



**ESCUELA SUPERIOR POLITÉCNICA DEL CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA ELECTRÓNICA**

**“DISEÑO Y CONSTRUCCIÓN DE UN ROBOT DE BATALLA**  
**CONTROLADO MEDIANTE DISPOSITIVO BLUETOOTH”**

TESIS DE GRADO

Previa la obtención del Título de:

**INGENIERO EN ELECTRÓNICA Y COMPUTACIÓN**

Presentado por:

**ANDERSON XAVIER TORRES GARCÉS**

**EDGAR RICARDO LOZANO GRANDA**

Riobamba – Ecuador

2010

Expresamos nuestro agradecimiento a la Escuela Superior Politécnica del Chimborazo, a la Facultad de Informática y Electrónica, a la Escuela de Ingeniería Electrónica por habernos dado la oportunidad de optar por una carrera terminal, la cual nos permitirá seguir superándonos personal y profesionalmente.

De igual manera al Ing. Wilson Zúñiga, Ing. Paúl Romero, Ing. Ángel Cruz; quienes colaboraron como asesores en este trabajo brindándonos su apoyo incondicional cuando se ha requerido.

Queremos hacer extensivo un sentimiento de gratitud a nuestras respectivas familias, por ser quienes nos han alentado para seguir adelante durante todo el acontecer académico.

A NUESTRAS FAMILIAS:

Por su inigualable amor,  
comprensión y sacrificio, que con  
ternura perdurable han sido el pilar  
fundamental en nuestras vidas.

**NOMBRE**

**FIRMA**

**FECHA**

Dr. Romeo Rodríguez

.....

.....

**DECANO FACULTAD INFORMÁTICA  
Y ELECTRÓNICA**

Ing. Paúl Romero

.....

.....

**DIRECTOR ESCUELA  
INGENIERÍA ELECTRÓNICA**

Ing. Paúl Romero

.....

.....

**DIRECTOR DE TESIS**

Ing. Wilson Zúñiga

.....

.....

**MIEMBRO DE TESIS**

Lcdo. Carlos Rodríguez

.....

.....

**DIRECTOR CENTRO DOCUMENTACIÓN**

NOTA: .....

Nosotros, ANDERSON XAVIER TORRES GARCÉS Y EDGAR RICARDO LOZANO GRANDA, somos responsables de las ideas, doctrinas y resultados expuestos en esta Tesis, y el patrimonio intelectual de la Tesis de Grado pertenecen a la Escuela Superior Politécnica del Chimborazo.

## ÍNDICE DE ABREVIATURAS

<b>C</b>	Lenguaje de programación de propósito general asociado, de modo universal, al sistema operativo UNIX.
<b>C++</b>	Versión de C orientado a objetos.
<b>JVM</b>	Máquina virtual JAVA
<b>JDK</b>	Java Development Kit
<b>API</b>	Interfaz de Programación de Aplicaciones
<b>HTML</b>	Lenguaje de marcas de Hipertexto
<b>J2ME</b>	Java 2 Platform, Micro Edition
<b>MIDlets</b>	Es un programa en lenguaje de programación Java para dispositivos embebidos.
<b>POO</b>	Programación orientada a objetos
<b>JRE</b>	Entorno en tiempo de ejecución JAVA
<b>J2SE</b>	Java 2, Standard Edition
<b>J2EE</b>	Java Platform, Enterprise Edition
<b>XML</b>	Metalenguaje de definición de documentos estructurados mediante marcas o etiquetas
<b>CDC</b>	Connected Device Configuration
<b>CLDC</b>	Connected Limited Device Configuration
<b>PDA</b>	Asistente personal digital
<b>KVM</b>	Kernel-based Virtual Machine
<b>MIDP</b>	Mobile Information Device Profile
<b>CPU</b>	Unidad central de procesamiento
<b>MB</b>	Megabyte
<b>WAV</b>	WAVEform audio format
<b>MID</b>	Mobile Information Device
<b>OEM</b>	Original Equipment Manufacturer
<b>HTTP</b>	HyperText Transfer Protocol
<b>HTTPS</b>	HyperText Transfer Protocol Secure
<b>SMS</b>	Short Message Service
<b>CBS</b>	Cell Broadcast Service
<b>GSM</b>	Sistema Global para las Comunicaciones Móviles
<b>CDMA</b>	Acceso múltiple por división de código
<b>WMA</b>	Wireless Messaging API
<b>JSR</b>	Java Community Process
<b>QWERTY</b>	Teclado común
<b>RAM</b>	Memoria de acceso aleatorio
<b>TCP/IP</b>	Protocolo de Control de Transmisión/Protocolo Internet
<b>WAP</b>	Protocolo de aplicaciones inalámbricas
<b>IDE</b>	Controlador Electrónico Integrado
<b>EJB</b>	Enterprise JavaBeans
<b>WPAN</b>	Red Inalámbrica de Área Personal
<b>ISM</b>	International Safety Management
<b>mW</b>	Milivatio
<b>dBm</b>	Decibelios en relación a un nivel de referencia de 1 mW.
<b>LAN</b>	Red de área local
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>WIFI</b>	Tecnología de comunicación inalámbrica de datos
<b>PAN</b>	Red de Área Personal

<b>TDM</b>	Time Division Multiplex
<b>GHz</b>	Giga hertz
<b>TX</b>	Transmisor
<b>RX</b>	Receptor
<b>USB</b>	Universal Serial Bus
<b>FCC</b>	Federal Communication Commission
<b>GFSK</b>	Modulación por desplazamiento de frecuencia gaussiana
<b>MHz</b>	Mega hertz
<b>FSK</b>	Modulación por desplazamiento de frecuencia
<b>RISC</b>	Procesador con un Conjunto Reducido de Instrucciones
<b>EEPROM</b>	Electrically-Erasable Programmable Read-Only Memory
<b>I/O</b>	Entrada/Salida
<b>PWM</b>	Modulación por ancho de pulsos
<b>GPR</b>	Pic de uso general
<b>SFR</b>	Pic de funciones especiales
<b>RA0-RA7</b>	Puertos bidireccionales
<b>TTL</b>	Tiempo de vida
<b>Vpp</b>	Valor pico pico
<b>TMR0</b>	Contador del pic
<b>VDD</b>	Pin de alimentación positiva
<b>VSS</b>	Pin de alimentación negativa
<b>MCLR</b>	Master clear (Reset)
<b>Vcc</b>	Voltaje de corriente continua
<b>OSC1/CLKIN</b>	Entrada de oscilador externo
<b>OSC2/CLKOUT</b>	Salida del oscilador
<b>CC</b>	Corriente continua
<b>F</b>	Fuerza en newtons
<b>B</b>	Inducción en teslas
<b>CD</b>	Corriente directa
<b>ISIS</b>	Intelligent Schematic Input System
<b>CAD</b>	Computer Assisted Draw
<b>CMOS</b>	Complementary Metal Oxide Semiconductor
<b>PLD</b>	Dispositivo lógico programable
<b>IC</b>	Circuito Integrado
<b>ASCII</b>	American Standard Code for Information Interchange
<b>PCB</b>	Printed Circuit Board
<b>HEX</b>	Hexadecimal
<b>Bas</b>	Extensión de guardado de PicBasic
<b>PC</b>	Computadora personal
<b>ICD</b>	Circuito depurador
<b>RS232</b>	Electronic Industries Alliance RS-232C
<b>OSC</b>	Oscilador
<b>Com</b>	Puerto de comunicación
<b>JDM</b>	Joint design Manufacturer

## ÍNDICE DE TABLAS

Tabla I.I.- Clases de Bluetooth.....	32
Tabla I.II.- Resumen de acciones del Puente H.....	46
Tabla III.III.- Tabla de verdad.....	93
Tabla III.IV.- Consumo de corriente de motor de CC.....	94
Tabla III.V.- Tiempo de funcionamiento de un robot.....	94
Tabla III.VI.- Consumo de corriente de un motor de arranque.....	95
Tabla III.VII.- Consumo de corriente del rodillo.....	95
Tabla III.VIII.- Consumo de corriente y voltaje del Bluetooth .....	96

## ÍNDICE DE FIGURAS

Fig. I.01.- Plataforma Java, Versiones y Aplicaciones.....	24
Fig. I.02.- Arquitectura J2ME.....	25
Fig. I.03.- Arquitectura MIDP.....	27
Fig. I.04.- Ciclo de vida de un MIDlet.....	30
Fig. I.05.- Scatternet o Red Dispersa.....	33
Fig. I.06.- Bluetooth BLUESMIRF-RN-V1.....	36
Fig. I.07.- Disposición de pines modulo BLUESMIRF-RN-V1.....	36
Fig. I.08.- Filtro gaussiano para datos.....	37
Fig. I.09.- PIC 16f628A.....	39
Fig. I.10.- Estructura interna de un cilindro eléctrico.....	41
Fig. I.11.- Estructura de un Motor de Corriente Continua.....	43
Fig. I.12.- Motor de corriente continua.....	43
Fig. I.13.- Motor de arranque.....	44
Fig. I.14.- Estructura básica de un Puente H.....	45
Fig. I.15.- Estados de un Puente H.....	46
Fig. I.16.- Esquema de ISIS.....	50
Fig. I.17.- Encapsulado de ISIS.....	51
Fig. I.18.- Buses de ISIS.....	52
Fig. I.19.- Diagrama de componentes de ISIS.....	53
Fig. I.20.- Entorno ARES.....	53
Fig. I.21.- Posicionamiento manual de selección de componentes.....	55
Fig. I.22.- Ruteo en ARES.....	55
Fig. I.23.- Ruteo manual.....	56
Fig. I.24.- Comunicación PIC-PC.....	58
Fig. I.25.- Configuración de WinPic800.....	60
Fig. II.26.- Vista inferior de la estructura del Robot de Batalla.....	63
Fig. II.27.- Vista superior de la estructura del Robot de Batalla.....	65
Fig. II.28.- Vista Lateral de la estructura del Robot de Batalla.....	65
Fig. II.29.-Estructura armada del Robot de Batalla en 3D.....	66
Fig. II.30.- Motor de CC con caja de engranaje del robot.....	67
Fig. II.31- Disposición de los Motores dentro de la Estructura.....	67
Fig. II.32.- Diseño en 3D del robot de Batalla con sus ruedas.....	68
Fig. II.33.- Motor de Arranque.....	69

Fig. II.34.- Diseño de la Pala Eléctrica .....	70
Fig. II.35.- Diseño Del Rodillo Eléctrico.....	71
Fig. II.36.- Estructura del Robot de Batalla.....	72
Fig. II.37.- Diseño de la etapa de Control de Potencia .....	73
Fig. II.38.- Diseño de la fuente para el Pic 16F628A .....	74
Fig. II.39.- Ubicación de Optoacopladores a la entrada del Pic 16F628A .....	75
Fig. II.40.- Ubicación de Optoacopladores a la salida del Pic 16F628A .....	76
Fig. II.41.- Sub-Etapa de Activación de la Etapa de Potencia.....	77
Fig. II.42.- Puente H .....	78
Fig. II.43- Protección con diodos del Puente H.....	79
Fig. II.44.- Circuito de control de la Pala.....	80
Fig. II.45.- Circuito de control del Rodillo.....	81
Fig. II.46.- Diagrama de flujo de funcionamiento del PIC 16F628A .....	82
Fig. II.47.- Placa de control de potencia.....	87
Fig. II.48.- Diseño del circuito de potencia.....	87
Fig. II.49.- Placa de potencia.....	88
Fig. II.50.- Circuito de la etapa de Control de las armas.....	88
Fig. II.51.- Diseño del circuito de control de las armas.....	89
Fig. II.52.- Placa de control de armas.....	89
Fig. III.53.- Circuito del modulo Bluetooth en ISIS.....	90
Fig. III.54.- Circuito del modulo Bluetooth en ISIS.....	91

## ÍNDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

FIRMAS RESPONSABLES Y NOTA

RESPONSABILIDAD DE LOS AUTORES

INDICE DE ABREVIATURAS

INDICE DE TABLAS

INDICE DE FIGURAS

INDICE GENERAL

INTRODUCCIÓN

### **CAPÍTULO I**

FUNDAMENTOS TEÓRICOS.....	18
1. El Lenguaje de Programación JAVA.....	18
1.1. Reseña Histórica.....	18
1.2. Principales Características.....	20
1.3. Aplicaciones de JAVA.....	20
1.4. Programación Orientada a Objetos POO.....	21
1.4.1. Propiedades de una POO.....	22
1.5. TECNOLOGIAS J2ME.....	23
1.5.1. Introducción.....	23
1.5.2. Arquitectura de J2ME.....	24
1.5.3. Configuraciones J2ME.....	25
1.5.4. Perfiles de J2ME.....	26
1.6. SOFTWARE DE DESARROLLO.....	30
1.6.1. NetBeans.....	30
1.7. TECNOLOGIA BLUETOOTH.....	32
1.7.1. Definición.....	32
1.7.2. Topología de las redes Bluetooth.....	33
1.7.3. Principio de funcionamiento.....	34
1.8. MODULO BLUETOOTH BLUESNIRF-RN-V1.....	35
1.8.1. Descripción.....	35
1.8.2. Características.....	35
1.8.3. Aplicaciones.....	35

1.9. MODULACION GFSK.....	36
1.10. MICROCONTROLADORES.....	37
1.10.1. Microcontrolador PIC 16F628A.....	37
1.10.2. Características.....	38
1.10.3. Pines de I/O (Entrada/Salida).....	39
1.11. CILINDRO ELÉCTRICO.....	40
1.11.1. Descripción.....	40
1.11.2. Principios de funcionamiento.....	40
1.11.3. Ventajas.....	41
1.12. MOTORES ELÉCTRICOS.....	42
1.12.1. Descripción.....	42
1.12.2. Principio de funcionamiento.....	42
1.13. MOTORES DE ARRANQUE.....	43
1.13.1. Descripción.....	43
1.13.2 Partes de un motor de arranque.....	44
1.13.3. Características.....	44
1.13.4. Ventajas.....	45
1.14. PUENTE H.....	45
1.14.1 Descripción.....	45
1.14.2. Aplicaciones.....	46
1.14.3. Montaje.....	47
1.14.4. Características de la interfaz Puente H.....	47
1.15. PROTEUS.....	47
1.15.1. Descripción.....	47
1.15.2. Aspectos básicos del programa Proteus.....	48
1.15.3. Características de Proteus.....	48
1.15.4. Elementos de Proteus.....	49
1.16. ISIS.....	49
1.16.1. Introducción.....	49
1.16.2. Herramienta gráfica para la gestión de los encapsulados.....	50
1.16.3. Completa gestión de diseños jerárquicos.....	51
1.16.4. Utilización de buses.....	52
1.16.5. Administrador de propiedades de los componentes.....	52
1.17. ARES.....	53
1.17.1. Entorno de trabajo de Ares.....	54

1.17.2. Herramientas.....	54
1.18. MICROCODE STUDIO.....	56
1.18.1. Introducción.....	56
1.18.2. Características.....	57
1.18.3. In Circuit Debugger (ICD) .....	57
1.18.4. PIC que Soporta Microcode Studio.....	57
1.18.5. Configuración de hardware.....	58
1.19. WinPic800.....	59
1.19.1. Introducción.....	59
1.19.2. Configuración.....	59
1.19.3. Grabar PIC con WinPic800.....	60
<b>CAPÍTULO II</b>	
<b>DISEÑO Y CONSTRUCCIÓN DEL ROBOT DE BATALLA.....</b>	<b>61</b>
2.1. DISEÑO.....	61
2.1.1. Diseño inferior de la estructura del robot.....	63
2.1.2. Diseño superior de la Estructura.....	64
2.1.3. Diseño lateral de la Estructura.....	65
2.1.4. Modelo final de la Estructura.....	66
2.1.5. Diseño de Movimiento.....	66
2.1.6. Diseño en 3D de la estructura.....	68
2.1.7. Diseño Mecánico de Armas.....	68
2.1.7.1. Pala Eléctrica.....	69
2.1.7.2. Rodillo Eléctrico .....	70
2.1.8. Diseño Eléctrico-Electrónico .....	71
2.1.8.1. Diseño de la etapa de Control de Potencia .....	72
2.1.8.2. Diseño de la etapa de Potencia.....	77
2.1.8.3. Diseño de la etapa de Control de Armas.....	80
2.1.8.4. Pala .....	80
2.1.8.5. Rodillo.....	81
2.2. Diagrama de flujo de operación del PIC 16F628A.....	82
2.3. Programa para control del PIC 16F628A.....	83
2.4. Implementación de la etapa de potencia.....	87
2.5. Implementación de la etapa de control de armas.....	88
2.6. Diseño del circuito de transmisión del modulo Bluetooth.....	90

## **CAPÍTULO III**

PRUEBAS Y RESULTADOS.....	92
3.1. Etapa de control.....	92
3.2. Etapa del Puente H.....	93
3.3. Circuitos de armas.....	94
3.3.1. Pala.....	95
3.3.2. Rodillo.....	95
3.4. Bluetooth.....	96

## **CONCLUSIONES Y RECOMENDACIONES**

### **RESUMEN**

### **SUMMARY**

### **BIBLIOGRAFÍA**

### **ANEXOS**

## **INTRODUCCIÓN**

El uso de robots industriales y comerciales junto con los sistemas de diseño asistidos por computadora (CAD), y los sistemas de fabricación asistidos por computadora (CAM), son la última tendencia.

