarrollo LPCXpresso1769, la fuente de alimentación para el módulo inalámbrico HM-TR232 y “Motor control kit”.

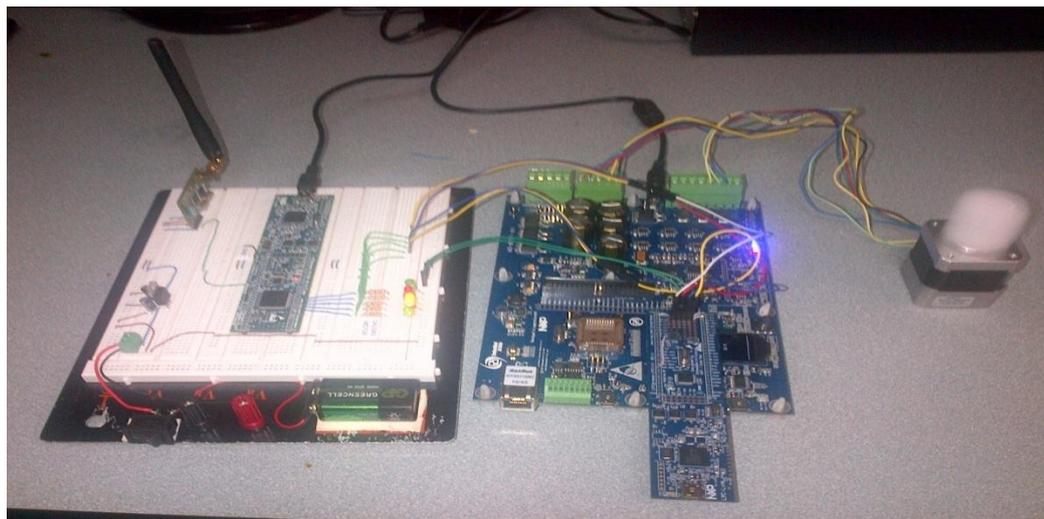


Figura 3.4 Receptor: Tarjeta LPCXpresso, MCK y módulo HM-TR232

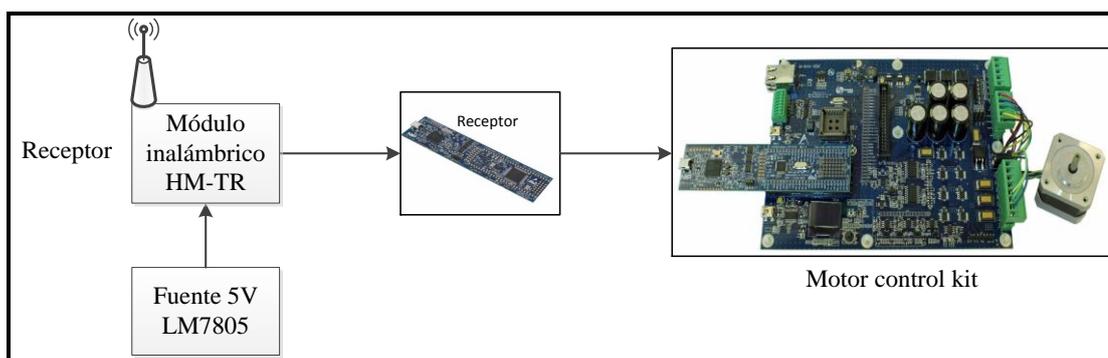


Figura 3.5 Detalle de componentes del receptor.

### **3.4 Ejercicios a desarrollar e implementar**

La implementación de los ejercicios mostrados tiene la finalidad de demostrar la comunicación UART por medio de las tarjetas AVR Butterfly y LPCXpresso, haciendo uso de la tarjeta “Motor Control Kit” para realizar el control al motor BLDC.

#### **3.4.1 Ejercicio 1: Comunicación UART**

Este ejercicio tiene como objetivo transmitir un dato desde la tarjeta AVR Butterfly hacia el computador (Virtual Terminal), este dato se genera de acuerdo al valor que se selecciona en el joystick. El funcionamiento del programa fuente cargado en la memoria flash de la tarjeta es el diagrama de flujo presentado en la figura 3.6.

Al iniciar se energiza la tarjeta y presionando el botón de arriba en la pantalla LCD aparecerá el mensaje “ANA JIMENEZ – XAVIER LOPEZ” y quedaran habilitadas las funciones del joystick.

En el Joystick de la AVR Butterfly se ejecutaran las funciones de posición configuradas previamente, presionando el botón de

arriba en la pantalla del virtual terminal de imprimirá el carácter “U” y en la pantalla LCD se muestra la palabra “ARRIBA”.

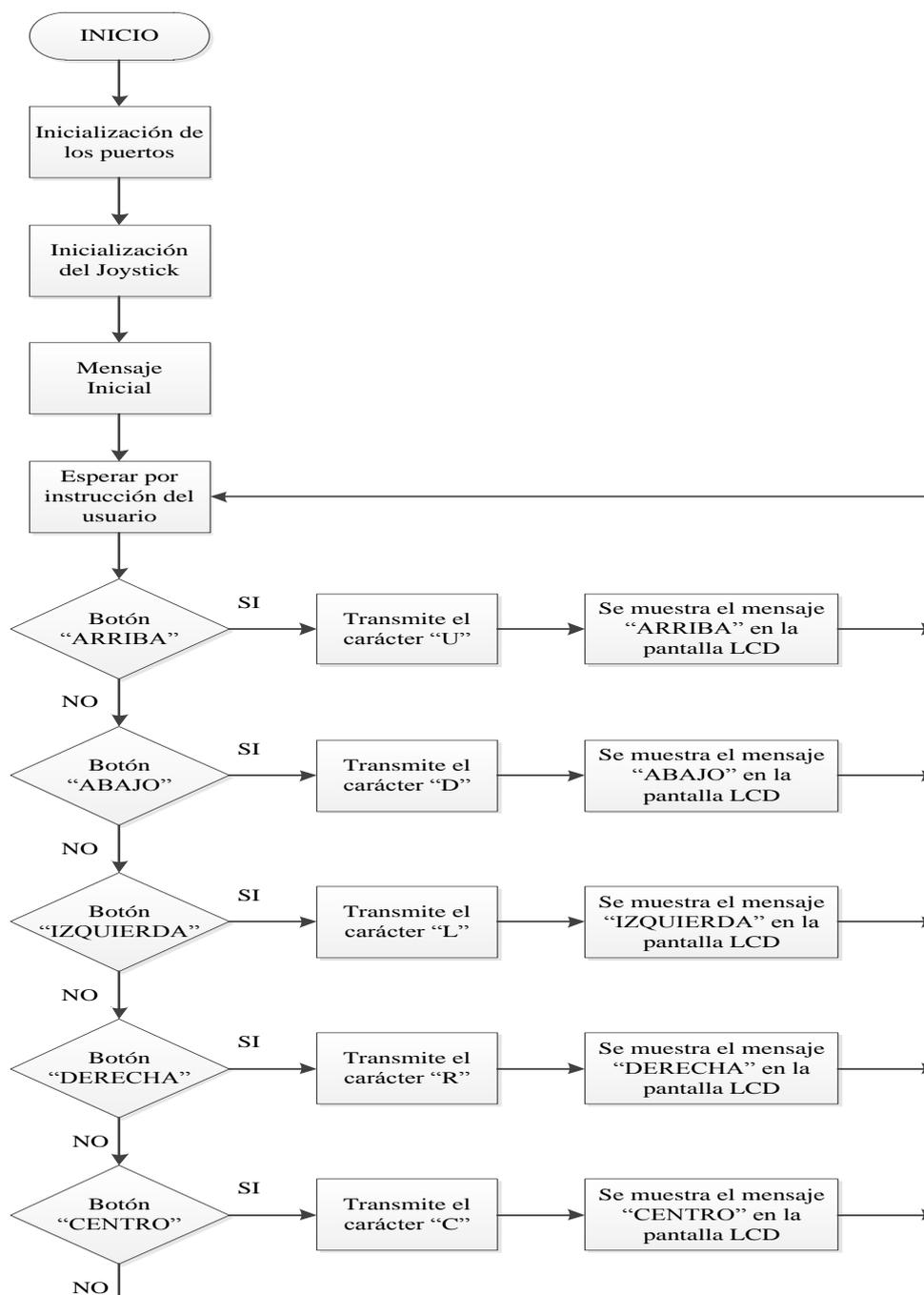


Figura 3.6 Diagrama de Flujo del ejercicio 1.

Si se presiona el botón de abajo en la pantalla del virtual terminal de imprimirá el carácter “D” y en la pantalla LCD se muestra la palabra “ABAJO”. Al presionar el botón de la izquierda en la pantalla del virtual terminal de imprimirá el carácter “L” y en la pantalla LCD se muestra la palabra “IZQUIERDA”. Presionando el botón de la derecha en la pantalla del virtual terminal de imprimirá el carácter “R” y en la pantalla LCD se muestra la palabra “DERECHA”. El botón del centro al ser presionado, imprimirá en la pantalla del virtual terminal el carácter “C” y en la pantalla LCD se muestra la palabra “CENTRO”.