La construcción de robots de batalla ha permitido a los estudiantes de la escuela de Ingeniería Electrónica participar en los diferentes eventos de robótica a nivel nacional e internacional, obteniendo logros importantes que dejan en alto el prestigio de nosotros como estudiantes y de la ESPOCH como institución.

Un robot de batalla tiene como objetivo primordial vencer a cada uno de los contrincantes que se le presenten en el ring de lucha. Las opciones de mando que controlaran al robot se realizarán mediante transmisión de radiofrecuencia utilizando tecnología Bluetooth.

El robot de batalla manipula una serie de motores, un cilindro eléctrico los cuales permitirán realizar el respectivo movimiento y el control del sistema de arma, dichas instrucciones se las recibirán y transmitirán en el dispositivo Bluetooth las mismas que serán comparadas en un circuito electrónico permitirán ejecutar las órdenes deseadas desde un teléfono celular.

El dispositivo Bluetooth posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia segura y globalmente libre a una frecuencia de 2,4 a 2,8 GHz. Los principales objetivos que se pretenden conseguir con la implementación del dispositivo es conseguir una transmisión más fiable, por utilizar un total de 79 frecuencias con intervalos de un MHz, lo cual nos permite una mayor seguridad y robustez y de esta manera no tener interferencias con otros robots de batalla, además de facilitar la sincronización con las etapas eléctrico-electrónicas del robot.

El avance de las Telecomunicaciones nos permite la utilización del protocolo de banda base (canales simples por vía) el cual combina la conmutación de circuitos y de paquetes para asegurar que los paquetes no lleguen fuera de orden.

## **JUSTIFICACIÓN**

La Robótica es una ciencia que nos permite el desarrollar aplicaciones mediante la fusión de diferentes tecnologías como la electrónica, mecánica, telecomunicaciones entre otras.

El proyecto planteado tiene como fin el desarrollo de un robot de batalla controlado mediante un dispositivo Bluetooth, el cual nos permitirá poder realizar la implementación de las diferentes tecnologías para su aplicación en el área de la robótica, y nos servirá como base de introducción en el campo de la mecatrónica.

Mediante el desarrollo del robot de batalla nos podemos dar cuenta como avanza día a día el mundo de la tecnología con lo que respecta a la programación, electrónica, telecomunicaciones, mecánica y la robótica, los mismos que combinan sus diferentes etapas para brindar un correcto funcionamiento y para esto nos valemos de los conceptos y laboratorios aprendidos en el transcurso de la carrera así como los recursos disponibles en Internet.

Mediante la realización del proyecto planteado se pretende lograr un óptimo desempeño en los diferentes concursos de robótica tanto a nivel nacional como internacional, además de servirles como base para futuras implementaciones de proyectos similares y afines a este.

Con la utilización del dispositivo Bluetooth podremos reducir los problemas relacionados con la interferencia de las telecomunicaciones obteniendo una mayor eficiencia y fidelidad del robot de batalla.

Con esto se fomenta la creación de nuevos grupos que se sientan identificados con el desarrollo de aplicaciones similares en el campo de la robótica, además de la realización de investigaciones de nuevas aplicaciones y su presentación en los diversos concursos de robótica.

Además este es un proyecto investigativo ya que nos ayuda a vincularnos y a conocer un poco más a fondo las diferentes tecnologías a utilizar.

## **HIPÓTESIS**

Con la implementación de este proyecto se pretende desarrollar una herramienta de investigación con ramas afines a la robótica como mecánica, electricidad, informática y otros a fines.

# CAPÍTULO I

## FUNDAMENTOS TEÓRICOS

### 1. El Lenguaje de Programación JAVA

#### 1.1. Reseña Histórica

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

La tecnología Java se creó como una herramienta de programación para ser usada en un proyecto de set-top-box en una pequeña operación denominada *the Green Project* en Sun Microsystems en el año 1991. El equipo (*Green Team*), compuesto por trece personas y dirigido por James Gosling, trabajó durante 18 meses en Sand Hill Road en Menlo Park en su desarrollo.

El lenguaje se denominó inicialmente *Oak* (por un roble que había fuera de la oficina de Gosling), luego pasó a denominarse *Green* tras descubrir que *Oak* era ya una marca

comercial registrada para adaptadores de tarjetas gráficas y finalmente se renombró a *Java*.

Los objetivos de Gosling eran implementar una máquina virtual y un lenguaje con una estructura y sintaxis similar a C++. Entre junio y julio de 1994, tras una sesión maratónica de tres días entre John Gage, James Gosling, Joy Naughton, Wayne Rosing y Eric Schmidt, el equipo reorientó la plataforma hacia la Web. Sintieron que la llegada del navegador web Mosaic, propiciaría que Internet se convirtiese en un medio interactivo, como el que pensaban era la televisión por cable. Naughton creó entonces un prototipo de navegador, WebRunner, que más tarde sería conocido como HotJava.

En 1994, se les hizo una demostración de HotJava y la plataforma Java a los ejecutivos de Sun. Java 1.0a pudo descargarse por primera vez en 1994, pero hubo que esperar al 23 de mayo de 1995, durante las conferencias de SunWorld, a que vieran la luz pública Java y HotJava, el navegador Web. El acontecimiento fue anunciado por John Gage, el Director Científico de Sun Microsystems. El acto estuvo acompañado por una pequeña sorpresa adicional, el anuncio por parte de Marc Andreessen, Vicepresidente Ejecutivo de Netscape, que Java sería soportado en sus navegadores. El 9 de enero del año siguiente, 1996, Sun fundó el grupo empresarial JavaSoft para que se encargase del desarrollo tecnológico. Dos semanas más tarde la primera versión de Java fue publicada.

La promesa inicial de Gosling era *Write Once, Run Anywhere* (Escríbelo una vez, ejecútalo en cualquier lugar), proporcionando un lenguaje independiente de la plataforma y un entorno de ejecución (la JVM) ligero y gratuito para las plataformas más populares de forma que los binarios (bytecode) de las aplicaciones Java pudiesen ejecutarse en cualquier plataforma.

El entorno de ejecución era relativamente seguro y los principales navegadores web pronto incorporaron la posibilidad de ejecutar applets Java incrustadas en las páginas web.

Java ha experimentado numerosos cambios desde la versión primigenia, JDK 1.0, así como un enorme incremento en el número de clases y paquetes que componen la biblioteca estándar.

Desde entonces, SUN ha sabido manejar inteligentemente el éxito obtenido por su lenguaje, concediéndose licencias a cualquiera sin ningún problema, fomentando su uso entre la comunidad informática y extendiendo las especificaciones y funcionalidad del lenguaje.

Es un lenguaje de desarrollo de propósito general, válido para realizar todo tipo de aplicaciones profesionales.

## 1.2. Principales Características

- Lenguaje simple: Java posee una curva de aprendizaje muy rápida. Resulta relativamente sencillo escribir aplicaciones interesantes desde el principio. Todos aquellos familiarizados con C++ encontrarán que Java es más sencillo, ya que se han eliminado ciertas características, como los punteros.
- Es un lenguaje **orientado a objetos**, eso implica que su concepción es muy próxima a la forma de pensar humana.
- Es un lenguaje que interpreta los ficheros y clases en una máquina virtual de java.
- Es un lenguaje **multiplataforma**: El mismo código Java que funciona en un sistema operativo, lo ara en cualquier otro que tenga instalada la máquina virtual java.
- Es un lenguaje seguro: La máquina virtual, al ejecutar el código Java, realiza comprobaciones de seguridad.
- Gracias al API de Java podemos ampliar el lenguaje para que sea capaz de, por ejemplo, comunicarse con equipos mediante red, acceder a bases de datos, crear páginas HTML dinámicas, crear aplicaciones visuales al estilo Windows.
- Es fácil de aprender y está bien estructurado.

## 1.3. Aplicaciones de Java

El diseño de Java, su robustez, el respaldo de la industria y su fácil portabilidad han hecho de Java uno de los lenguajes con un mayor crecimiento y amplitud de uso en distintos ámbitos de la industria de la informática.

## **En dispositivos móviles y sistemas empotrados**

Desde la creación de la especificación J2ME (Java 2 Platform, Micro Edition), una versión del entorno de ejecución Java reducido y altamente optimizado, especialmente desarrollado para el mercado de dispositivos electrónicos de consumo se ha producido toda una revolución en lo que a la extensión de Java se refiere.

Es posible encontrar microprocesadores específicamente diseñados para ejecutar bytecode Java y software Java para tarjetas inteligentes (JavaCard), teléfonos móviles, buscapersonas, set-top-boxes, sintonizadores de televisión y otros pequeños electrodomésticos.

El modelo de desarrollo de estas aplicaciones es muy semejante a las *applets* de los navegadores salvo que en este caso se denominan MIDlets.

### **1.4. Programación Orientada a Objetos POO**

La programación orientada a objetos o **POO** (**OOP** según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora, estos objetos colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener, y reutilizar. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento. De aquella forma, un objeto contiene toda la información que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases e incluso frente a objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos. A su vez, los objetos disponen de mecanismos de interacción llamados métodos que favorecen la comunicación entre ellos.

Los métodos (comportamiento) y atributos (estado) están estrechamente relacionados por la propiedad de conjunto. Esta propiedad destaca que una clase requiere de métodos para poder tratar los atributos con los que cuenta. El programador debe pensar indistintamente en ambos conceptos, sin separar ni darle mayor importancia a ninguno de ellos. Hacerlo podría producir el hábito erróneo de crear clases contenedoras de información por un lado y clases con métodos que manejen a las primeras por el otro.

De esta manera se estaría realizando una programación estructurada camuflada en un lenguaje de programación orientado a objetos.

Clase y objeto están íntimamente ligados en la POO donde su diferencia radica que al declarar un objeto hay que definir el tipo de objeto al que pertenece siendo este tipo la clase.

#### **1.4.1. Propiedades de la POO**

Para ser considerado un lenguaje orientado a objetos debe cumplir con tres propiedades fundamentales:

##### **Encapsulamiento**

El encapsulamiento permite a los objetos elegir qué información es publicada y qué información es ocultada al resto de los objetos. Para ello los objetos suelen presentar sus métodos como interfaces públicas y sus atributos como datos privados e inaccesibles desde otros objetos.

Para permitir que otros objetos consulten o modifiquen los atributos de los objetos, las clases suelen presentar métodos de acceso. Con el encapsulado de los datos se consigue que las personas que utilicen un objeto sólo tengan que comprender su interfaz, olvidándose de cómo está implementada, y en definitiva, reduciendo la complejidad de utilización ya que esta información solo la maneja el programador.

##### **Herencia**

Es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en C. Los componentes registrados como "privados" (private) también se heredan, pero como no pertenecen a la clase, se mantienen escondidos al programador y sólo pueden ser accedidos a través de otros métodos públicos. Esto es así para poder mantener hegemónico el ideal de POO. Esta propiedad permite la reutilización del código, siendo muy fácil aprovechar el código de clases ya existentes, modificándolas mínimamente para adaptarlas a las nuevas especificaciones.

## **Polimorfismo**

Las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama *asignación tardía* o *asignación dinámica*. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.

### **1.5. Tecnologías J2ME**

#### **1.5.1. Introducción**

Las tecnologías **J2ME** contienen un JRE altamente optimizado, especialmente desarrollado para el mercado de gran consumo, abarcan una amplia gama de aparatos de tamaño muy reducido y permiten ejecutar programas de seguridad, conectividad y utilidades en tarjetas inteligentes, buscapersonas, sintonizadores de TV y otros pequeños electrodomésticos. Las tecnologías J2ME representan únicamente una parte de la gama de productos de software de Java. Las plataformas Java relacionadas son la Plataforma Java 2, Edición estándar (plataforma J2SE) y la Plataforma Java 2, Edición empresa (plataforma J2EE). La tecnología Java ofrece, asimismo, métodos de creación de servicios Web, transferencia de información XML, numerosos protocolos de red, kits de herramientas y la aplicación Java Web Start.

La figura I.01 muestra la plataforma Java y las diferentes versiones que se han desarrollado y sus aplicaciones.

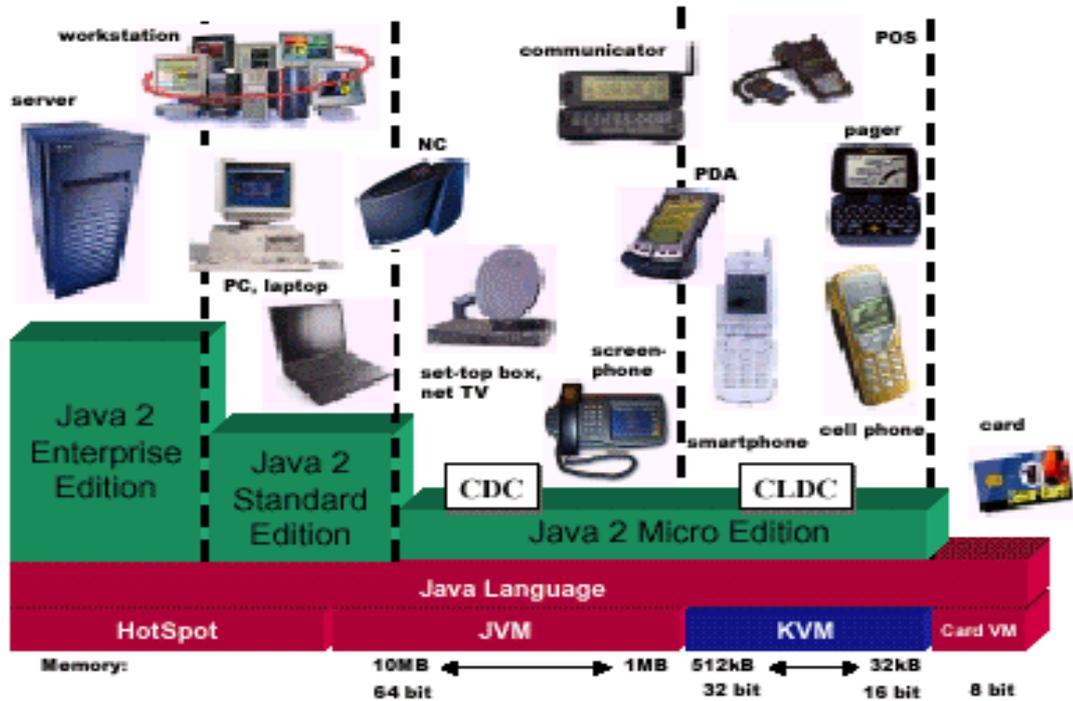


Fig. I.01.- Plataforma Java, Versiones y Aplicaciones

### 1.5.2. Arquitectura de J2ME

La arquitectura Java™ 2 Micro Edition está orientada a pequeños dispositivos y sistemas embebidos como son teléfonos móviles, PDAs, Set-Top Boxes, máquinas expendedoras y un largo gama de productos existentes o futuros.

Al igual que sucede con J2EE™, que está orientado a entornos corporativos o J2SE™, orientado a sistemas de sobremesa, la arquitectura J2ME está formada por un conjunto de APIs estándares que permiten que las aplicaciones desarrolladas se beneficien de las características multiplataforma de Java y que abren la puerta a la distribución de aplicaciones a millones de dispositivos. Como podemos ver en el siguiente diagrama, la arquitectura J2ME se puede dividir en dos grandes bloques de arquitecturas que dependen del tipo de dispositivo y las características de los mismos. En función de la familia de dispositivos tomaremos una u otra opción.

En la figura I.02 se muestra dicha arquitectura.

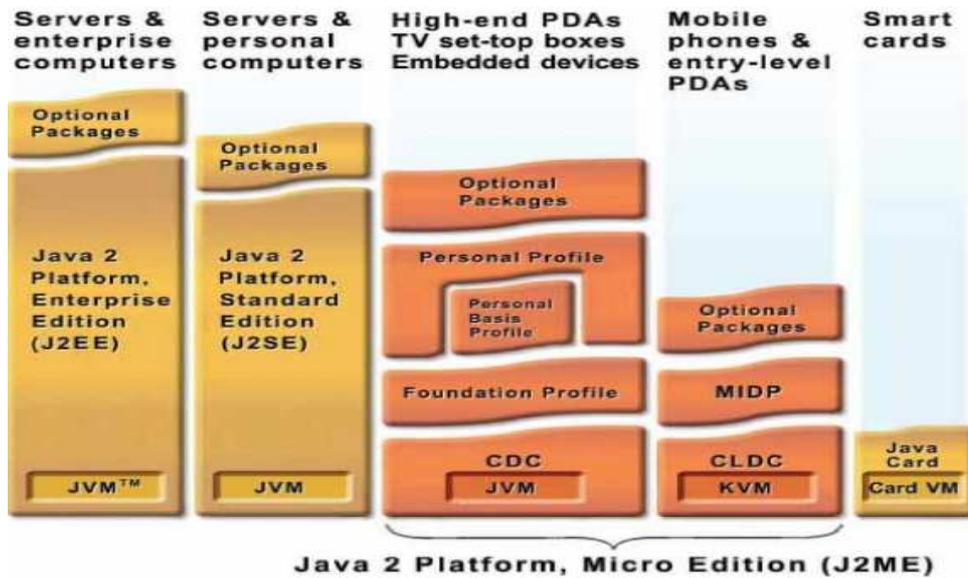


Fig. I.02.- Arquitectura J2ME

Para poder tener un entorno de ejecución Java para J2ME que cumpla los requisitos de un rango amplio de dispositivos y mercados objetivo es necesario que se componga de:

- Configuración
- Perfiles
- Paquetes Opcionales

Cada combinación de estos elementos se optimiza para la memoria, potencia de proceso y capacidades de E/S de una categoría de dispositivos.

### 1.5.3. Configuraciones J2ME

Las configuraciones se componen de una máquina virtual y un conjunto mínimo de bibliotecas de función. Proporcionan la funcionalidad básica para un conjunto de dispositivos que comparten características similares, tales como gestión de memoria o conectividad a la red.

En la actualidad existen dos configuraciones J2ME:

- Connected Limited Device Configuration (CLDC)
- Connected Device Configuration (CDC)

## ***CLDC***

Esta configuración está diseñada para dispositivos con conexiones de red intermitentes, procesadores lentos y memoria limitada: teléfonos móviles, asistentes personales (PDAs), etc. Es habitual que estos dispositivos tengan CPUs de 16 o 32 bits y un mínimo de entre 128 y 256 KB de memoria disponible para la implementación de la plataforma Java y sus aplicaciones asociadas. Los CLDC están basados en las máquinas virtuales K (K Virtual Machine, KVM).

## ***CDC***

Esta configuración está diseñada para dispositivos que tienen más memoria, procesadores más rápidos y un ancho de banda mayor, como Set-top boxes, pasarelas residenciales, asistentes personales de gran capacidad, etc. Incluye una máquina virtual Java completa (Java Virtual Machine, JVM) y un subconjunto de APIs de la arquitectura J2SE mucho mayor. Se orienta a dispositivos con CPU de 32 bits y un mínimo de 2MB de memoria disponible para la plataforma Java y aplicaciones asociadas.

### **1.5.4. Perfiles de J2ME**

Para conformar un entorno de ejecución completo orientado a una categoría de dispositivos, las configuraciones se han de combinar con un conjunto de APIs de un nivel más alto, llamadas perfiles, que van un paso más allá en la definición del modelo de ciclo de vida de las aplicaciones, la interfaz de usuario y acceso a las propiedades específicas de los dispositivos.

En la actualidad existen los siguientes perfiles asociados a J2ME:

- Mobile Information Device Profile (MIDP)
- Foundation Profile
- Personal Profile
- Personal Basis Profile

## MIDP (Mobile Information Device Profile)

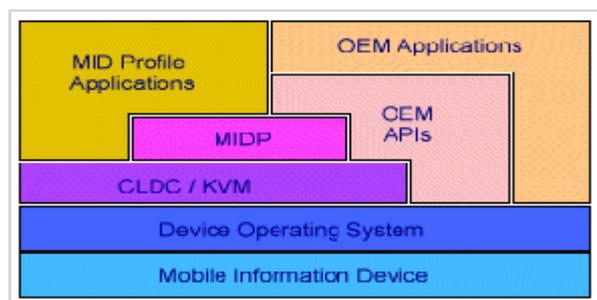
Está diseñado para teléfonos móviles y PDAs con capacidades básicas. Ofrece la funcionalidad básica para las aplicaciones móviles, incluyendo la interfaz de usuario, conectividad a redes, almacenamiento local de datos y gestión del ciclo de vida de cada una de las aplicaciones.

Al combinarlo con la configuración CLDC, MIDP proporciona un entorno de ejecución Java completo que incrementa la capacidad de los dispositivos móviles y que reduce el consumo de memoria y energía.

### Funcionalidad multimedia y de juego

MIDP incorpora un API de bajo nivel para la interfaz de usuario que complementa al API de alto nivel, permitiendo un control fino de gráficos e interfaz. El API de juegos incorpora funcionalidad específica para juegos, como son los "sprites", y capas, aprovechando las capacidades gráficas nativas de los dispositivos. El audio incorporado permite tonos, secuencias de tonos y archivos WAV.

La arquitectura MIDP se muestra en la figura I.03:



*Fig. I.03.- Arquitectura MIDP*

### Conectividad

MIDP permite utilizar las capacidades de mensajería y redes nativas de datos de los dispositivos de información móviles. Soporta estándares de conectividad que incluyen:

- HTTP
- HTTPS
- Datagramas
- Sockets

- Sockets de servidor
- Comunicación serie
- Servicio de mensajería (Short Message Service, SMS)
- Servicio de multidifusión (Cell Broadcast Service, CBS) de las redes GSM y CDMA (mediante el paquete opcional Wireless Messaging API, WMA)

## **Desarrollo con MIDP**

Existen dos versiones de MIDP:

- MIDP 2.0 (JSR 118): MIDP 2.0 es la versión revisada y mejorada de la especificación MIDP 1.0.
- MIDP 1.0 (JSR 37)

Los requisitos hardware mínimos que exige MIDP 1.0 son los siguientes:

- Pantalla: 96x54 con una profundidad de color por pixel de 1 bit
- Introducción de datos: Se ha de permitir alguno de los siguientes mecanismos:
  - Teclado de una mano (teclado de teléfono IUT-T)
  - Teclado de dos manos (teclado QWERTY)
  - Pantalla táctil
- Memoria (exclusivamente para aplicaciones MIDP):
  - 128 KB de memoria no volátil para las aplicaciones MIDP
  - 8 KB de memoria no volátil para los datos persistentes de la aplicación
  - 32 KB de memoria volátil para el entorno de ejecución Java (como por el Heap Java, por ejemplo)
- Trabajo en red: Ancho de banda limitada, bidireccional, sin cables, normalmente intermitente.

## **Implementación de MIDlets**

Las aplicaciones MIDP se denominan "MIDlets", las cuales pueden utilizar tanto las facilidades aportadas por MIDP como las APIs que MIDP hereda de CLDC, pero nunca acceden directamente al sistema operativo subyacente, por lo que no serían portables. Un MIDlet consiste en una clase Java, como mínimo, derivada de la clase abstracta MIDP, y que se ejecutan en un entorno de ejecución dentro de la máquina virtual, la cual provee un ciclo de vida bien definido controlado mediante métodos de la clase MIDlet que cada MIDlet se debe implementar. También estas aplicaciones usan métodos de esta clase abstracta para conseguir servicios de su entorno. Un grupo de MIDlets que están relacionados se suelen agrupar en un MIDlet suite. Todos estos MIDlet se empaquetan, instalan, desinstalan y borran como una única entidad y comparten recursos tanto en tiempo de ejecución virtual como estáticos.

El MIDP necesita 128 KB de RAM disponible para la implementación correspondiente. A esta cantidad debemos sumarle la que necesita CLDC y como mínimo 32 Kbytes para almacenar la pila de la aplicación, tamaño que obliga al programador a tener bastante cuidado a la hora de diseñar las aplicaciones. Además, los dispositivos MIDP cuentan con 8 Kbytes como mínimo de memoria no volátil que se utiliza como almacenamiento persistente, que no se borra tras apagar el aparato.

Sobre las pantallas de los dispositivos, la especificación MIDP indica que ésta requiere 96 pixels de ancho por 54 de alto y que debe soportar al menos dos colores. El rango de entrada de el dispositivo es muy amplio: desde los que tienen un teclado alfanumérico completo, hasta aquellos que permiten escribir en ciertas áreas de la pantalla, pasando por los teclados de los teléfonos móviles.

La especificación mínima requiere un teclado que permita marcar los números del 0 al 9, junto con el equivalente a las teclas del cursor y un botón de selección.

MIDP no asume que los dispositivos estén permanentemente conectados a una red, ni siquiera que soportan TCP/IP, pero que sí tienen algún tipo de acceso a una red. En este sentido, la especificación sí establece que soporte HTTP 1.1, bien mediante una pila de protocolos o una pasarela WAP.

## Ciclo de vida de un MIDlet

La figura I.04 muestra el ciclo de vida de una MIDlet:

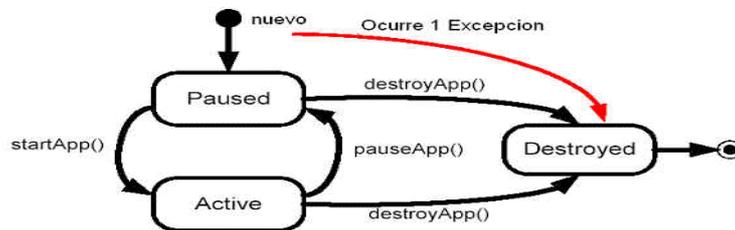


Fig. I.04.- Ciclo de vida de un MIDlet

- **startApp ()**
  - Método setCurrent () debería ser llamado aquí si no fue llamado antes. setCurrent () define que display será visible al usuario, solo un display puede ser visible al mismo tiempo.
  - Puede ser ejecutado más de una vez, así que es mejor no poner ninguna inicialización aquí
- **pauseApp()**
  - Cuando una aplicación es reactivada, puede aparecer con otro display usando el método setCurrent()
- **destroyApp()**
  - Libera y destruye todos los recursos usados por la aplicaciones, incluidos los componentes del interfaz de usuario

Estos tres métodos tienen que aparecer siempre.

## 1.6. Software de Desarrollo

### 1.6.1. NETBEANS

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un

archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos.

Durante el desarrollo del NetBeans IDE ocurrió una cosa interesante. La gente empezó a construir aplicaciones usando el NetBeans core runtime con sus propios plug-ins, de hecho, esto se convirtió en un mercado bastante grande con plataformas fácilmente integrables.

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

- Administración de las interfaces de usuario (ej. menús y barras de herramientas)
- Administración de las configuraciones del usuario
- Administración del almacenamiento (guardando y cargando cualquier dato)
- Administración de ventanas
- Framework basado en asistentes (diálogos paso a paso)
- El **IDE NetBeans** es un IDE - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación gratuito sin restricciones de uso.
- El **NetBeans IDE** es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

## 1.7. Tecnología Bluetooth

### 1.7.1. Definición

**Bluetooth** es una especificación industrial para Redes Inalámbricas de Área Personal (WPANs) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz. Los principales objetivos que se pretenden conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos...
- Eliminar cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Los dispositivos que con mayor frecuencia utilizan esta tecnología pertenecen a sectores de las telecomunicaciones y la informática personal, como PDA, teléfonos móviles, computadoras portátiles, ordenadores personales, impresoras o cámaras digitales.

Se denomina Bluetooth al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, con una cobertura baja y basada en transceptores de bajo costo. Gracias a este protocolo, los dispositivos que lo implementan pueden comunicarse entre ellos cuando se encuentran dentro de su alcance. Las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen que estar alineados y pueden incluso estar en habitaciones separadas si la potencia de transmisión lo permite. Estos dispositivos se clasifican como "Clase 1", "Clase 2" o "Clase 3" en referencia a su potencia de transmisión, siendo totalmente compatibles los dispositivos de una clase con los de las otras. La tabla. I.I muestra las diferencias entre cada una de las clases de Bluetooth:

Clase	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Rango (aproximado)
Clase 1	100 mW	20 dBm	~100 metros
Clase 2	2.5 mW	4 dBm	~25 metros
Clase 3	1 mW	0 dBm	~1 metro

Tabla. I.I.- Clases de bluetooth

En la mayoría de los casos, la cobertura efectiva de un dispositivo de clase 2 se extiende cuando se conecta a un transceptor de clase 1. Esto es así gracias a la mayor sensibilidad y potencia de transmisión del dispositivo de clase 1, es decir, la mayor potencia de transmisión del dispositivo de clase 1 permite que la señal llegue con energía suficiente hasta el de clase 2. Por otra parte la mayor sensibilidad del dispositivo de clase 1 permite recibir la señal del otro pese a ser más débil.