Los componentes necesarios para la realización del ejercicio son los siguientes: una tarjeta electrónica AVR Butterfly, que es alimenta por el puerto B o puerto D con un par de baterías AA (1.5 V c/u) y para realizar la comunicación con la PC se requiere de un cable serial RS232 con adaptador de USB a DB9.

Se puede concluir que el ejercicio realizado, hace una introducción a la comunicación con la tarjeta AVR Butterfly por el puerto serial, para su posterior aplicación en el ejercicio 3 en conjunto con las tarjetas LPCXpresso1769 y Motor control kit en el control del motor BLDC.

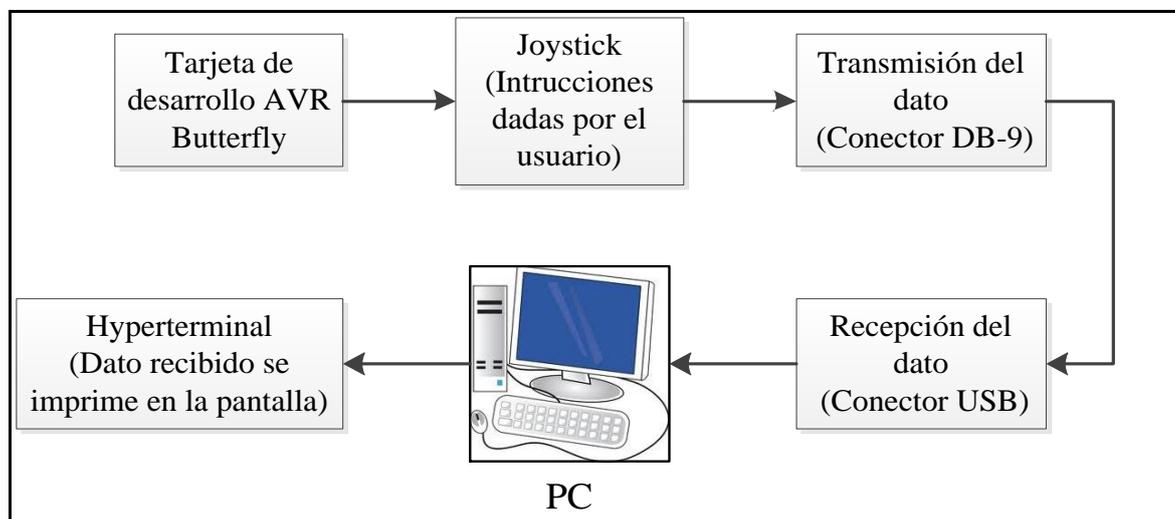


Figura 3.7 Diagrama de Bloques del ejercicio 1.

### 3.4.2 Ejercicio 2: Comunicación UART LPCXpresso

En el siguiente ejercicio se hará uso de la tarjeta de desarrollo LPCXpresso1769 para realizar la comunicación UART. Para efectuar la comunicación UART se utiliza la configuración Maestro-Esclavo, una botonera conectado al maestro habilitara la transmisión del dato hacia el esclavo que recibe el dato que esta representado por un carácter previamente seleccionado en el maestro, a la salida del esclavo esta conectado un Led que nos indicara que el dato enviado fue recibido correctamente. El procedimiento de la comunicación se detalla en el diagrama de flujo correspondiente a la figura 3.8.

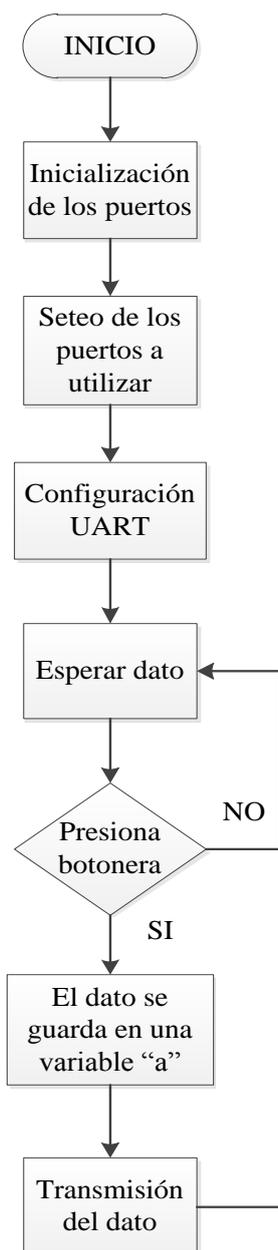
Se carga el programa transmisor y receptor en cada una de las tarjetas de desarrollo LPCXpresso1769. Se realizan las conexiones para la botonera en la parte transmisora y a la salida del receptor el Led. Presionando la botonera se habilitará la transmisión del carácter que es transmitido por el puerto UART3 en el pin TX3 de la tarjeta LPCXpresso. El carácter se recibe por el puerto UART3 en el pin RX3 de la tarjeta LPCXpresso y para confirmar su correcta recepción se colocó un Led como indicador.

Para la realización de este ejercicio se requerirá de los siguientes componentes: dos tarjetas LPCXpresso1769 una se empleará como transmisor y la otra como receptor, dos cables de alimentación USB para cada una de las tarjetas LPCXpresso1769, una botonera, una resistencia  $220\Omega$  para protección del puerto de la tarjeta LPCXpresso1769, un led, una resistencia de  $330\Omega$ , que se colocará como indicador que se está recibiendo los datos, todos estos componentes son colocados en un Protoboard.

Se presenta el diagrama de bloques del ejercicio 2 de los componentes para la transmisión y recepción del dato

corresponde a la figura 3.9; los componentes para la habilitación del envío del dato y el indicador del dato recibido.

### Transmisor



### Receptor

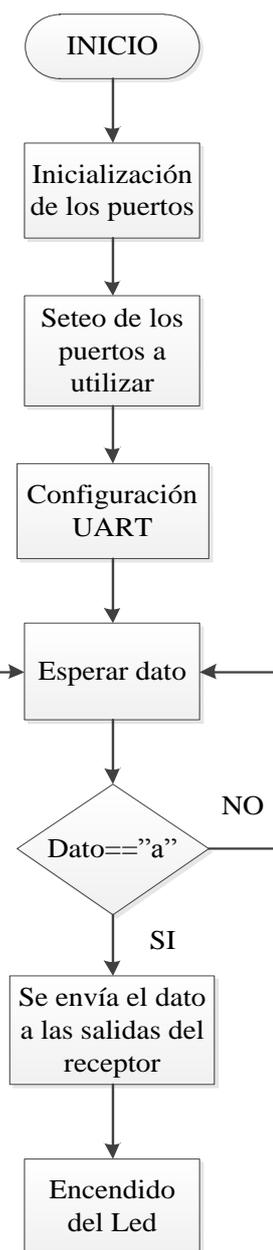


Figura 3.8 Diagrama de Flujo del ejercicio 2.

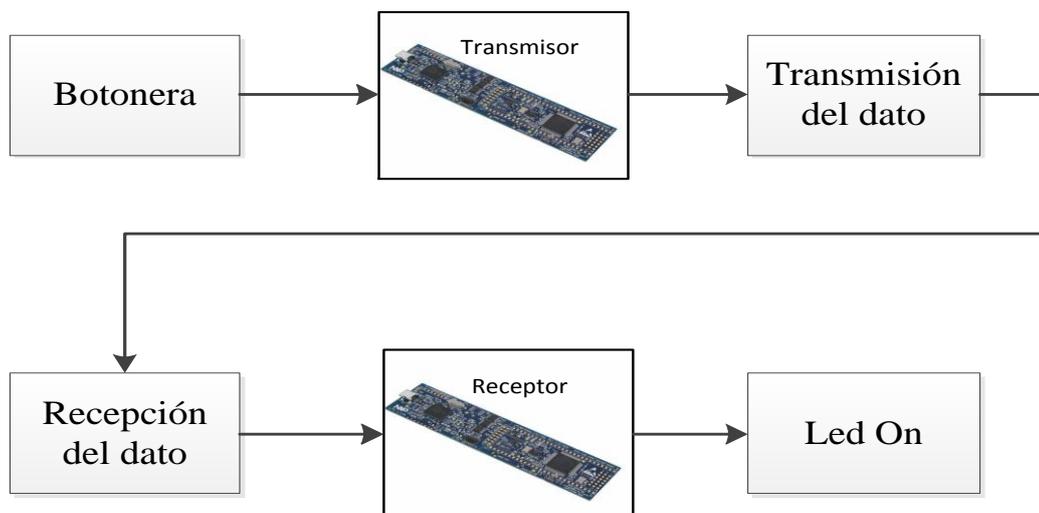


Figura 3.9 Diagrama de Bloques del ejercicio 2.

En este ejercicio se pretende familiarizarse con las opciones que proporciona la tarjeta LPCXPRESSO para realizar la comunicación UART. Es una introducción a la comunicación entre tarjetas, analizando el dato a enviar, la velocidad de transferencia, que debe ser igual para ambas tarjetas para que no exista conflicto de recepción.