### 1.7.2. Topología de las redes Bluetooth

A diferencia de otras tecnologías LAN inalámbricas, como IEEE 802.11 (Wi-Fi), diseñadas para dispositivos que se hallen dentro o en los alrededores de un mismo edificio, los dispositivos que utilicen las redes PAN inalámbricas IEEE 802.15, incluyendo Bluetooth, podrán comunicarse en cualquier parte del mundo de forma stand-alone, incluso a bordo de un barco o avión y sin necesidad de utilizar equipo hardware adicional, como puntos de acceso. Cuando un dispositivo Bluetooth está dentro del radio de cobertura de otro, pueden establecer un enlace entre ellos. Hasta ocho unidades Bluetooth pueden comunicarse entre ellas y formar lo que se denomina una **Piconet** o **Picorred**.

La unión de varias piconets se denomina Scatternet o Red Dispersa lo cual se muestra en la figura I.05:

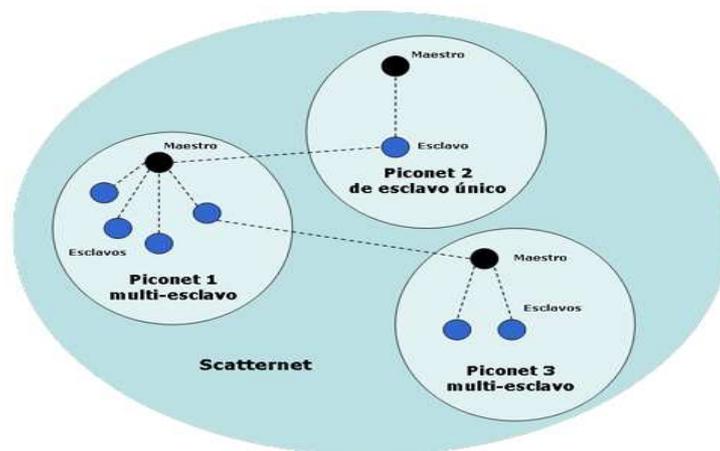


Fig. I.05.- Scatternet o Red Dispersa

Los dispositivos dentro de una piconet juegan dos papeles: maestro o esclavo. En todas las piconets sólo puede haber una unidad maestro, que normalmente es quien inicia la

conexión, el resto de unidades Bluetooth en la piconet se denominan esclavos. Cualquier dispositivo puede realizar las funciones de maestro y esclavo, pero un mismo dispositivo únicamente puede ser maestro de una piconet cuyo reloj y patrón de saltos se utilizan para sincronizar a todos los demás dispositivos esclavos.

La topología Bluetooth se puede describir como una estructura de piconets múltiples. Dado que la especificación Bluetooth soporta tantas conexiones punto a punto como punto a multipunto, se pueden establecer y enlazar varias piconets en forma de scatternet. Las piconets pertenecientes a una misma scatternet no están coordinadas y los saltos de frecuencia suceden de forma independiente, es decir, todos los dispositivos que participan en la misma piconet se sincronizan con su correspondiente tiempo de reloj y patrón de saltos determinado. Aunque no se permite la sincronización de diferentes piconets, los dispositivos pueden participar en diferentes piconets gracias a una multiplexación por división de tiempo (TDM). Esto permite a un dispositivo participar de forma secuencial en diferentes piconets, estando activo en sólo una piconet cada vez.

### **1.7.3. Principio de Funcionamiento**

Las redes de Bluetooth transmiten datos por medio de ondas de radio de baja potencia. Se comunica en una frecuencia de entre 2.402 GHz y 2.480 GHz, para ser exactos. Estas frecuencias se han dado por una decisión conjunta a nivel internacional para su uso en dispositivos industriales, científicos y médicos. Los monitores de control de bebés, las puertas de garaje automáticas, y las nuevas generaciones de teléfonos inalámbricos, hacen uso de las frecuencias en la banda ISM. Asegurarse de que Bluetooth y estos otros dispositivos no se interfieren entre ellos, ha sido una parte crucial en el proceso de diseño.

Una de las maneras en las que Bluetooth evita interferir con otros sistemas, es enviando señales muy débiles de más o menos 1 milivatio. Por comparación, los teléfonos móviles más potentes pueden transmitir una señal de 3 vatios. Esta baja potencia limita el rango de acción de un dispositivo Bluetooth, a unos 10 metros, atajando las posibilidades de interferencias entre tu sistema informático y tu teléfono móvil o televisión.

## **1.8. Modulo Bluetooth BLUESMIRF-RN-V1**

### **1.8.1. Descripción**

Éste módulo funciona en modo serial (TX/RX) y permite establecer una comunicación inalámbrica con una velocidad de transferencia de entre 9600 y 115200bps. El alcance aproximado es de unos 100 metros. Puede ser alimentado con una tensión de alimentación de entre 3.3 y 6V por lo que es ideal para su uso con baterías. Puede conectarse un ordenador. Se entrega sin conectores y utiliza el nuevo módulo Bluetooth de Roving Networks, por lo que no es compatible con los comandos de la antigua versión "Mitsumi". Éste utiliza un set de comandos mucho más funcional y robusto que su predecesor.

### **1.8.2. Características**

- FCC Approved Class 1 Bluetooth<sup>®</sup> Radio Modem
- Muy pequeño: 51.5x15.8x5.6mm
- Link inalámbrico muy robusto con verificación de datos hasta 100m
- Bajo consumo: 25mA
- Operar al mismo tiempo que otras redes tales como WiFi, 802.11g y Zigbee
- Conexión encriptada
- Frecuencia: 2.4~2.524 GHz
- Alimentación: 3.3V-6V
- Comunicación serie:: 2400-115200bps
- Rango de temperatura: -40 ~ +70C
- Antena integrada
- Comunicaciones seguras cifrado de 128 bits
- Corrección de errores para garantizar la entrega de los datos
- Modulación GFSK, 79 canales en intervalos de 1 MHz

### **1.8.3. Aplicaciones**

- Se utiliza como sustitución de cables
- Escáneres de códigos de barras
- Sistemas de medición y control

- Sensores industriales y los controles
- Aplicaciones médicas

La figura I.06 muestra el modulo Bluetooth de manera de física:

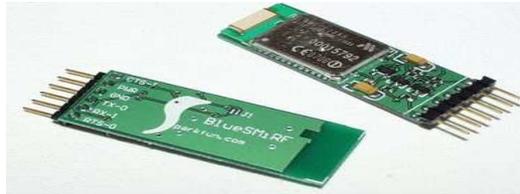


Fig. I.06.- Bluetooth BLUESMIRF-RN-V1

La figura I.07 muestra la disposición de pines del módulo BLUESMIRF-RN-V1

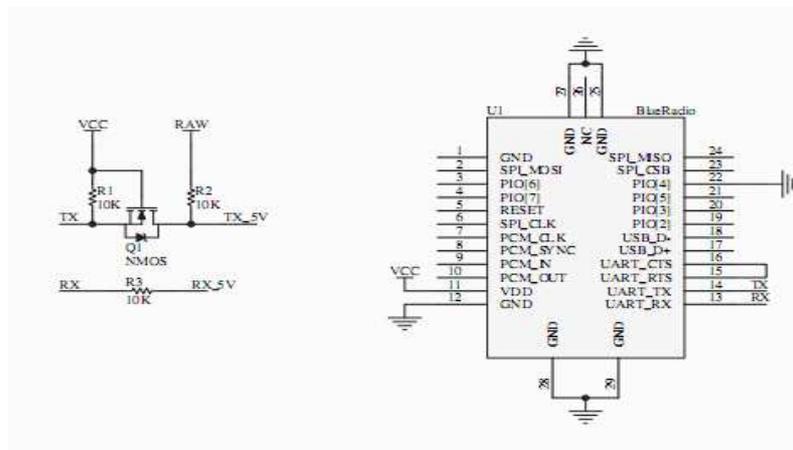


Fig. I.07.- Disposición de pines modulo BLUESMIRF-RN-V1

## 1.9. Modulación GFSK

La modulación por desplazamiento de frecuencia gaussiana (en inglés Gaussian Frequency Shift Keying o GFSK) es un tipo de modulación donde un 1 lógico es representado mediante una desviación positiva (incremento) de la frecuencia de la onda portadora, y un 0 mediante una desviación negativa (decremento) de la misma.

GFSK es una versión mejorada de la modulación por desplazamiento de frecuencia (FSK). En GFSK la información es pasada por un filtro gaussiano antes de modular la señal. Esto se traduce en un espectro de energía más estrecho de la señal modulada, lo cual permite mayores velocidades de transferencia sobre un mismo canal.

La figura I.08 muestra un filtro gaussiano para datos:

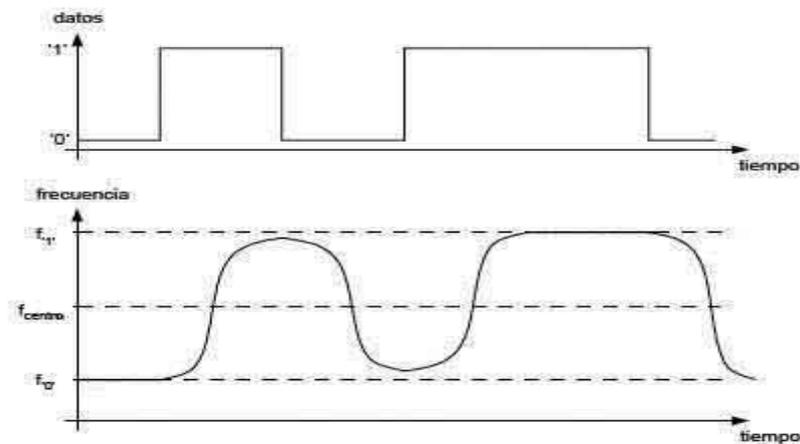


Fig. I.08.- Filtro Gaussiano para datos

## 1.10 Microcontroladores

### 1.10.1 Microcontrolador PIC 16F628A

El PIC 16F628A es un microcontrolador de gama baja de 18 pines fabricado por la empresa MICROCHIP que incorpora las mejores características para los Microcontroladores de esta gama:

El PIC 16F628A incorpora tres características importantes que son:

- Procesador tipo RISC (Procesador con un Conjunto Reducido de Instrucciones)
- Procesador segmentado
- Arquitectura HARVARD

Con estos recursos el PIC es capaz de ejecutar instrucciones solamente en un ciclo de instrucción. Con la estructura segmentada se pueden realizar simultáneamente las dos fases en que se descompone cada instrucción, ejecución de la instrucción y búsqueda de la siguiente.

La separación de los dos tipos de memoria son los pilares de la arquitectura Harvard, esto permite acceder en forma simultánea e independiente a la memoria de datos y a la de instrucciones. El tener memorias separadas permite que cada una tenga el ancho y

tamaño más adecuado. Así en el PIC 16F628A el ancho de los datos es de un byte, mientras que la de las instrucciones es de 14 bits.

### 1.10.2. Características

- Conjunto reducido de instrucciones (RISC). Solamente 35 instrucciones que aprender a utilizar
- Oscilador interno de 4MHz
- Las instrucciones se ejecutan en un sólo ciclo de máquina excepto los saltos (*goto* y *call*), que requieren 2 ciclos. Aquí hay que especificar que un ciclo de máquina se lleva 4 ciclos de reloj, si se utiliza el reloj interno de 4MHz, los ciclos de máquina se realizarán con una frecuencia de 1MHz, es decir que cada instrucción se ejecutará en 1µs (microsegundo)
- Opera con una frecuencia de reloj de hasta 20 MHz (ciclo de máquina de 200 ns)
- Memoria de programa: 2048 locaciones de 14 bits
- Memoria de datos: Memoria RAM de 224 bytes (8 bits por registro)
- Memoria EEPROM: 128 bytes (8 bits por registro)
- Stack de 8 niveles
- 16 Terminales de I/O que soportan corrientes de hasta 25 mA
- 3 Temporizadores
- Módulos de comunicación serie, comparadores, PWM

Otra característica de los PICs es el manejo de los bancos de registros. En línea general, los registros se clasifican como de uso general (GPR) y de uso específico o de funciones especiales (SFR).

- Los registros de uso general pueden ser usados directamente por el usuario, sin existir restricciones. Pueden servir para almacenar resultados que se reciben desde el registro W (acumulador), datos que provienen de las puertas de entradas, etc.
- Los registros de uso específicos no pueden ser usados directamente por el usuario. Estos registros controlan prácticamente todo el funcionamiento del microcontrolador, pues toda la configuración necesaria para funcionamiento del microcontrolador es hecho a través de algún tipo de SFR.

### 1.10.3. Pines de I/O (Entrada/Salida)

La figura I.09 muestra los pines del PIC:

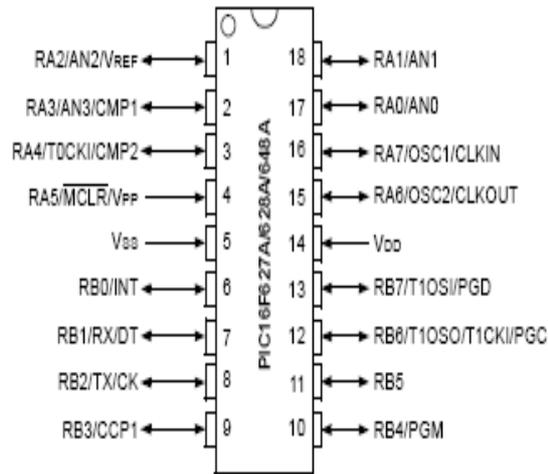


Fig. I.09.- PIC 16f628A

#### **PUERTO A: RA0-RA7**

- Los pines RA0-RA4 y RA6–RA7 son bidireccionales y manejan señales TTL
- El pin RA5 es una entrada Schmitt Trigger que sirve también para entrar en el modo de programación cuando se aplica una tensión igual a  $V_{pp}$  (13,4V mínimo)
- El terminal RA4 puede configurarse como reloj de entrada para el contador TMR0
- Los pines RA0-RA3 sirven de entrada para el comparador analógico

#### **PUERTO B: RB0-RB7**

- Los pines RB0-RB7 son bidireccionales y manejan señales TTL
- Por software se pueden activar las resistencias de pull-up internas, que evitan el uso de resistencias externas en caso de que los terminales se utilicen como entrada (permite, en algunos casos, reducir el número de componentes externos)
- El pin RB0 se puede utilizar como entrada de pulsos para provocar una interrupción externa

- Los pines RB4-RB7 están diseñados para detectar una interrupción por cambio de estado. Esta interrupción puede utilizarse para controlar un teclado matricial, por poner un ejemplo.

## **Otros pines**

- VDD: Pin de alimentación positiva. De 2 a 5,5 Vcc
- VSS: Pin de alimentación negativa. Se conecta a tierra o a 0 Vcc
- MCLR: Master Clear (Reset). Si el nivel lógico de este terminal es bajo (0 Vcc), el microcontrolador permanece inactivo. Este Reset se controla mediante la palabra de configuración del PIC
- OSC1/CLKIN: Entrada de oscilador externo
- OSC2/CLKOUT: Salida del oscilador. El PIC 16F628 dependiendo de cómo se configure puede proporcionar una salida de reloj por medio de este pin

## **1.11. Cilindro Eléctrico**

### **1.11.1. Descripción**

Los cilindros eléctricos se aplican en todos los sectores de producción de la industria, empezando por válvulas para chatarra en prensas, pasando por sistemas de reajuste para la construcción en bruto, hasta llegar a sistemas de transporte y montaje, desde el pintado, hasta el montaje final.