### 3.4.3 Ejercicio 3: Control de un motor BLDC

En este ejercicio se programa, diseña e implementa el proyecto de graduación, el cual consiste en el control de un motor BLDC, las instrucciones de operación para el control son enviadas por el joystick de la AVR Butterfly y se muestra en la pantalla LCD de la misma el envío de dicha instrucción se emplea la radiofrecuencia

haciendo uso de los módulos inalámbricos, tanto para transmisión y recepción, una vez que se desmodula señal esta, la recibe la tarjeta de desarrollo LPCXpresso por el puerto RX3 y luego las instrucciones son procesadas y enviadas a las salidas de la misma tarjeta. Estas salidas son las instrucciones para el “Motor Control Kit” que tiene acoplado el motor BLDC. Las instrucciones que son enviadas por el joystick de la tarjeta AVR Butterfly se detallan en el siguiente diagrama de flujo:

Para habilitar la instrucciones del joystick, se presiona el botón de arriba, inmediatamente se mostrara en la pantalla el mensaje “ANA JIMENEZ – XAVIER LOPEZ”. Luego el joystick quedara listo para ejecutar cualquiera de las cinco posiciones que posee, por ordenes del usuario.

Como primer paso se debe energizar las tarjetas y módulos inalámbricos y esperar a que el programa de inicialización en el “Motor Control Kit” este totalmente cargado en el OL Led de esta tarjeta. Luego se puede proceder a efectuar el control del motor. Para encender el motor se debe presionar el botón de izquierda en el Joystick, en la pantalla LDC se mostrará el mensaje “ON-OFF”, lo cual indica que el motor esta en funcionamiento. Si se

presiona el botón de arriba en el Joystick, en la pantalla LDC se mostrará el mensaje “RISE SPEED”, con que se incrementa la velocidad del motor.

### Para el transmisor

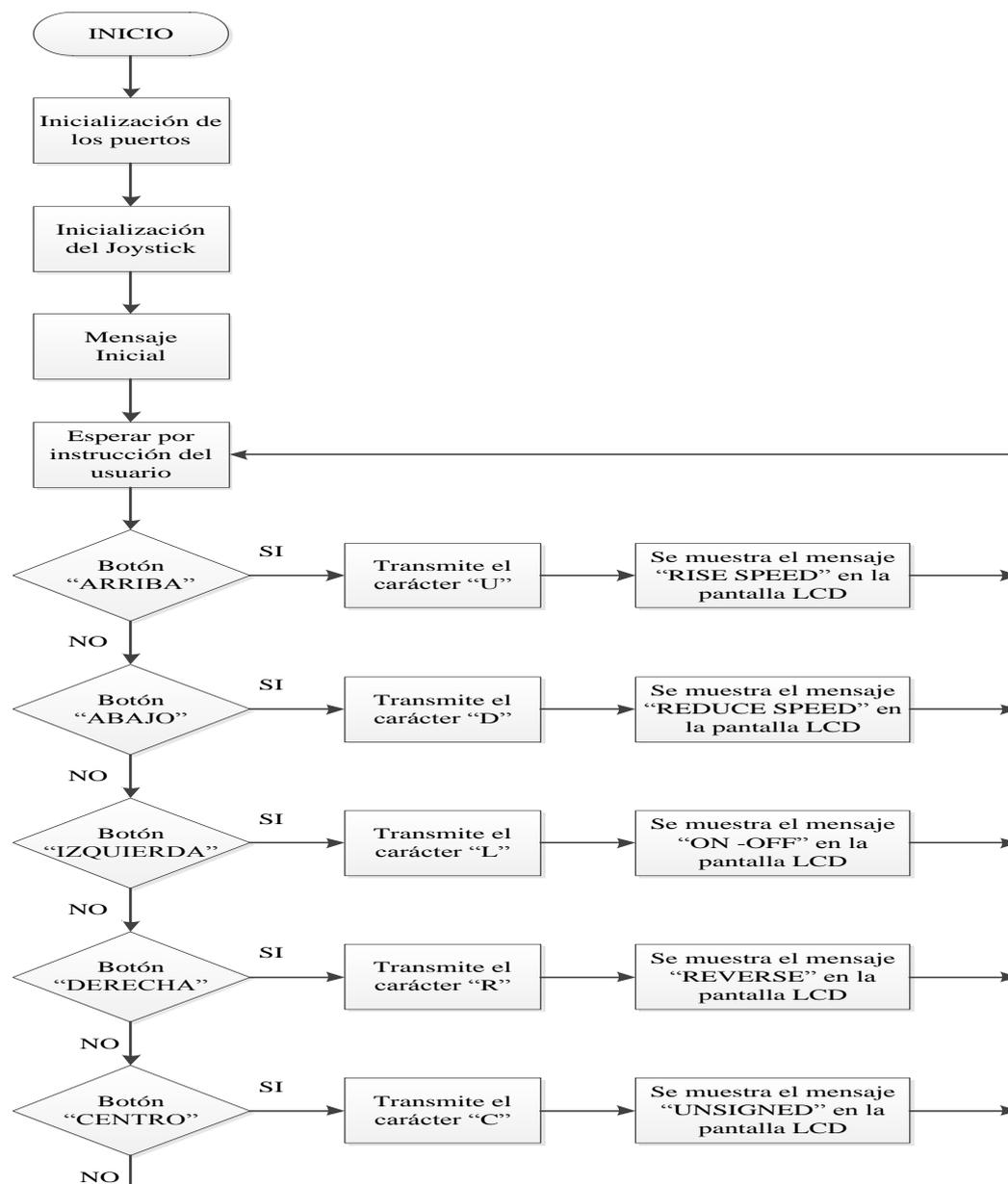


Figura 3.10 Diagrama de Flujo (Transmisor) del ejercicio 3.

## Para el receptor



Figura 3.11 Diagrama de Flujo (Receptor) del ejercicio 3.

Si se presiona el botón de arriba en el Joystick, en la pantalla LDC se mostrará el mensaje “REDUCE SPEED”, con que se reduce la velocidad del motor. Si se presiona el botón de la derecha en el Joystick, en la pantalla LDC se mostrará el mensaje “REVERSE”, con que se cambia el giro del motor.

Si se presiona el botón del centro en el Joystick, en la pantalla LDC se mostrará el mensaje “UNSIGNED”, este botón no realiza ninguna operación en el motor.

Para la implementación de este ejercicio se requiere de los siguientes componentes: para la parte transmisora; una tarjeta AVR-Butterfly con su respectiva alimentación externa que consta de un porta pila con dos baterías AA (1.5V c/u); dos fuentes de alimentación de 5V DC para cada uno de los módulos inalámbricos; un circuito inversor de señal que consta de un transistor 2N2222, una resistencia de 1K $\Omega$  y una resistencia de 220 $\Omega$ ; para la parte receptora; una tarjeta LPCXpresso1769, con su respectivo cable de alimentación USB; un protoboard en que armara el circuito receptor. Finalmente el “Motor Control Kit” con su alimentación de 110V AC; el motor BLDC y un osciloscopio

para observar la forma de las señales que están siendo transmitida y recibidas por los módulos inalámbricos.

En la figura 3.12 se presenta el diagrama de bloques del ejercicio 3, se indica la conexión de los componentes empleados con su respectiva función; la alimentación requerida por cada uno de los módulos inalámbricos y la tarjeta “Motor Control Kit”.

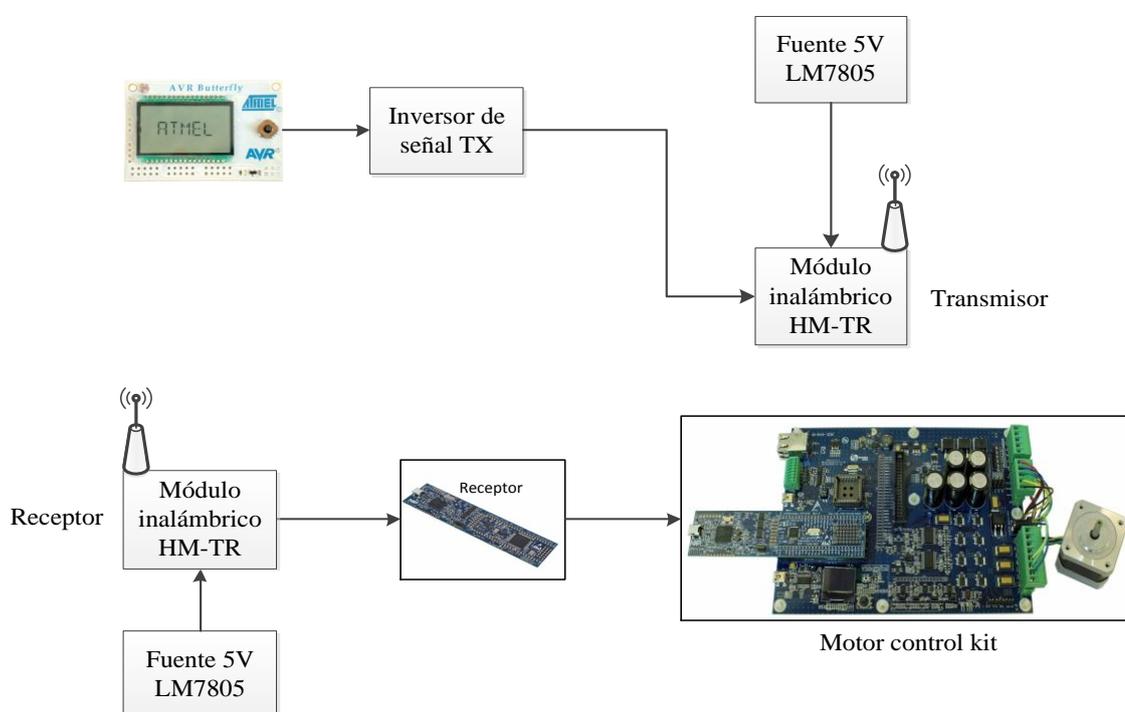


Figura 3.12 Diagrama de Bloque del ejercicio 3.

El objetivo de este ejercicio la recopilación de los dos ejercicios anteriores incluyendo la aplicación de control de un motor BLDC, además de la transmisión del dato se lo realiza por medio de radiofrecuencia, dando grandes ventajas en la implementación, entre las que están reducción de cableado para la transmisión y

recepción, la portabilidad de la estación de trabajo y diseño modular de la misma.

## **CAPÍTULO 4**

### **IMPLEMENTACIÓN DEL PROYECTO DE GRADUACIÓN**

En este capítulo se implementan los ejercicios propuestos en el capítulo 3. Se hará uso de la estación de trabajo, el Hyperterminal, el programa de simulación PROTEUS, el AVR Studio 4 y el LPCXpresso IDE. El objetivo principal de los ejercicios es demostrar el funcionamiento de la comunicación UART y los recursos que esta posee.

#### 4.1 Especificación

En este ejercicio se programa, diseña e implementa el proyecto de graduación, el cual consiste en el control de un motor BLDC, las instrucciones de operación para el control son enviadas por el joystick de la AVR Butterfly y se muestra en la pantalla LCD de la misma el envío de dicha instrucción se la efectúa por medio de radiofrecuencia haciendo uso de los módulos inalámbricos, tanto para transmisión y recepción, una vez que se desmodula señal esta, la recibe la tarjeta de desarrollo LPCXpresso por el puerto RX3 y luego las instrucciones son procesadas y enviadas a las salidas de la misma tarjeta. Estas salidas son las instrucciones para el “Motor Control Kit” que tiene acoplado el motor BLDC. Las instrucciones que son enviadas por el joystick de la tarjeta AVR Butterfly, para habilitar la instrucciones del joystick, se presiona el botón de arriba, inmediatamente se mostrara en la pantalla el mensaje “ANA JIMENEZ – XAVIER LOPEZ”. Luego el joystick quedara listo para ejecutar cualquiera de las cinco posiciones que posee, por ordenes del usuario.

Como primer paso se debe energizar las tarjetas y módulos inalámbricos y en el Motor control kit esperar a que el programa de inicialización este totalmente cargado en el OL Led de esta tarjeta. Luego se puede proceder a efectuar el control del motor. Para encender

el motor se debe presionar el botón de izquierda en el Joystick, en la pantalla LDC se mostrará el mensaje “ON-OFF”, lo cual indica que el motor esta en funcionamiento. Si se presiona el botón de arriba en el Joystick, en la pantalla LDC se mostrará el mensaje “RISE SPEED”, con que se incrementa la velocidad del motor.

Si se presiona el botón de arriba en el Joystick, en la pantalla LDC se mostrará el mensaje “REDUCE SPEED”, con que se reduce la velocidad del motor.

Si se presiona el botón de la derecha en el Joystick, en la pantalla LDC se mostrará el mensaje “REVERSE”, con que se cambia el giro del motor.

Si se presiona el botón del centro en el Joystick, en la pantalla LDC se mostrará el mensaje “UNSIGNED”, este botón no realiza ninguna operación en el motor.

#### **4.1.1 Simulaciones**

En diseño de la AVR Butterfly realizado en PROTEUS se carga el archivo Ejercicio\_3T.hex creado en el programa AVR Studio 4 al compilar el .C. Se muestra en la figura 4.1 los nombres de los integrantes del proyecto de graduación.

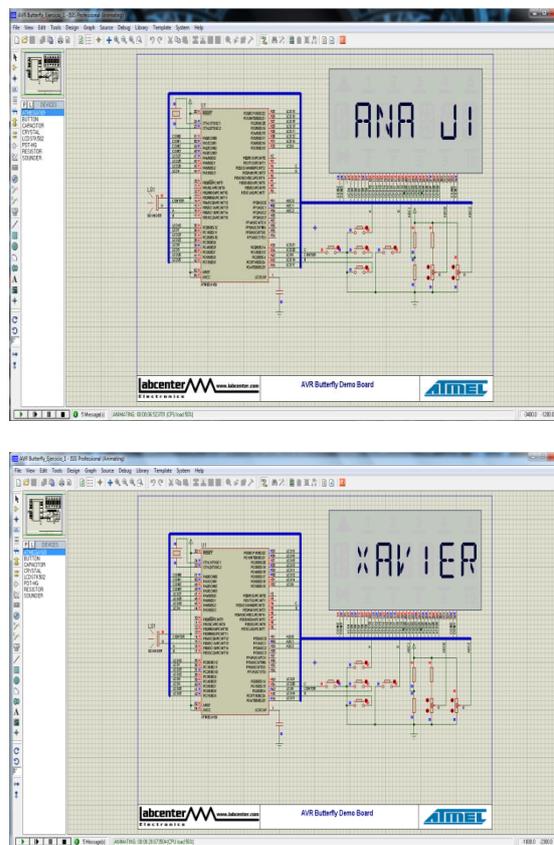


Figura 4.1 Nombre de los integrantes del proyecto.

En la figura 4.2 se muestra el mensaje asociado con cada instrucción de joystick que es mostrado en la pantalla LCD.

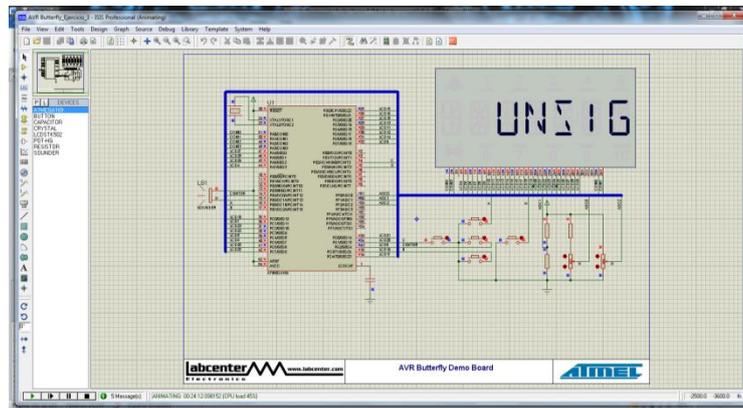
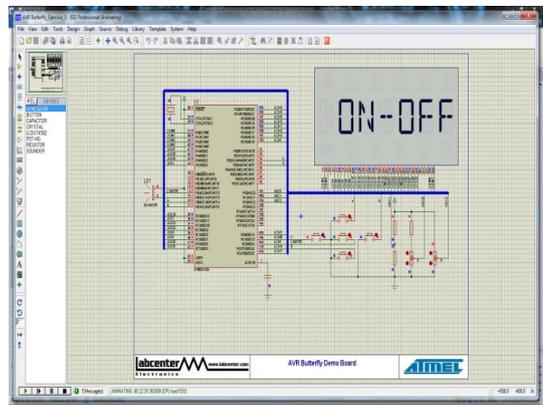
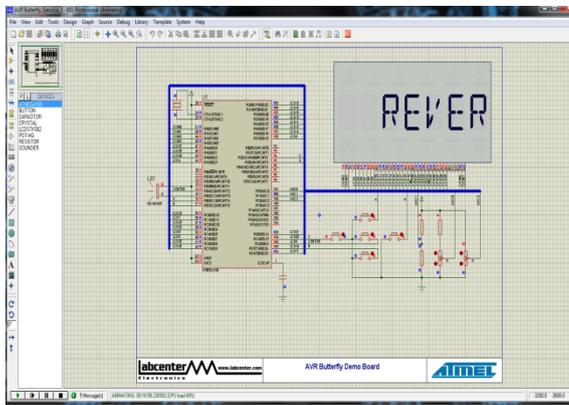
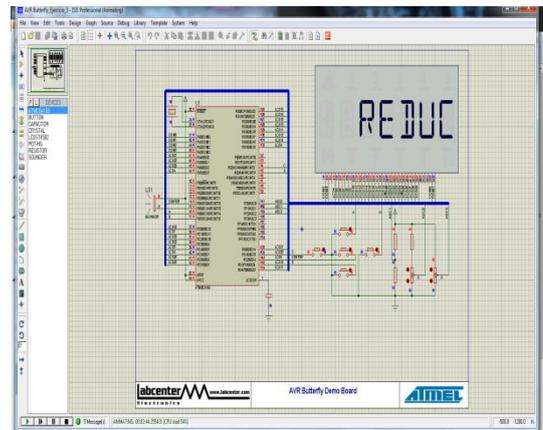
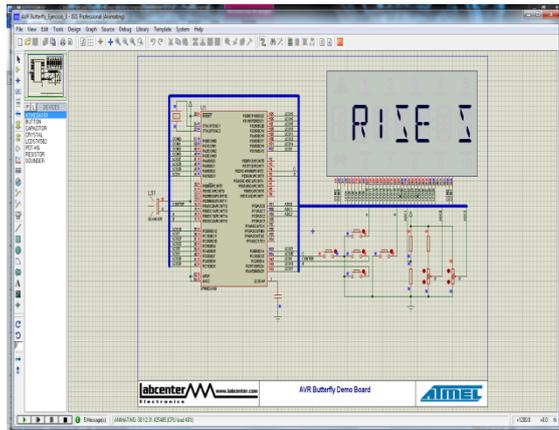


Figura 4.2 Mensajes de cada uno de los botones del joystick.