Por lo cual es un instrumento de ayuda muy importante para la realización de diversos tipos de máquinas el cual es muy eficiente debido a su pequeño diseño, pero con grandes prestaciones al momento de realizar trabajos pesados ya que su fuerza no depende del cilindro solamente, funciona con un motor de corriente continua el cual le proporciona toda la fuerza que necesita de acuerdo al trabajo a realizar.

### **1.11.2. Principios de funcionamiento**

En los husillos trapeciales con un pequeño ángulo de paso está garantizada una autorretención de los cilindros eléctricos. Los esfuerzos estáticos que pueden ser mucho mayores que la fuerza de desplazamiento nominal, son mantenidos con seguridad en cualquier posición intermedia, independientemente del emplazamiento. Para aumentar

las velocidades de desplazamiento, se utilizan husillos trapeciales de filetes múltiples y/o pasos de husillo más grandes. Ya que la autorretención de estos husillos está suspendida, la función de retención se puede realizar montando un freno electromagnético. Lo cual se muestra en la figura I.10:



*Fig. I.10.- Estructura interna de un cilindro eléctrico*

Para alcanzar una larga vida útil con una alta frecuencia de maniobras, se utilizan preferentemente transmisiones por bolas rodantes con un mayor rendimiento específico. En virtud del contacto rodante entre las bolas y las vías de rodadura en el sistema husillo/tuerca, se origina sólo una fricción muy pequeña en comparación con las transmisiones por rosca trapecial. El posicionamiento preciso en cualquier posición intermedia y la retención de esta posición están garantizados por un freno electromagnético.

### **1.11.3. Ventajas**

Las ventajas de usar un cilindro eléctrico son las siguientes:

- Elevado nivel de rendimiento mecánico
- Larga vida útil, bajos costes de mantenimiento
- Instalación y puesta en marcha sencilla y económica
- Baja potencia eléctrica y bajo consumo de corriente
- Prácticamente libre de mantenimiento
- Posicionamiento exacto, fácil de regular
- Muy dinámico
- Incremento de fuerza cinco veces más rápido que en los cilindros neumáticos
- Ajuste constante de la fuerza
- Conexión sencilla a los procesos de automatización

## **1.12. Motores Eléctricos**

### **1.12.1. Descripción**

Un motor eléctrico es una máquina eléctrica que transforma energía eléctrica en energía mecánica por medio de interacciones electromagnéticas. Algunos de los motores eléctricos son reversibles, pueden transformar energía mecánica en energía eléctrica funcionando como generadores. Los motores eléctricos de tracción usados en locomotoras realizan a menudo ambas tareas, si se los equipa con frenos regenerativos.

Son ampliamente utilizados en instalaciones industriales, comerciales y de particulares. Pueden funcionar conectados a una red de suministro eléctrico o a baterías. Así, en automóviles se están empezando a utilizar en vehículos híbridos para aprovechar las ventajas de ambos.

### **1.12.2. Principio de funcionamiento**

Los motores de corriente alterna y los motores de corriente continua se basan en el mismo principio de funcionamiento, el cual establece que si un conductor por el cual circula una corriente eléctrica se encuentra dentro de la acción de un campo magnético, éste tiende a desplazarse perpendicularmente a las líneas de acción del campo magnético.

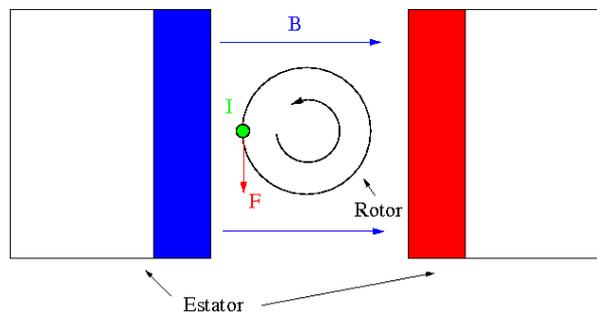
El conductor tiende a funcionar como un electroimán debido a la corriente eléctrica que circula por el mismo adquiriendo de esta manera propiedades magnéticas, que provocan, debido a la interacción con los polos ubicados en el estator, el movimiento circular que se observa en el rotor del motor.

Según la Ley de Lorentz, cuando un conductor por el que pasa una corriente eléctrica se sumerge en un campo magnético, el conductor sufre una fuerza perpendicular al plano formado por el campo magnético y la corriente, siguiendo la regla de la mano derecha, con módulo, lo cual se muestra en la figura I.11:

$$F = B \cdot l \cdot I$$

- F: Fuerza en Newtons
- I: Intensidad que recorre el conductor en amperios

- $l$ : Longitud del conductor en metros lineales
- $B$ : Inducción en teslas



*Fig. I.11.- Estructura de un Motor de Corriente Continua*

La figura I.12 muestra un motor de corriente continua:



*Fig. I.12.- Motor de corriente continua*

Partiendo del hecho de que cuando pasa corriente eléctrica por un conductor se produce un campo magnético, además si lo ponemos dentro de la acción de un campo magnético potente, el producto de la interacción de ambos campos magnéticos hace que el conductor tienda a desplazarse produciendo así la energía mecánica. Dicha energía es comunicada al exterior mediante un dispositivo llamado flecha.

### 1.13. Motores de Arranque

#### 1.13.1. Descripción

Se denomina **motor de arranque** o motor de partida a un motor eléctrico de corriente continua del tipo excitación separada (cuenta con imanes) que posee reducidas dimensiones y que se utiliza para facilitar la puesta en marcha de los motores de combustión interna, para que pueda vencer la resistencia inicial que ofrecen los órganos cinemáticos del motor en su inicio de funcionamiento. Los hay tanto en motores de dos tiempos como en los de cuatro tiempos.

La figura I.13 muestra un motor de arranque:



*Fig. I.13.- Motor de arranque*

El motor de arranque tiene la función de hacer girar el cigüeñal del motor térmico con el fin de que comience el ciclo de explosión o combustión, y hasta que este último es capaz de continuar por sí solo.

### **1.13.2. Partes de un Motor de Arranque**

Los motores de arranque constan de dos elementos principales:

- Motor eléctrico simple suele ser un motor "serie" de CC, quiere decir que la corriente pasa inicialmente por sus bobinas inductoras y a continuación por el inducido sin ninguna derivación. Este tipo de motor se caracteriza por un elevado par de arranque que lo hace óptimo en esta aplicación.
- El relé principal de arranque que tiene la misión de conectar al motor eléctrico con la batería directamente y en segundo lugar desplazar el piñón del arranque para que este se conecte con la corona del volante de inercia del motor térmico y así poder transmitir el giro del arranque al cigüeñal.

### **1.13.3. Características**

La principal característica del motor de corriente continua es la posibilidad de regular la velocidad desde vacío a plena carga.

Una máquina de corriente continua (generador o motor) se compone principalmente de dos partes, un estator que da soporte mecánico al aparato y tiene un hueco en el centro generalmente de forma cilíndrica. En el estator además se encuentran los polos, que pueden ser de imanes permanentes o devanados con hilo de cobre sobre núcleo de hierro. El rotor es generalmente de forma cilíndrica, también devanado y con núcleo, al que llega la corriente mediante dos escobillas.

### 1.13.4. Ventajas

Las ventajas de usar motores de corriente continua son:

- Su potencia, tamaño y peso son más reducidos.
- Se pueden construir de cualquier tamaño.
- Tiene un par de giro elevado y, según el tipo de motor, prácticamente constante.
- Su rendimiento es muy elevado (típicamente en torno al 75%, aumentando el mismo a medida que se incrementa la potencia de la máquina).
- Este tipo de motores no emite contaminantes, aunque en la generación de energía eléctrica de la mayoría de las redes de suministro se emiten contaminantes.

### 1.14. Puente H

#### 1.14.1 Descripción

Un **Puente H** es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, *avance* y *retroceso*. Son ampliamente usados en robótica y como convertidores de potencia. Los puentes H están disponibles como circuitos integrados, pero también pueden construirse a partir de componentes discretos, ya que la interfaz de potencia para motores de CC, PUENTE H, es un sistema que permite controlar motores en rangos entre 12 y 30 voltios y con consumos de hasta 30 amperios por medio de relés de potencia.

La figura I.14 siguiente la estructura básica de un puente H.

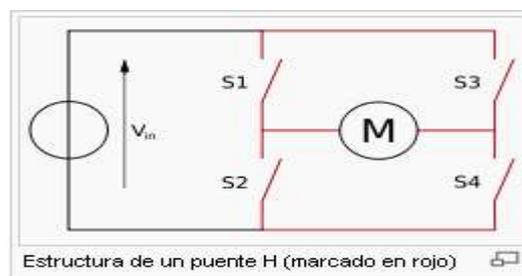


Fig. I.14.- Estructura básica de un Puente H

El término "puente H" proviene de la típica representación gráfica del circuito. Un puente H se construye con 4 interruptores (mecánicos o mediante transistores). Cuando los interruptores S1 y S4 (ver primera figura) están cerrados (y S2 y S3 abiertos) se

aplica una tensión positiva en el motor, haciéndolo girar en un sentido. Abriendo los interruptores S1 y S4 (y cerrando S2 y S3), el voltaje se invierte, permitiendo el giro en sentido inverso del motor.

Con la nomenclatura que estamos usando, los interruptores S1 y S2 nunca podrán estar cerrados al mismo tiempo, porque esto cortocircuitaría la fuente de tensión. Lo mismo sucede con S3 y S4. La figura I.15 muestra los estados de un Puente H:

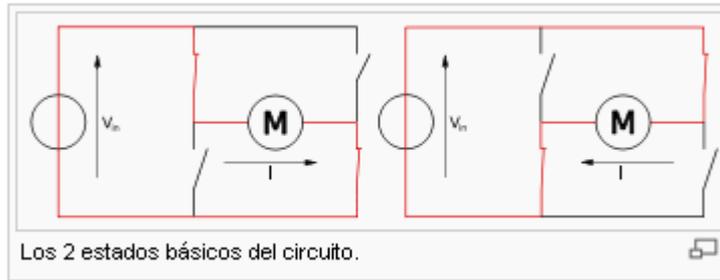


Fig. I.15.- Estados de un Puente H

### 1.14.2 Aplicaciones

Como hemos dicho el puente H se usa para invertir el giro de un motor, pero también puede usarse para frenarlo (de manera brusca), al hacer un corto entre las bornas del motor, o incluso puede usarse para permitir que el motor frene bajo su propia inercia, cuando desconectamos el motor de la fuente que lo alimenta.

En la Tabla I.VI se muestra un cuadro de resumen de las diferentes acciones que posee un Puente H:

S1	S2	S3	S4	Resultado
1	0	0	1	El motor gira en <i>avance</i>
0	1	1	0	El motor gira en <i>retroceso</i>
0	0	0	0	El motor se detiene bajo su inercia
0	1	0	1	El motor frena ( <i>fast-stop</i> )

Tabla. I.II.- Resumen de Acciones de un Puente H

### **1.14.3 Montaje**

Lo más habitual en este tipo de circuitos es emplear interruptores de estado sólido (como Transistores), puesto que sus tiempos de vida y frecuencias de conmutación son mucho más altos. Pero en la aplicación de los Robots de Batalla es recomendable la utilización de Relés de alto amperaje debido al alto consumo de corriente que presentan tanto los motores de CC y de arranque los cuales utilizaremos para el funcionamiento de las armas del robot.

Además los interruptores se acompañan de diodos (conectados a ellos en paralelo) que permitan a las corrientes circular en sentido inverso al previsto cada vez que se conmute la tensión, puesto que el motor está compuesto por bobinados que durante breves períodos de tiempo se opondrán a que la corriente varíe.

### **1.14.4 Características de la interfaz Puente H**

Las principales características del Puente H se enumeran a continuación:

- Activación de motores en un rango entre 9 y 30 voltios DC.
- Capacidad para entregar hasta 30 Amperios a la carga.
- Capacidad para activar el giro del motor en cualquiera de los dos sentidos.
- Fácil y rápida conexión gracias a sus borneras de tornillo.
- Reducido tamaño.
- Excelente calidad.
- Alta confiabilidad.
- Capacidad de responder a las variaciones de corriente de una manera efectiva.

## **1.15. Proteus**

### **1.15.1. Descripción**

**Proteus** es una compilación de programas de diseño y simulación electrónica, desarrollado por Labcenter Electronics que consta de los dos programas principales: Ares e Isis, y los módulos VSM y Electra.

Proteus es un entorno integrado diseñado para la realización completa de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño, simulación, depuración y construcción.

### **1.15.2. Aspectos Básicos del Programa Proteus**

PROTEUS es un programa para simular circuitos electrónicos complejos integrando inclusive desarrollos realizados con microcontroladores de varios tipos, en una herramienta de alto desempeño con unas capacidades graficas impresionantes.

Presenta una filosofía de trabajo semejante al SPICE, arrastrando componentes de una barra e incrustándolos en la aplicación.

### **1.15.3. Características de Proteus**

- Entorno de diseño gráfico de esquemas electrónicos (ISIS) extremadamente fácil de utilizar y dotado de poderosas herramientas para facilitar el trabajo del diseñador.
- Entorno de diseño de placas de circuito impreso (ARES) de ultra-altas prestaciones con bases de datos de 32 bits, posicionador automático de elementos y generación automática de pistas con tecnologías de autocorte y regeneración.
- Moderno y atractivo interface de usuario estandarizado a lo largo de todas las herramientas que componen el entorno PROTEUS.
- La mayor parte de los módulos que componen PROTEUS han sido escritos por el mismo equipo, garantizando al máximo nivel posible la compatibilidad e inter-operatividad asegurando su estabilidad al máximo.
- Ejecutable en los diferentes entornos Windows: 98, Me, 2000, XP.
- Herramienta de máximas prestaciones, basadas en los más de 15 años de continuo desarrollo y presencia en el mercado.
- Miles de instalaciones vendidas en más de 35 países a todo lo largo del mundo.

#### **1.15.4. Elementos de Proteus**

La suite se compone de cuatro elementos, perfectamente integrados entre sí los cuales son:

- ISIS
- ARES
- VSM
- Electra

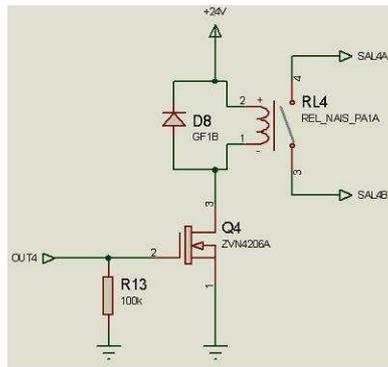
De los cuales utilizaremos ISIS Y ARES, son los que nos servirán para el diseño de los circuitos electrónicos del robot de batalla con una mayor facilidad.

#### **1.16. Isis**

##### **1.16.1. Introducción**

El Programa ISIS, **I**ntelligent **S**chematic **I**nput **S**ystem (*Sistema de Enrutado de Esquemas Inteligente*) permite diseñar el plano eléctrico del circuito que se desea realizar con componentes muy variados. **ISIS** se suministra con una extensa librería compuesta por más de 8.000 dispositivos. Esta librería incluye elementos estándar como resistencias, condensadores, transistores, diodos, válvulas, TTL, CMOS, ECL, microprocesadores, memorias, PLDs, ICs analógicos y amplificadores operacionales, sócalos, conectores de Tierra, positivo y otros componentes con prestaciones diferentes. ISIS es la herramienta ideal para una rápida realización de complejos diseños de esquemas electrónicos destinados tanto a la construcción de equipos electrónicos como a la realización de tareas de simulación y prueba. Además, ISIS es una herramienta excepcional para la realización de atractivos esquemas electrónicos destinados a su publicación en libros, manuales o documentos técnicos, e inclusive, antes de hacer el plano electrónico, se pueden elegir rotulados y tamaños para impresión, desde tamaños portátiles (A5, A4, A3), hasta tamaños estilo plotter (grande): A2, A1 e inclusive, A0.

La figura I.16 muestra un esquema de ISIS:



*Fig. I.16.- Esquema de ISIS*

**ISIS** viene provisto de las habituales herramientas de cortar, copiar y pegar hacia y desde el portapapeles de Windows. De esta manera usted puede copiar, de manera muy sencilla, la totalidad o parte de su esquema electrónico y pegarla en cualquier otra herramienta que utilice el estándar del portapapeles de Windows.

Además, la librería incorpora dispositivos específicos de los fabricantes más importantes como Philips, National Semiconductor, Motorola, Teccor, Texas y Zetex.

Un proceso de actualización permanente posibilita que cada cuatro meses la librería se incremente con nuevos productos aparecidos en el mercado. Los usuarios registrados pueden sugerir la inclusión de nuevos productos en dicha librería.

### **1.16.2. Herramienta Gráfica para la Gestión de los Encapsulados**

El establecimiento de los enlaces entre los esquemas electrónicos y los encapsulados de los dispositivos para su utilización en el diseño de las placas se ha simplificado enormemente con la herramienta gráfica para la gestión de los encapsulados.

La figura I.17 muestra el encapsulado de Isis:

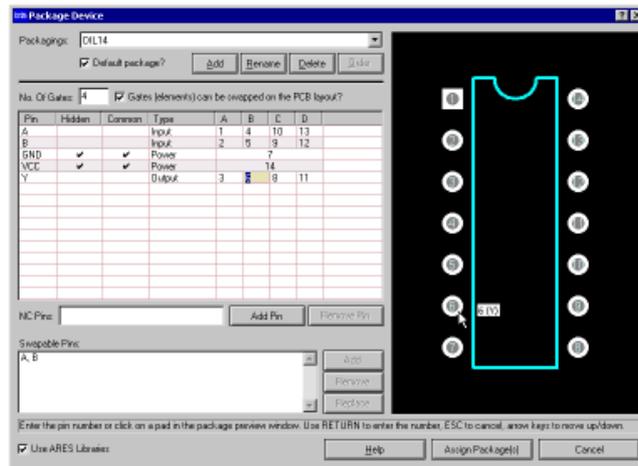


Fig. I.17.- Encapsulado de Isis

Con esta herramienta se dispone de la vista de la huella (footprint) del componente electrónico junto a una hoja de datos con cada uno de los pines definidos en el esquema electrónico de ese componente, de esta manera facilitando el diseño.

La asignación de los pines del esquema y los pines de la huella puede realizarse tanto de forma gráfica, como introduciendo la información directamente como texto.

### 1.16.3. Completa Gestión de Diseños Jerárquicos

En la actualidad los circuitos electrónicos han aumentado su complejidad incluso en los proyectos más pequeños. Para facilitar la gestión de los esquemas complejos, **ISIS** incorpora la gestión de diseños multihojas, es decir, permite repartir un esquema electrónico en varias hojas de papel de forma rápida y cómoda.

Por añadidura, la potencia de **ISIS** para el tratamiento de esquemas complejos se pone de manifiesto con su **gestión de los diseños jerárquicos**. El diseño jerárquico consiste en que un particular componente puede ser definido como un módulo que representa un determinado esquema electrónico y que se comporta de acuerdo con él.

En la figura anterior podemos ver un ejemplo. El amplificador operacional 741 se ha definido como un módulo al que hemos asignado el correspondiente esquema electrónico. En cualquiera de nuestros diseños podemos utilizar dicho módulo 741 y lo

conectaremos al resto de los componentes que forman nuestro esquema por medio de los pines definidos. A partir de ese momento.

Evidentemente, a lo largo de nuestro diseño, podemos utilizar el módulo tantas veces como deseemos e **ISIS** gestionará automáticamente su sustitución cada vez por el esquema representado. De esta manera la utilización de subcircuitos que se repiten varias veces en nuestro diseño queda enormemente simplificada y el tiempo utilizado en el diseño se reduce drásticamente.

#### 1.16.4. Utilización de Buses

Un bus es una serie de conductores en paralelo. Su uso más extendido es la gestión de las líneas de datos y direcciones en el diseño de microprocesadores. La figura I.18 muestra la disposición de buses en Isis:

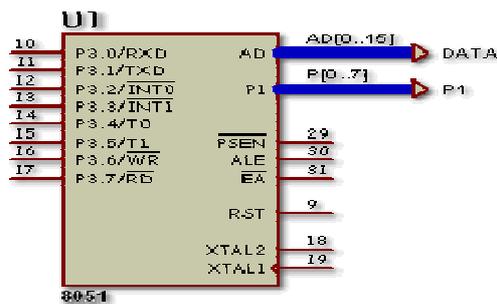


Fig. I.18.-Buses de Isis

#### 1.16.5. Administrador de Propiedades de los Componentes

Cada uno de los componentes que se incluye en un diseño electrónico posee una lista de propiedades. Algunas de ellas se refieren de forma específica a su empaquetado o al comportamiento del modelo utilizado para la simulación de los circuitos electrónicos. **ISIS** además le permite añadir las propiedades que usted desee para sus propias necesidades, incluye una completa colección de herramientas para permitir la selección de grupos de componentes de acuerdo con sus propiedades o la manipulación de las propiedades de un determinado grupo de componentes.. La herramienta para la importación de datos en formato ASCII (ADI) que incorpora **ISIS**, posibilita la importación de propiedades desde ficheros formato ASCII utilizando reglas muy simples para determinar a qué tipo de componentes se deben asignar.

La figura I.19 muestra los componentes de uno de los diferentes dispositivos utilizados para el diseño de circuitos:

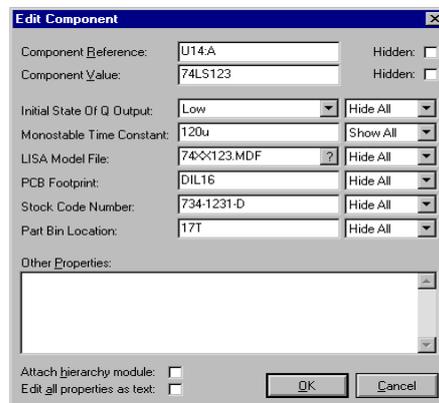


Fig. I.19.- Diagrama de componentes de Isis

Cuando se crea un nuevo elemento de librería, se pueden definir las propiedades de ese componente y sus valores por defecto. Además cada propiedad puede ser definida totalmente, tanto en lo referente a su, como a su comportamiento (visible, editable, etc.).

## 1.17. Ares

Es la herramienta para la elaboración de placas de circuito impreso con posicionador automático de elementos y generación automática de pistas, que permite el uso de hasta 16 capas. Con ARES el trabajo duro de la realización de placas electrónicas recae sobre el PC en lugar de sobre el diseñador. Como se muestra en la figura I.20:

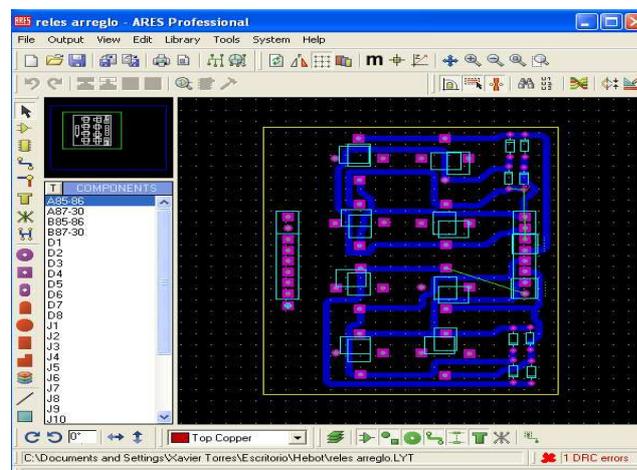


Fig. I.20.- Entorno de ARES

### 1.17.1. Entorno de Trabajo de Ares

- **Menús:** Encontramos los menús para guardar el diseño, crear los ficheros GERBER, distintas opciones de visualización, ruteado, opciones...etc.
- **Barra Herramientas:** Selección de las herramientas para la creación de las pistas, comprobación del netlist...etc.
- **Barra Selección:** nos permitirá seleccionar la capa de trabajo, dentro de la pantalla de edición, también podremos seleccionar los elementos que deseemos...etc
- **Barra de Estado:** nos muestra la posición en la que se encuentra el cursor, esta se puede dar en Th (mils) o milímetros (mm). La relación que hay entre estas medidas es la siguiente: 40mils = 1mm, de forma que con esta relación ya podemos hacernos una idea de las medidas.
- **Pantalla de Edición:** Aquí realizaremos el diseño de nuestro ruteo.

Una vez visto nuestro entorno de trabajo comenzaremos a ver las herramientas básicas para poder realizar el ruteo.

### 1.17.2. Herramientas

Dentro de la barra de herramientas, disponemos de la herramienta “Package Mode”, que nos permitirá escoger el encapsulado que deseemos añadir.

- **Realización del Diseño de la PCB:**

A continuación mostraremos los pasos necesarios para poder llevar a cabo el ruteo de una placa PCB a partir del esquemático generado en ISIS.

- **Creación de la placa PCB:**

En primer lugar deberemos delimitar nuestra placa PCB, para ello debemos escoger la herramienta “2D Graphics Box” . , y seleccionar la capa Board Edge, con esto ya podemos realizar el borde de nuestra PCB.

- **Posicionamiento de los Componentes:**

Partiendo del hecho de que todos los componentes que tenemos en la placa disponen de encapsulado pasamos a realizar el posicionamiento de los mismos, para ello existen 2 opciones:

**1- Posicionamiento automático:** Realizado por ARES, para lo cual bastará con seleccionar dentro del menú Tools la herramienta “Auto Placer”. Para poder utilizar esta herramienta, es necesario haber definido previamente los límites de nuestra placa.

**2- Posicionamiento manual:** Mediante los botones presentados en la figura I.21:



Fig. I.21.- Posicionamiento manual de selección de componentes

- **Ruteo de las pistas**

Con los componentes ya dispuestos en su posición, pasamos a realizar el ruteo de las pistas. Se debe tener en cuenta que las líneas “verdes” que son las líneas correspondientes a las uniones (netlist) entre los componentes, es decir las líneas que indican qué está conectado con qué.

La figura I.22 muestra cómo debemos hacer el ruteo en ares:

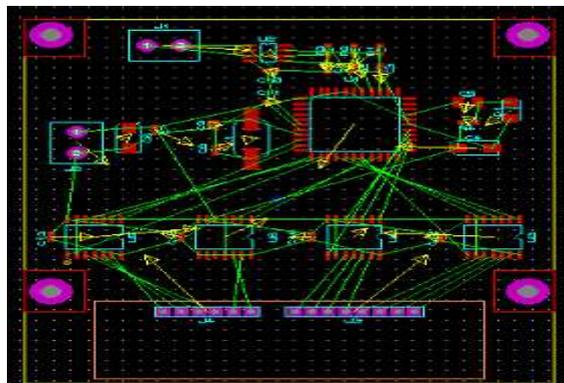


Fig. I.22.- Ruteo en ARES

**Ruteo Manual:** Para ello solo tendremos que pinchar sobre el icono “Track mode” de la barra de herramientas. La figura I.23 muestra el ruteo manual:

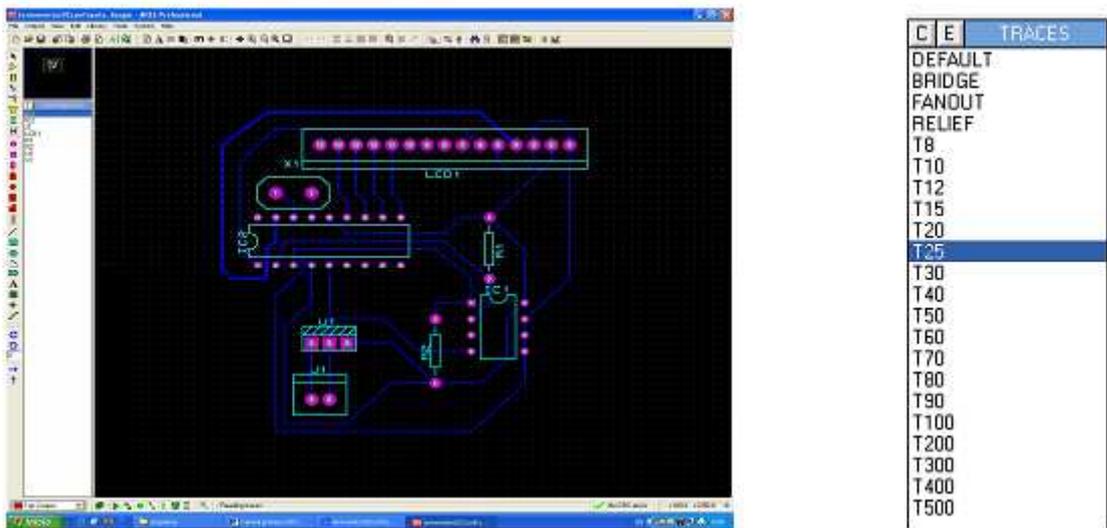


Fig. I.23.- Ruteo Manual

## 1.18. Microcode Studio

### 1.18.1. Introducción

El MicroCode Studio es una interface en el cual escribes el código del programa, te corrige errores de sintaxis, te ordena visualmente las subrutinas... El Microcode queda enlazado con el PICBASIC y el IC-PROG, de manera que una vez que terminas el programas, compilas y generas el archivo \*.HEX, los programas los guardas en formato Picbasic \*.BAS.

El explorador de código le permite automáticamente saltar a incluir los archivos, define, constantes, variables, alias y modificadores, los símbolos y las etiquetas, que están contenidos dentro de su código fuente. Permite cortar, copiar, pegar y deshacer se proporciona, junto con las características de búsqueda y reemplazo.

Microcode Studio ahora incluye EasyHID Wizard, una herramienta de generación de código que permite a un usuario implementar rápidamente la comunicación bidireccional entre un microcontrolador PIC integrado y un PC.

### 1.18.2 Características

- Completo resultado de sintaxis de su código fuente.
- Saltar rápidamente para incluir los archivos, los símbolos, definir las variables y las etiquetas utilizando la ventana de explorador de código.
- Identificar y corregir la compilación y el ensamblador de errores.
- Ver la salida de serie de su microcontrolador.
- Palabra clave basada en el contexto de ayuda sensible al soporte para MPASM.
- Es fácil de configurar su compilador, ensamblador y las opciones de programador o puede dejar que Microcode Studio lo haga.
- Compilación y errores en ensamblador puede ser fácilmente identificada y corregida mediante la ventana de error los resultados.
- Simplemente haga clic en un error de compilación y Microcode Studio lo llevará automáticamente a la línea de error.
- Microcode Studio incluso viene con una ventana de comunicaciones serie, que le permite depurar y ver la salida de serie de su microcontrolador.

### 1.18.3 In Circuit Debugger (ICD)

En el circuito depurador (ICD), cada línea de código fuente está animado en la ventana del editor principal, mostrando que la línea del programa está siendo ejecutado por el microcontrolador de acogida. Usted puede incluso cambiar varios puntos de interrupción y paso a través de su línea de PICBASIC PRO código línea por línea.