## CONCLUSIONES

1. La importancia de la comunicación UART para permitir la interacción de las tarjetas de desarrollo en la transmisión y recepción de las instrucciones de control para el motor BLDC, además de uso de módulos inalámbricos que son aspecto relevante para el proyecto, lo que demuestra que la comunicación UART entre la tarjeta AVR Butterfly que es el encargado de transmitir las instrucciones dadas por usuario a través del joystick de esta tarjeta hacia la tarjeta LPCXpresso 1769 que es la que recibe estas instrucciones, es muy versátil y cual sea el dato a ser enviado, la comunicación UART nos garantiza que el dato no sufrirá ninguna alteración en su contenido.
2. Para la programación de las tarjetas de desarrollo disponemos de un entorno muy amigable al momento de escribir, compilar y ejecutar el código fuente para cada una de ellas; lo que representa un punto muy importante para el desarrollo del proyecto ya que no se requiere de un conocimiento avanzado de control de motores BLDC, lo que proporciona al usuario realizar modificaciones en el funcionamiento a su conveniencia sin ningún tipo de contratiempos, garantizando así la operatividad de la estación de trabajo bajo instrucciones previamente establecidas.

3. Los ejercicios de prueba realizados con las tarjetas AVR Butterfly, LPCXpresso 1769 muestran que, sea interactuando con la PC al transmitir un dato y mostrarlo en pantalla o en el encendido de un led, nos asegura que el dato que se envía no sufre alteraciones al momento de transmitirlo o recibirlo y mas aun cuando se emplean los módulos inalámbricos para realizar la comunicación UART tampoco se ve alterada, lo cual nos indica que el sistema modular de la estación de trabajo es el indicado para realizar esta comunicación sin ningún inconveniente.
  
4. La pantalla grafica (Oled) que posee la tarjeta "Motor Control Kit" nos permite apreciar el funcionamiento correcto de las instrucciones de control del motor; cuando hace el arranque del mismo, la variación de su velocidad y su cambio de giro, lo que garantiza que las instrucciones enviadas desde el transmisor hacia el receptor son las mismas que el usuario selecciona cuando opera el joystick de la tarjeta AVR Butterfly, lo que indica la óptima operación de la estación de trabajo.

## RECOMENDACIONES

1. Un punto fundamental de la configuración de la comunicación UART, es tener en cuenta la velocidad de transferencia de datos (Baud Rates), el valor establecido debe ser el mismo para la parte transmisora representada por la AVR Butterfly, como para la parte receptora, la tarjeta LPCXpresso1769 para evitar problemas al momento de realizar la comunicación.
2. En la implementación de todo proyecto o prototipo se debe tomar en cuenta que su uso por parte de otras personas, las cuales estén o no familiarizado con la operatividad del dispositivo, se debe en consideración la creación de una guía del usuario que permita el uso del mismo sea lo mas fácil y sin ningún tipo de inconveniente previo que ocasione un mal funcionamiento.
3. Una revisión previa a la documentación de los dispositivos utilizados en la implementación de este proyecto, ya sean estos los manuales de usuario proporcionados por cada uno de los fabricantes de dichos dispositivos, fundamentos teóricos de control a motores BLDC, y principios de radiofrecuencia para que hacer un uso mas eficiente de la estación de trabajo.

4. Un aspecto importante cuando se haga uso de módulos de radiofrecuencia es analizar su configuración de pines, la comunicación del dato que este maneja, su velocidad de transferencia de datos y modos de operación, para que al momento de realizar la comunicación UART se eviten problemas que a primera vista se asocian al cableado del proyecto, problemas de programación, etc.

# **ANEXOS**

# **Anexo I**

**Código Fuente del Ejercicio 1:**

**Comunicación UART**

## **CODIGO FUENTE**

```
//----- SEMINARIO DE GRADUACION-----
```

```
/* Profesor: Ing. Carlos Valdivieso A.
```

```
/* Creado: 20/03/2012
```

```
/* Autores: Ana Jiménez & Xavier López
```

```
//-----
```

```
//***** 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)
```

```
{
```

```
    PGM_P statetext = PSTR("ANA_JIMENEZ-XAVIER_LOPEZ");
```

```
    uint8_t input;
```

```
    char *cadena;
```

```

int i;

ACSR = (1<<ACD);           // Deshabilita comparador analógico
DIDR0 = (7<<ADC0D);       // 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 Baud Rates

//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("CENTRO");  
        Usart_Tx('C');  
        break;  
    case KEY_NEXT:  
        statetext = PSTR("DERECHA");  
        Usart_Tx('R');  
        break;  
    case KEY_PREV:  
        statetext = PSTR("IZQUIERDA");  
        Usart_Tx('L');  
        break;  
    case KEY_PLUS:  
        statetext = PSTR("ARRIBA");  
        Usart_Tx('U');  
        break;  
    case KEY_MINUS:  
        statetext = PSTR("ABAJO");  
        Usart_Tx('D');  
        break;  
    default:  
        break;  
}
```

```
}  
    return 0;  
}  
  
//***** Fin del Código*****
```

# **Anexo II**

**Código Fuente del Ejercicio 2:**

**Comunicación UART LPCXpresso**

## Para el transmisor

/\*\*\*\*\*

\* Autores: Ana Jiménez & Xavier López

\$

\* Project: uart: Simple UART echo for LPCXpresso 1700

\* File: uarttest.c

\* Description:

\* LPCXpresso Baseboard uses pins mapped to UART3 for  
\* its USB-to-UART bridge. This application simply echos  
\* all characters received.

\*\*\*\*\*

\* Software that is described herein is for illustrative purposes only which provide customers with programming information regarding the products. This software is supplied "AS IS" without any warranties. NXP Semiconductors assumes no responsibility or liability for the use of the software conveys no license or title under any patent, copyright, or mask work right to the product. NXP Semiconductors reserves the right to make changes in the software without notification. NXP Semiconductors also make no representation or warranty that such application will be suitable for the specified use without further testing or modification.

\*\*\*\*\*/

/\*\*\*\*\*

\* History

\* 2010.07.01 ver 1.01 Added support for UART3, tested on LPCXpresso 1700

\* 2009.05.27 ver 1.00 Preliminary version, first Release

```

*****/

/*****Librerías empleadas *****/

#include <cr_section_macros.h>

#include <NXP/crp.h>

#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];

/***** Programa principal *****/

int main (void)

{

    const char* encender = 'a';

    uint32_t BOTON;

    UARTInit(3, 9600);    /*Configuración del Baud Rate*/

    LPC_GPIO2->FIODIR = 0xFFFFFEFF;    /* P2 definido como
salidas*/

    LPC_GPIO2->FIOCLR = 0xFFFFFFFF;    /*Apaga todos los
Led's*/

    LPC_GPIO2->FIOMASK= 0x00000000;

```

```
while(1)
{
    if(LPC_GPIO2->FIOPIN==0x00000000 ){
        UARTSend(3, (uint8_t *)encender , strlen(encender) );
    }
}
}
```

**/\*\*\*\*\*\* Fin del código\*\*\*\*\*\*/**

### **Para el receptor**

**/\*\*\*\*\*\***

\* Autores: Ana Jiménez & Xavier López

\$

\* Project: uart: Simple UART echo for LPCXpresso 1700

\* File: uarttest.c

\* Description:

\* LPCXpresso Baseboard uses pins mapped to UART3 for  
\* its USB-to-UART bridge. This application simply echos  
\* all characters received.

**\*\*\*\*\***

\* Software that is described herein is for illustrative purposes only which provide customers with programming information regarding the products. This software is supplied "AS IS" without any warranties. NXP Semiconductors assumes no responsibility or liability for the use of the software conveys no license or title under any patent, copyright, or mask work right to the product. NXP Semiconductors

reserves the right to make changes in the software without notification. NXP Semiconductors also make no representation or warranty that such application will be suitable for the specified use without further testing or modification.

```
*****/
/*****
* History
* 2010.07.01 ver 1.01 Added support for UART3, tested on LPCXpresso 1700
* 2009.05.27 ver 1.00 Preliminary version, first Release
```

```
*****/
/*****Librerías empleadas*****/
```

```
#include <cr_section_macros.h>
```

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

```
#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];
```

```
/***** Programa principal*****/
```

```
int main (void)
```

```
{
```

```
    const char* encender = 'a';
```

```

uint32_t j;

UARTInit(3, 9600);    /*Configuración del Baud Rate*/

    LPC_GPIO2->FIODIR = 0xFFFFFEFF;    /* P2 definido como
salidas*/

    LPC_GPIO2->FIOCLR = 0xFFFFFFFF;    /*Apaga todos los
Led's*/

    LPC_GPIO2->FIOMASK= 0x00000000;

while (1)
{
    if ( UART3Count != 0 )
    {
        LPC_UART3->IER = IER_THRE | IER_RLS;

                                                /*Deshabilita RBR
                                                */

        if (UART3Buffer=="a"){
            LPC_GPIO2->FIOSET = 1 << 1;
            for(j = 5000000; j > 0; j--);
            LPC_GPIO2->FIOCLR = 0x000000FF;
            for(j = 5000000; j > 0; j--);
        }

        UART3Count = 0;
    }
}

```

```
LPC_UART3->IER = IER_THRE | IER_RLS | IER_RBR;
```

```
/*
```

```
Rehabilita RBR */
```

```
    }
```

```
  }
```

```
}/***** Fin del código *****/
```

# **Anexo III**

**Código Fuente del Ejercicio 3:**

**Control de un motor BLDC**

## Para el transmisor

//----- SEMINARIO DE GRADUACION-----

/\* Profesor: Ing. Carlos Valdivieso A.

/\* Creado: 28/03/2012

/\* Autores: Ana Jiménez & Xavier López

//-----

**//\*\*\*\*\* 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)

{

PGM\_P statetext = PSTR("ANA\_JIMENEZ-XAVIER\_LOPEZ");

uint8\_t input;

char \*cadena;

```

int i;

ACSR = (1<<ACD);           // Deshabilita comparador analógico
DIDR0 = (7<<ADC0D);       // 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 Baud Rates

    //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 ***** */

Para el receptor

/*----- SEMINARIO DE GRADUACION-----*/

* Profesor: Ing. Carlos Valdivieso A.

* Creado: 29/03/2012

* Autores: Ana Jiménez & Xavier López

-----*/

/** ***** 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 i, j;

    LPC_GPIO2->FIODIR = 0xFFFFFFFF;    /* P2 definidos como salidas */
    LPC_GPIO2->FIOCLR = 0xFFFFFFFF;
    LPC_GPIO2->FIOSET = 0x1F;
    UARTInit(3, 9600);                  /* Configuración del Baud Rate
*/

    while (1)                           /*Lazo infinito*/
    {
        if ( UART3Count != 0 )
        {
            LPC_UART3->IER = IER_THRE | IER_RLS;
/*

Deshabilita 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;  
    LPC_UART3->IER = IER_THRE | IER_RLS | IER_RBR;  
    /*  
Rehabilita RBR */  
        }  
    }  
}  
/****** Fin del código******/
```

# **Anexo IV**

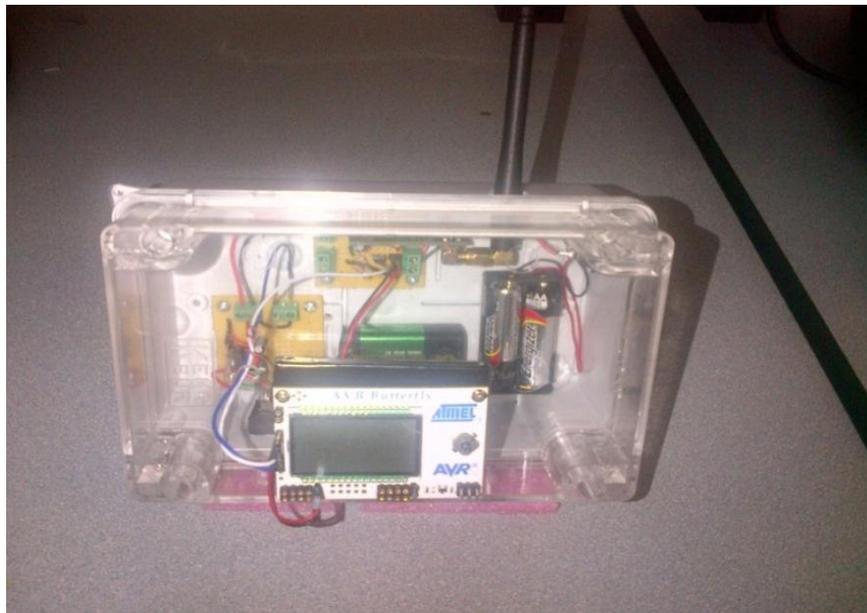
**Manual de Usuario**

## **Guía rápida de uso de las tarjetas AVR Butterfly, LPCXpresso y “Motor Control Kit”**

### **Transmisor AVR Butterfly**

El transmisor esta compuesto de la tarjeta AVR Butterfly, un acondicionador de señal y el modulo inalámbrico.

Para inicialización del transmisor se activan los interruptores para la alimentación de la tarjeta AVR Butterfly y el modulo inalámbrico respectivamente. Posteriormente se procede a habilitar el joystick presionando hacia arriba del mismo para que se muestre en pantalla el mensaje de los integrantes del proyecto de graduación.





### **Receptor LPCXpresso**

El receptor consta de la tarjeta LPCXpresso, modulo inalámbrico y bornes de salidas, que son las señales enviadas a la tarjeta "Motor Control Kit". Se energiza el modulo inalámbrico y por medio de la conexión entre el puerto USB de la PC y el puerto de alimentación de la tarjeta LPCXpresso se energiza la misma. A la salida de la LPCXpresso se encuentran las señales de control que serán enviadas por el joystick de la AVR Butterfly.

Los pines asignados a estas señales son desde el 42 al 46. El orden establecido para cada instrucción es el siguiente:

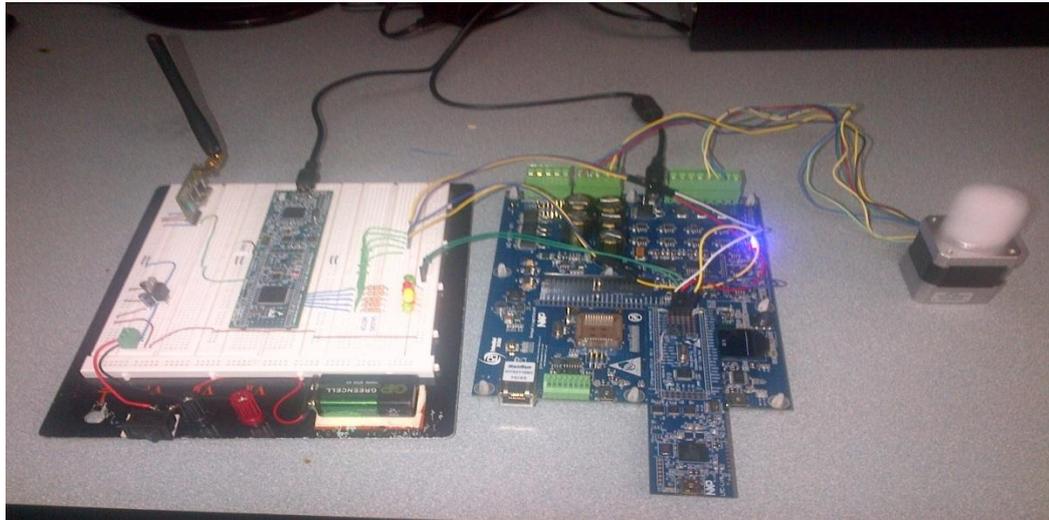
Pin 42: incremento de velocidad

Pin 43: disminución de velocidad

Pin 44: cambio de giro del motor

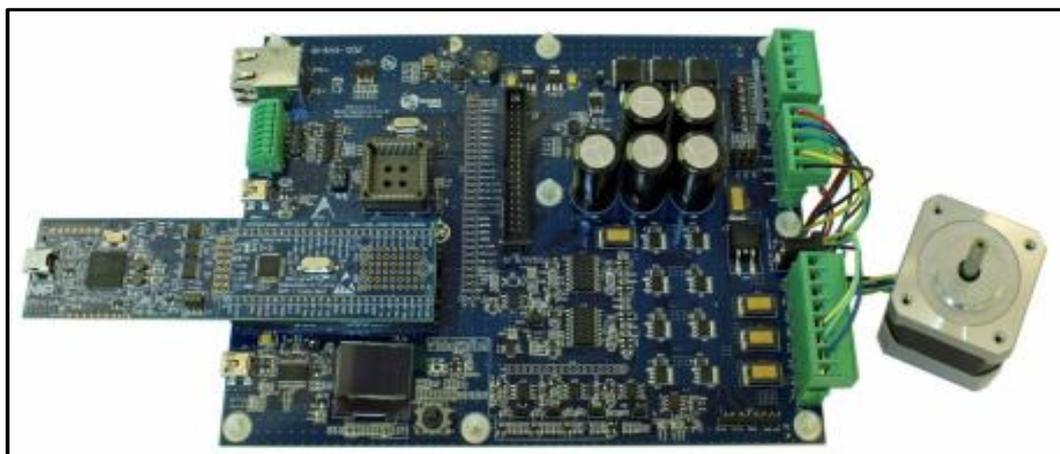
Pin 45: On/off motor

Pin 46: No asignado instrucción alguna



### **Tarjeta “Motor Control Kit”**

La tarjeta “Motor Control Kit consta de la tarjeta LPCXpresso 1114, la tarjeta de desarrollo y el motor BLDC. Se energiza la tarjeta de desarrollo y se acopla previamente el motor a los bornes de alimentación que la tarjeta posee para su efecto.