Microcode Studio es **totalmente gratuito** para uso no comercial. Sin embargo, sólo puede utilizar un modelo ICD de Microcode Studio.

Microcode no tiene derechos de autor.

### 1.18.4. Pic que Soporta Microcode Studio

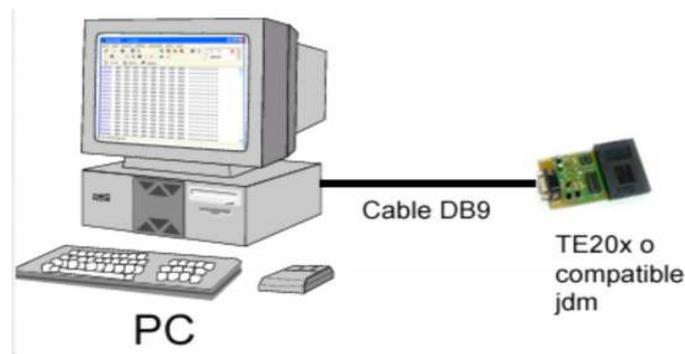
Microcode Studio soporta actualmente los siguientes dispositivos de microcontroladores PIC de las diferentes gamas existentes:

16F627(A), 16F628(A), 16F73, 16F74, 16F76, 16F77, 16F870, 16F871, 16F873(A), 16F874(A), 16F876(A), 16F877(A), 16F87, 16F88, 18F242, 18F248, 18F252, 18F258, 18F442, 18F448, 18F452, 18F458 , 18F1220, 18F1320, 18F2220, 18F2320, 18F4220,

18F4320, 18F6620 and 18F8620 16F627 (A), 16F628 (A), 16F73, 16F74, 16F76, 16F77, 16F870, 16F871, 16F873 (A), 16F874 (A), 16F876 (A), 16F877 (A), 16F87, 16F88, 18F242, 18F248, 18F252, 18F258, 18F442, 18F448, 18F452, 18F458, 18F1220, 18F1320, 18F2220, 18F2320, 18F4220, 18F4320, 18F6620 y el 18F8620 .

### 1.18.5 Configuración de hardware

Microcode Studio ICD no requiere ningún hardware especial para albergar el microcontrolador. Sin embargo, necesitará una tarjeta de acogida que soporta la comunicación serial RS232 o USB. Por ejemplo, un programador de PIC's que es ideal. La figura I.24 muestra como se realizará la comunicación entre el PC y el PIC 16F628A:



*Fig. I.24.- Comunicación PIC – PC*

Microcode Studio ICD se comunica con el microcontrolador utilizando su hardware USART, WinPic, Icprog. Nos debemos remitir a la ficha de datos de los microchips en el microcontrolador que se utiliza.

El CIE colaborará con los microcontroladores que corre a velocidades de reloj diferentes. En la actualidad las velocidades soportadas incluyen 4, 8, 10, 12, 16, 20, 24, 25, 32, 33 y 40 MHz. Sin embargo, es esencial que el código PICBasic tenga el oscilador de ajuste correcto definido. Si el dispositivo de destino se está ejecutando a una velocidad diferente, usted debe asegurarse de la adecuada definición que se utiliza en el código PICBasic. Por ejemplo, si se tiene un microcontrolador funciona a 20MHz, utilice la siguiente declaración al comienzo de su programa:

**DEFINE OSC 20 DEFINE OSC 20**

A falta de establecer el oscilador correcto ajuste ocasiona problemas para la CIE. En resumen, si la configuración en el código no coincide con la velocidad real de su microcontrolador PIC, el ICD no funcionará.

## **1.19. WinPic800**

### **1.19.1 Introducción**

Un microcontrolador es un circuito integrado programable que cada vez más tiene funciones dentro del mismo encapsulado en el que puedes controlar muchos dispositivos como, control de alarmas, cerradura codificada, sirenas de colegios, funciones de TV como el mando a distancia por infrarrojo, motores, encendido y apagados automáticos, microrrobótica, LCD, etc.

En este caso, el más usado para aprender es el popular PIC16F628A. PIC de la firma Microchip significa Peripheral Interface Controller o lo que es lo mismo, Interfaz Control de Periféricos.

### **1.19.2 Configuración**

Para configurar el programador vamos a **Hardware** del menú desplegable **Configuración**.

Si en Selección del Hardware tenemos el **Hga-Prog**, lo seleccionamos y pulsamos el botón Confirmar cambios. De lo contrario:

- En Selección del Hardware elegimos **JDM Programmer**.
- Sacamos la tilde de la casilla de **Bloqueo configuración**, para habilitar el acceso a la configuración.
- Vemos que este marcado la opción **Com**, que es el Puerto de comunicación.
- Ahora en **Data** tildamos la casilla **Inv**, cambiará de color el Estado. Se encenderá el Led Rojo en el Programador.
- Pulsamos el botón **Confirmar cambios**.

La figura I.25 nos muestra la configuración en WinPic800:

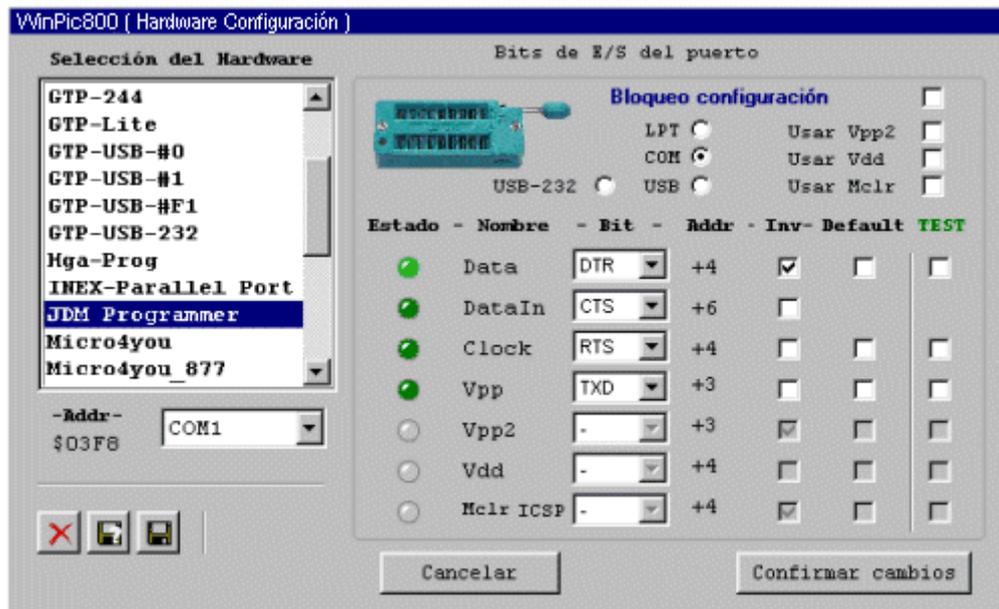


Fig. I.25.- Configuración de WinPic800

### 1.19.3 Grabar Pic con WinPic800

Para grabar un Pic:

- Seleccionar el dispositivo a programar, elegimos **Pic F**, luego el **PIC 18F628A** por ejemplo.
- En el *Programador* seleccionar **Pic**, con el **Sw 1**.
- Abrir el archivo que contiene los datos a programar en el Pic. El programa trabaja con ficheros **.HEX**. En el menú **Archivo** seleccionamos **Abrir archivo**, en el cuadro de diálogo que nos aparece seleccionamos el fichero que deseamos grabar en el Pic.
- Pulsamos el botón **Config**. Para ajustar el tipo de oscilador y los bits de configuración.
- Para programar el dispositivo seleccionamos **Programar todo** del menú **Dispositivo**. Con ello comenzará la grabación.

## **CAPÍTULO II**

### **DISEÑO Y CONSTRUCCIÓN DEL ROBOT**

#### **2.1 Diseño**

Antes de iniciar la construcción del Robot de Batalla se realizó un profundo análisis tomando en cuenta todas las participaciones en los concursos de Robótica con el propósito de conocer los efectos contraproducentes al movimiento de cada una de las partes del robot, además de los efectos causados por el uso de frecuencias similares al momento de controlarlos y de esta manera poder obtener un diseño que nos brinde las facilidades para su manejo, mediante la utilización de materiales acorde a las exigencias de cada uno de los reglamentos que se establecen en dichos concursos y de esta manera obtener el diseño más confiable y para poderlo controlar mediante frecuencia de Bluetooth de 2.4 GHz lo que nos brindará la seguridad de que no tendremos interferencias durante su control en cada una de las batallas que este Robot tenga que enfrentar.

Manteniendo claros estos conceptos, el robot fue dotado con dos armas. La primera consta de una pala que se mueve verticalmente con la ayuda de un cilindro eléctrico controlado por un motor de arranque y la segunda en la parte posterior (Rodillo eléctrico el cual fue crea con el fin de proteger los circuitos de control y de potencia del robot, pero gracias a la fuerza genera por otro motor de arranque sirve como gran ayuda para

la obtención de puntos en los combates), que son justamente los principales movimientos que puede ejecutar el robot para realizar las tareas de ataque y defensa durante la pelea en el ring de batalla, de esta forma se puede generar movimientos de ataque y defensa manteniendo un alto grado de dificultad para que pueda ser atacado por un rival.

Una de las partes fundamentales en la construcción del robot de batalla es la elección o el dimensionamiento de la estructura, la elección de estas medidas no fue tarea fácil, pues existían grandes limitantes como: el costo, limitantes de los motores, tamaño de los elementos, consumo de energía, etc., que representan grandes problemas y complicaciones tanto en la parte electrónica como mecánica.

Tras un breve análisis se puede observar que si se diseña un robot de tamaño natural, la construcción de las partes mecánicas se simplifica, ya que se cuenta con un gran espacio en el cual se puede trabajar y ubicar fácilmente no solamente los actuadores y sensores, sino además todas las placas electrónicas y demás elementos. Por otro lado, el gran tamaño del robot obliga a utilizar motores grandes donde la respuesta será lo suficientemente rápida para poder operar y efectuar los movimientos; además, el costo y la demanda de energía que se necesitaría para poder generar movimiento y la utilización de sus armas serían elevados, razón por la cual, se tomo la decisión de utilizar materiales reciclados en su mayoría para la construcción lo cual abarata los costos y nos permite la utilización de mejores elementos en partes del robot que se consideran indispensables.

Teniendo presente todo ello y algunas consideraciones más, se optó por construir un robot cuyas dimensiones giren alrededor de  $1 \text{ m}^3$ .

Para la construcción del robot de batalla se decidió diseñar parte por parte su estructura debido al peso que soportará para lo cual se diseño una estructura resistente, tanto al peso y a los golpes que recibirá en batalla, además que dicha estructura brindara la facilidad para poder ubicar dentro de ella todos los circuitos eléctricos, baterías, cajas de engranaje, etc., de tal manera que su ubicación sea de una manera compacta, pero con la facilidad de poder tener al alcance cada uno de los circuitos y motores en caso de una avería para poder dar solución de una manera muy rápida y eficiencia a los desperfectos que se presenten durante una batalla debido a los escasos minutos de tiempo existentes para la solución de dichos problemas.

### 2.1.1. Diseño Inferior de la Estructura del Robot

Para el diseño total de la estructura se dispone de  $1\text{m}^3$ , de los cuales se ha tomado como referencia un área de  $56.5\text{cm}$  de Ancho \*  $97\text{cm}$  de Largo con armas extendidas, debido a que el robot dispondrá de armas las cuales estarán desplegadas de la estructura central.

Para el diseño inferior se ubico 4 ruedas (1,2,3,4) al interior del robot para tener seguridad contra armas corto punzantes que se encuentren ubicados en otros robots, teniendo en cuenta esto y el área que se dispone además de conocer que el área necesaria para las ruedas que ubicaremos es de  $12\text{cm}$  de ancho y dependiendo de la ubicación se dará su largo (largo de rueda delantera  $32\text{cm}$  y el largo de la rueda trasera  $23\text{cm}$ ), este diseño esta realizado de la siguiente manera el cual se muestra en la figura II.26, del cual dispondremos de un área 2 la cual nos servirá para poder ubicar cada uno de los circuitos eléctricos, baterías y el espacio necesario para poder realizar todo su cableado:

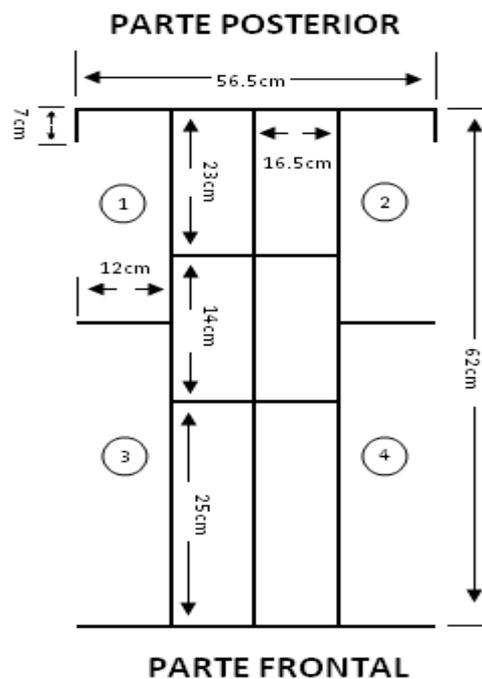


Fig. II.26.- Vista inferior de la estructura del Robot

### **2.1.2. Diseño Superior de la Estructura**

La parte superior de la estructura esta realizada en tres niveles con dos inclinaciones (la primera a  $30^{\circ}$  y la segunda a  $85^{\circ}$  la cual es netamente estética para el robot) debido a la ubicación de cada uno de los componentes eléctrico-electrónicos y de motricidad del robot.

El primer nivel servirá para la ubicación de los motores 1 y 2 los cuales serán utilizados para el movimiento de las 2 armas del robot y además servirá para la ubicación de los circuitos de control del robot en su totalidad.

Para lo cual se dispone de un área de 30cm de ancho \* 56.5 de largo debido a que los motores pueden estar ubicados en la parte posterior fijados a la estructura sin provocar ningún tipo de inconveniente con el movimiento de las ruedas traseras del robot. Este nivel se encuentra diseñado de tal manera que su área es una sola a diferencia que los niveles 2 y 3 los cuales por disposición de cilindros eléctricos y baterías tiene una abertura central.

El segundo nivel esta a un ángulo de  $30^{\circ}$  con referencia al primer nivel el cual se lo realizo con el propósito de proteger las ruedas delanteras y las baterías que generan la corriente y voltaje necesario para la alimentación del circuito de potencia que nos da la movilidad del robot y además genera corriente para el funcionamiento de las armas del robot. Este nivel posee un área de 56.5cm de largo por 25cm de ancho.

El tercer nivel es el que posee una mayor inclinación debido a cuestiones netamente estéticas del robot para lo cual se dispuso darle una inclinación de  $85^{\circ}$ .

Como se dijo anteriormente los niveles 2 y 3 tienen una apertura central de 5 cm la cual nos ayuda para un mejor manejo de las baterías en caso de sustitución por descarga o avería además de poder controlar el funcionamiento del cilindro eléctrico el cual es el arma principal. Este nivel posee un área de 56.5cm de largo por 5cm de ancho.