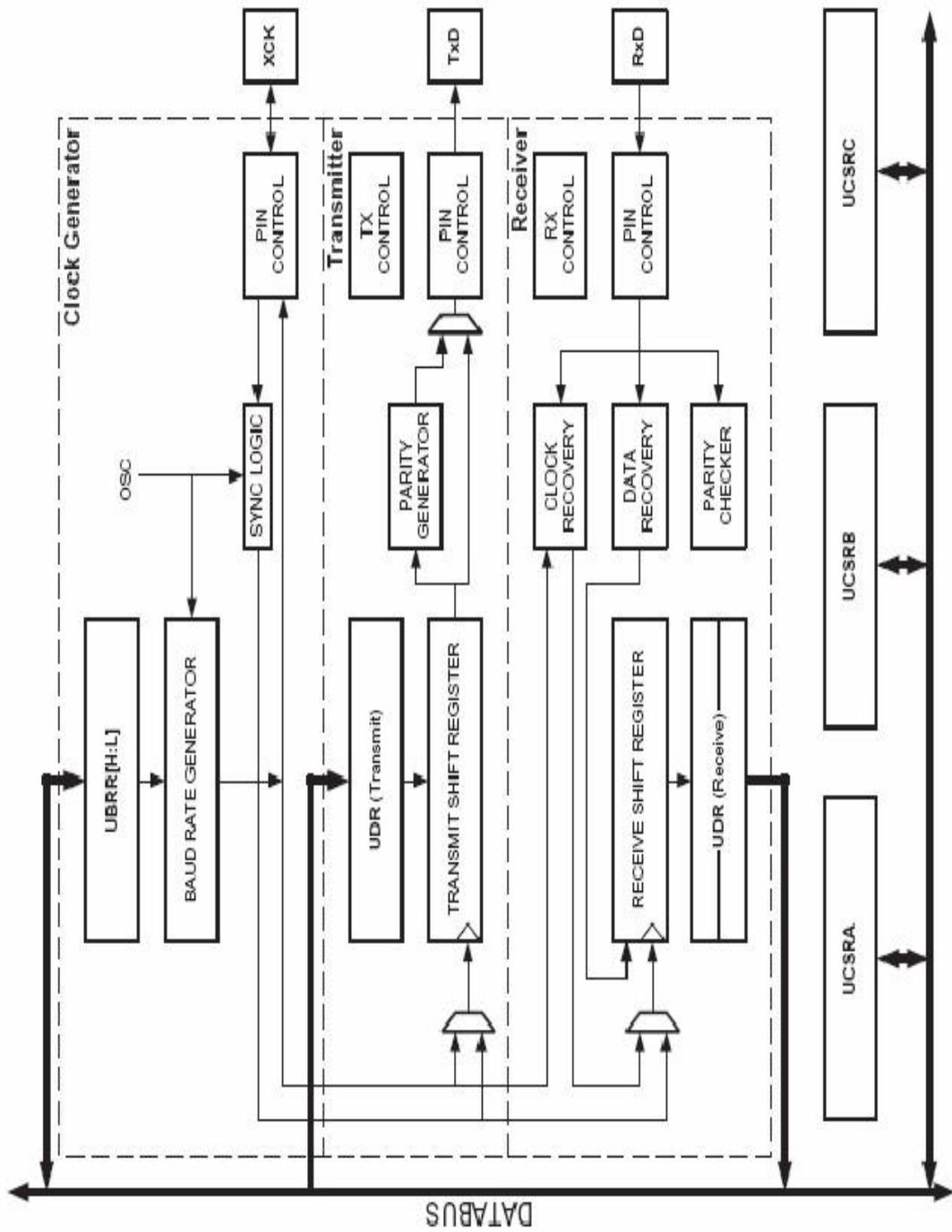


Motor control kit

La tarjeta LPCXpresso es la encargada de recibir las instrucciones enviadas por la LPCXpresso 1769 en las salidas de esta. Las instrucciones son procesadas y enviadas a la tarjeta de desarrollo que es la encargada de realizar el control por voltaje del motor BLDC según sea la instrucción enviada por el usuario. Para mayor información revise la página 89 donde se detalla el procedimiento del control del motor BLDC.

# **Anexo V**

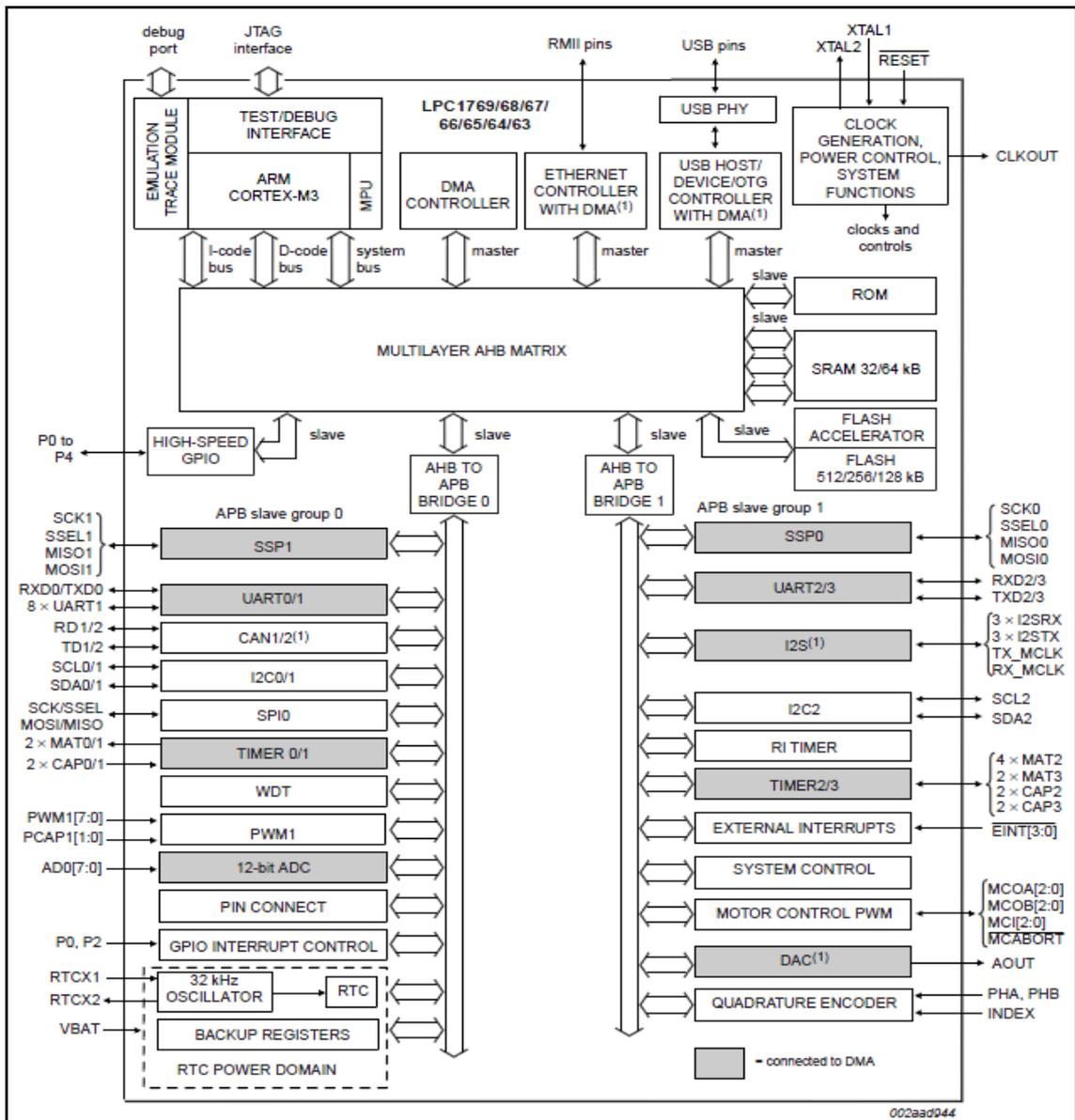
**Diagrama de bloques de la UART en la  
tarjeta AVR Butterfly**



# **Anexo VI**

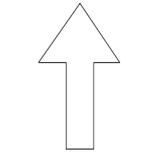
**Diagrama de bloques de la tarjeta**

**LPCXpresso 1769**



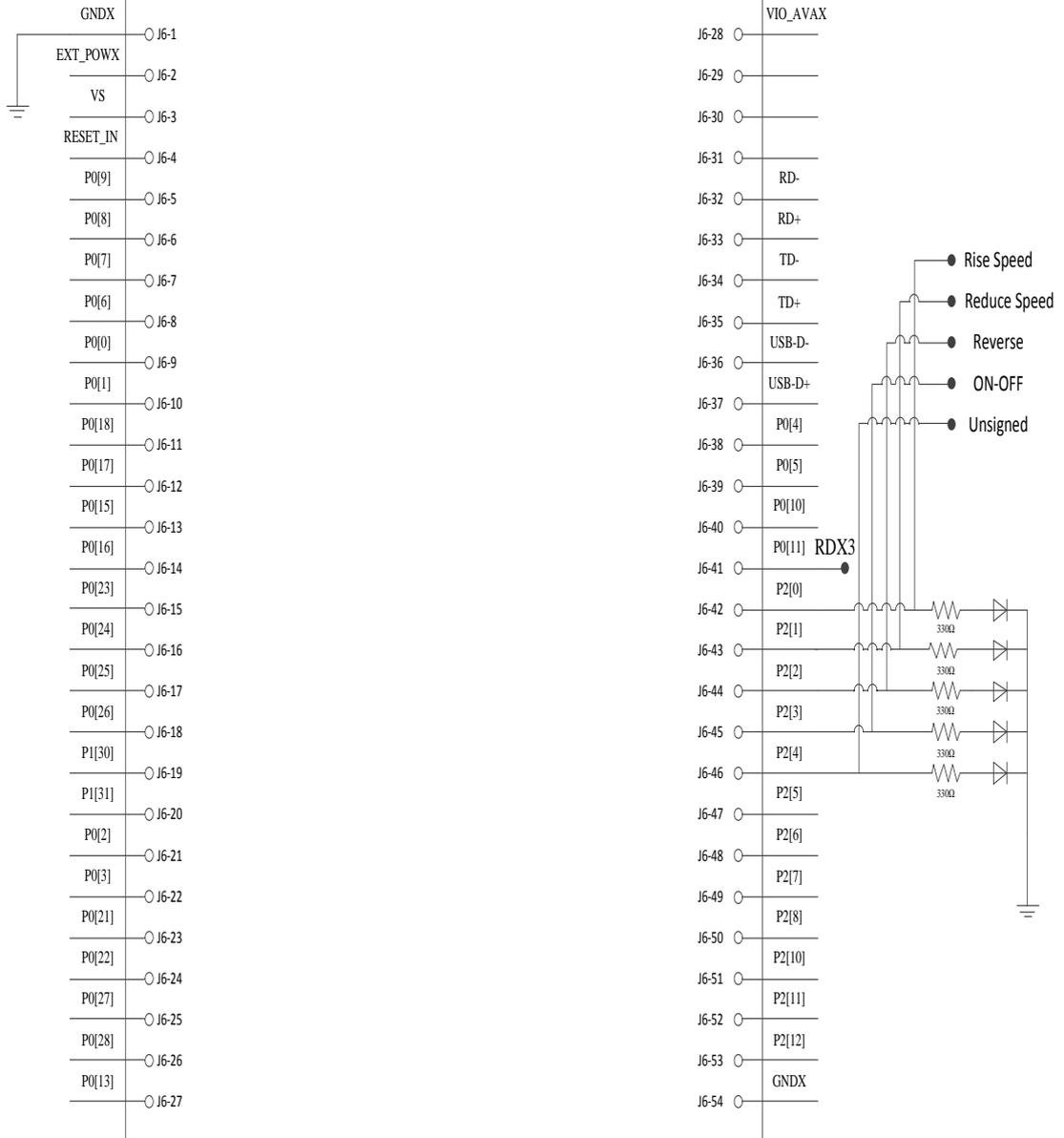
# **Anexo VII**

**Diagrama eléctrico de las conexiones  
en la LPCXpresso**



LPC-LINK SIDE

Expansion Connector  
(superset of mbed pinng)



## BIBLIOGRAFÍA

- [1] ATMEL, AVR Butterfly Evaluation Kit, User Guide. [www.atmel.com/literature](http://www.atmel.com/literature); Fecha de consulta Febrero 2012.
- [2] ATMEL, AVR Butterfly Quick Start, User Guide. [www.atmel.com/products/AVR/butterfly](http://www.atmel.com/products/AVR/butterfly); Fecha de consulta Febrero 2012.
- [3] ATMEL, 8-bit AVR Microcontroller with 16k Bytes In-System Programmable Flash. [www.atmel.com/literature](http://www.atmel.com/literature); Fecha de consulta Marzo 2012.
- [4] EMBEDDED ARTIST, LPCXpresso Motor Control kit. User Guide. <http://ics.nxp.com/support/microcontrollers/motor.control/>; Fecha de consulta Marzo 2012.
- [5] HOPE MICROELECTRONICS CO., LTD HM-TR Transparent Wireless Data Link Module. User Guide. [www.hoperf.com](http://www.hoperf.com); Fecha de consulta Marzo de 2012.
- [6] Kharsansky Alan, Introduccion al LPCXpresso y repaso del lenguaje C. Seminario de Sistemas Embebidos. [http://laboratorios.fi.uba.ar/lse/seminario/material-2011/Sistemas Embebidos 2doC-Intro a LPCXpresso y repaso lenguaje C-Kharsansky.pdf](http://laboratorios.fi.uba.ar/lse/seminario/material-2011/Sistemas_Embebidos_2doC-Intro_a_LPCXpresso_y_repaso_lenguaje_C-Kharsansky.pdf); Fecha de consulta Marzo de 2012.
- [7] Marcela Rudas, Julian Higuera. Modulador y Desmodulador FSK. [www.miteleprocesos.tripod.com/monitoreo\\_archivos/FSK.pdf](http://www.miteleprocesos.tripod.com/monitoreo_archivos/FSK.pdf); Fecha de consulta Marzo 2012.

- [8] MLX90287 12V PWM Fan Driver, Melexis, [www.melexis.com/Hall-Effect-Sensor-ICs/Hall-Effect-Fan-Motor-Drivers/MLX90287-739.aspx](http://www.melexis.com/Hall-Effect-Sensor-ICs/Hall-Effect-Fan-Motor-Drivers/MLX90287-739.aspx); Fecha de consulta Marzo 2012.
- [9] MLX80151 High current NFET half bridge motorcontroller, Melexis, [www.melexis.com/DC-Motor-Control-ICs/DC-Motor-Control-ICs/MLX80151-712.aspx](http://www.melexis.com/DC-Motor-Control-ICs/DC-Motor-Control-ICs/MLX80151-712.aspx); Fecha de consulta Marzo 2012.
- [10] MOTOR BRUSHLESS (sin escobillas), Master Ingenieros S.A, [www.masteringenieros.com/master/Ficheros/File/motor.pdf](http://www.masteringenieros.com/master/Ficheros/File/motor.pdf); Fecha de consulta Marzo 2012.
- [11] MOTOR BRUSHLESS (sin escobillas), Master Ingenieros S.A, [www.masteringenieros.com/master/Ficheros/File/motor.pdf](http://www.masteringenieros.com/master/Ficheros/File/motor.pdf); Fecha de consulta Marzo 2012.
- [12] Motores brushless o BLDC, Mario Sacco, [www.neoteo.com/motores-brushless-blcd](http://www.neoteo.com/motores-brushless-blcd); Fecha de consulta Febrero 2012.
- [13] AVR Simulation with the ATMEL AVR Studio 4, [http://www2.tech.purdue.edu/ecet/courses/ecet309/Reference\\_Materials/Simulation\\_AVR\\_Studio\\_4.pdf](http://www2.tech.purdue.edu/ecet/courses/ecet309/Reference_Materials/Simulation_AVR_Studio_4.pdf); Fecha de consulta Marzo 2012.
- [14] Tutorial LPCXpresso IDE CodeRed. Seminario de Sistemas Embebidos. [www.laboratorios.fi.uba.ar/lse/seminario/material\\_2011/Sistemas\\_Embebidos-2011\\_2doC-Mini\\_Tutorial\\_CodeRed\\_LPCXpresso\\_IDE\\_Kharsansky.pdf](http://www.laboratorios.fi.uba.ar/lse/seminario/material_2011/Sistemas_Embebidos-2011_2doC-Mini_Tutorial_CodeRed_LPCXpresso_IDE_Kharsansky.pdf); Fecha de consulta Febrero del 2012.

- [15] Universidad de la Am3ricas Puebla. Propagaci3n RF, Capitulo 1. [www.catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lem/.../capitulo1.pdf](http://www.catarina.udlap.mx/u_dl_a/tales/documentos/lem/.../capitulo1.pdf); Fecha de consulta Marzo 2012.
- [16] UM10360. LPCXpresso User Manual. [www.npx.com](http://www.npx.com); Fecha de consulta Febrero del 2012.