La figura II.27 muestra la disposición superior de la estructura:

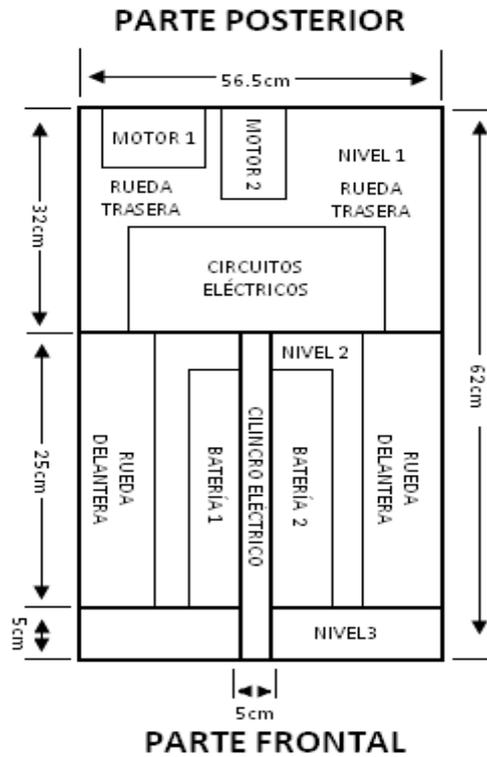


Fig. II.27.- Vista superior de la estructura del Robot de Batalla

### 2.1.3. Diseño Lateral de la Estructura

Para el diseño de la parte lateral del robot de batalla se tomará relación del largo, ancho y altura de cada uno de los niveles que se realizaron tanto por estética como para la ubicación de los componentes del robot. El diseño lateral se muestra a continuación en la figura II.28:

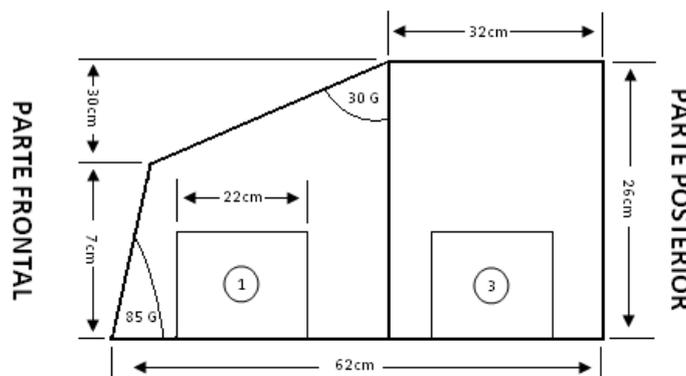


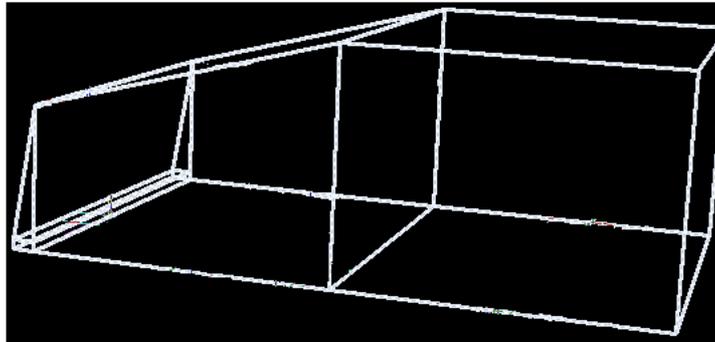
Fig. II.28.- Vista Lateral de la estructura del Robot de Batalla

#### 2.1.4. Modelo Final de la Estructura

Después el diseño de las diferentes partes del robot en una estructura de vistas llegamos al diseño final el cual servirá como base para la construcción de cada uno de los diseños de cada parte del robot.

La estructura está realizada en base de tubo rectangular estructural de 1\*1/2 pulgadas y para recubrir la estructura se utilizo diferentes tipos de lata de al menos 2mm de espesor los cuales se los obtuvo de chatarra.

En la figura II.29 se puede visualizar como quedara la estructura del robot:



*Fig. II.29.-Estructura armada del Robot de Batalla en 3D*

El diseño se lo realizo pensando en la seguridad de cada uno de los componentes que forman parte del robot de tal forma que cada componente quede dentro de este de la manera más compacta posible, pero al mismo tiempo que brinda la posibilidad de llegar de una manera muy sencilla a cada uno de sus componentes en caso de averías o de sustitución de baterías, cambios que son muy frecuentes durante una batalla.

#### 2.1.5. Diseño de Movimiento

Para llegar al diseño final se ha hecho un análisis de todas las participaciones en concursos anteriores, por tal motivo se llego a la conclusión que la manera adecuada de mover el peso con una fuerza y una velocidad adecuada es proporcionando tracción en las 4 ruedas del robot.

El movimiento del robot estará dado por motores de corriente continua de 12V - 6A con cajas de engranajes los cuales le proporcionan la fuerza adecuada para poder mover los

50Kg que pesará aproximadamente dicho robot y además proporcionen la tracción necesaria para mover a robots rivales dentro de una batalla.

Consiguiendo movimientos que sean lo más coordinado posibles y de esta forma lograr obtener precisión, fiabilidad y seguridad al momento de tener una participación en cualquier concurso que deba presentarse. Las cajas a utilizar se presentan en la siguiente figura II.30:



Fig. II.30.- Motor de CC con caja de engranaje del robot

La disposición de estos motores con caja de engranaje dentro de la estructura del robot se muestra a continuación en la figura II.31, para lo cual el área de ubicación de llantas, cajas de engranaje y motores es de  $22\text{cm}^2$ , ubicados de tal manera que se compactan y dejan el mayor espacio posible para que se puedan ubicar los demás componentes eléctrico-electrónicos dentro del robot:

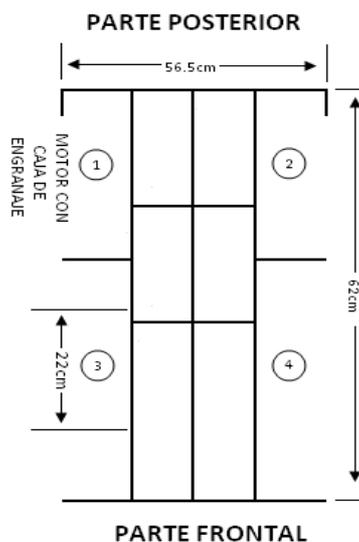
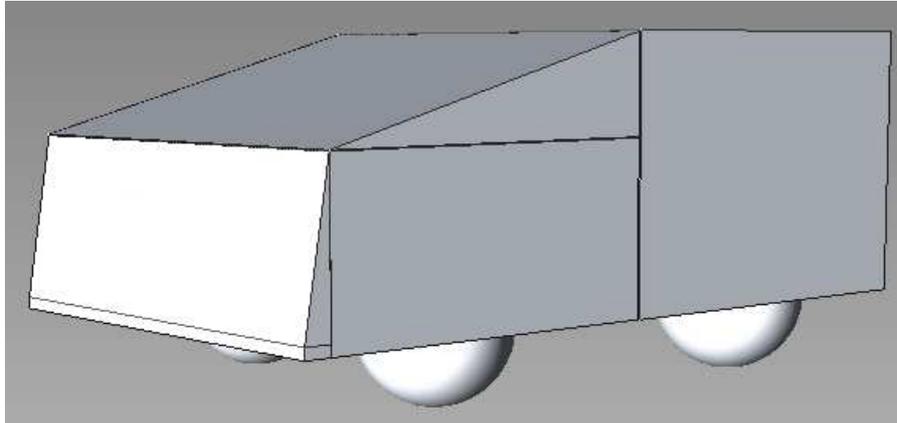


Fig. II.31- Disposición de los Motores dentro de la Estructura

### 2.1.6. Diseño en 3D de la Estructura

En la figura II.32 se muestra una imagen tridimensional de cómo se verá el robot de batalla con sus ruedas ya ubicadas. Diseño al cual se debe llegar por aspectos de seguridad y ubicación de componentes ya citados anteriormente:



*Fig. II.32.- Diseño en 3D del robot de Batalla con sus ruedas*

### 2.1.7. Diseño Mecánico de Armas

Para la elección de las armas a utilizar se tomo como referencia armas ya realizadas anteriormente, como el rodillo, el cual será nuestra arma secundaria y se encontrara ubicada en la parte posterior del robot. Para la elección del principal se considero el desempeño que brindan los cilindros eléctricos, en comparación con cilindros neumáticos ya utilizados en anteriores proyectos los cuales además de llevar un altísimo costo, espacios muy grandes y además necesita de herramientas externas como: Compresor de alta presión (150psi mínimo), tanque de almacenamiento de aire (de 100psi como mínimo), y una electroválvula para su funcionamiento eléctrico.

Mientras que con la utilización de un cilindro eléctrico solamente es necesario un motor de arranque de moto, y un cable de acero especial con el cual se proporciona la fuerza necesaria para levantar un peso de más de 20kg con la corriente proporcionada por una batería de 12A, los cuales se los puede ubicar de una manera mucho más sencilla dentro de la estructura diseñada.

### 2.1.7.1. Pala Eléctrica

La Pala Eléctrica está diseñada para poder levantar aproximadamente 50Kg, La pala eléctrica estará ubicada en la parte frontal del robot, su diseño esta realizado para que abarque solamente una parte de un robot, de esta manera se consigue que el robot levante un máximo de 30Kg, con esto se maximiza el uso de energía al momento de levantar cualquier tipo de peso. Su funcionamiento se basa en el movimiento de un motor de arranque acoplado con un cable de acero al cilindro eléctrico el cual funciona al encenderse el motor y mediante un circuito de control se puede controlar la subida y bajada del mecanismo de levantamiento.

De tal forma que dicho mecanismo realiza el levantamiento con el menor esfuerzo posible y minimizando el consumo de corriente logrando realizar más repeticiones en el lapso de la batalla.

Cabe recalcar que para la construcción de este mecanismo solamente se usaron materiales reciclados los cuales se detallan a continuación:

- Un cilindro eléctrico
- Un motor de arranque
- Un cable de acero especial
- Láminas de acero de 5mm de espesor por 4 cm de ancho
- Un eje de acero de 2cm
- 2 Rodamientos de 5 cm

El motor de arranque a utilizar se muestra en la figura II.33:



*Fig. II.33.- Motor de Arranque*

El motor de arranque con un cilindraje de 200cc, nos proporciona un aproximado de 500 RPM lo suficiente para poder levantar un peso de 35 Kg, siempre y cuando tenga un cilindro que soporte dicho cilindraje.

El diseño de la pala eléctrica se muestra a continuación en la figura II.34:

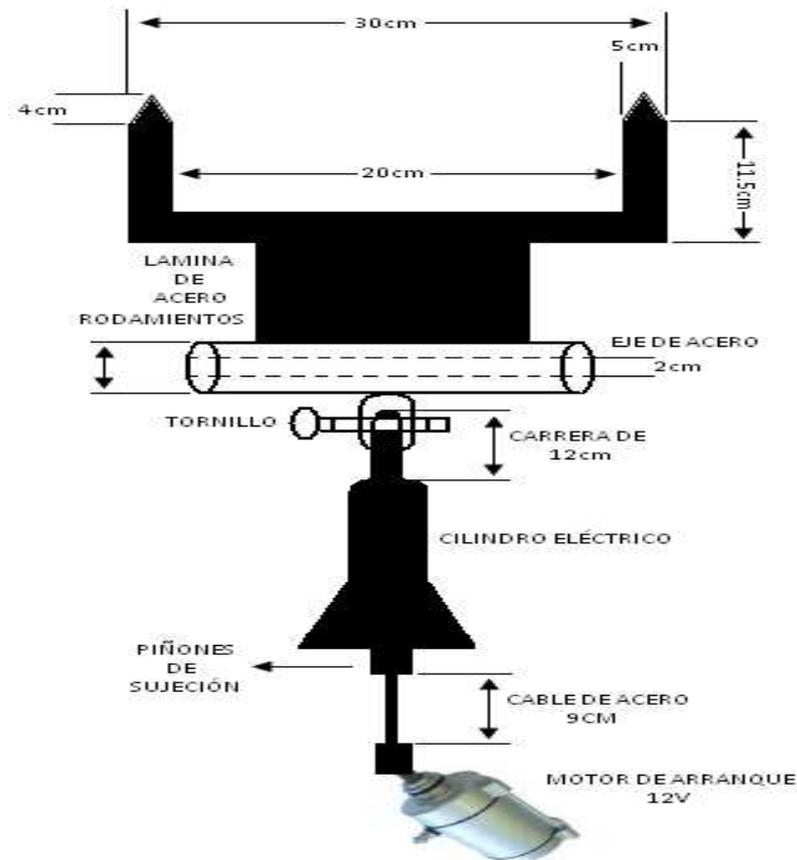


Fig. II.34- Diseño de la Pala Eléctrica

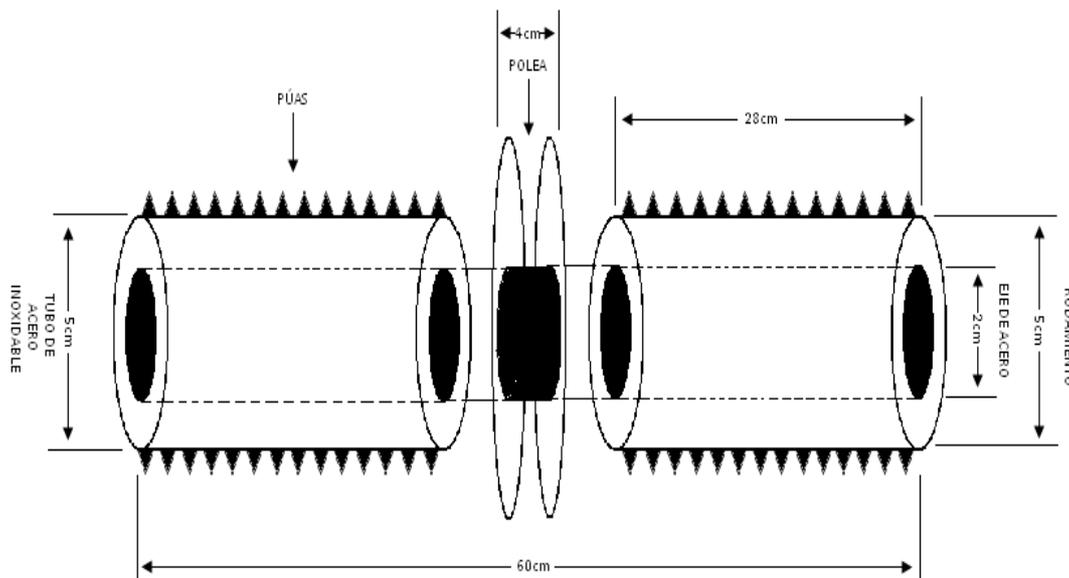
### 2.1.7.2. Rodillo Eléctrico

El rodillo eléctrico está diseñado a partir de una polea central y dos soportes colocados a los extremos de la estructura. La parte fundamental del funcionamiento de dicho rodillo se basa en 4 rodamientos colocados dentro de un tubo de acero el cual se divide en 2 partes y unidos entre sí por la polea central que mediante la utilización de una banda de rodamiento se acopla a un motor de arranque el cual al girar brinda el movimiento de dicho rodillo.

Los materiales utilizar son:

- Motor de arranque
- Bandas de rodamiento
- Púas de acero
- Tubo de acero inoxidable de 4cm
- Eje de acero de 2cm
- 2 Poleas de 7cm
- Pernos

El diseño del rodillo eléctrico se muestra a continuación en la figura II.35:



*Fig. II.35.- Diseño Del Rodillo Eléctrico*

### 2.1.8. Diseño Eléctrico-Electrónico

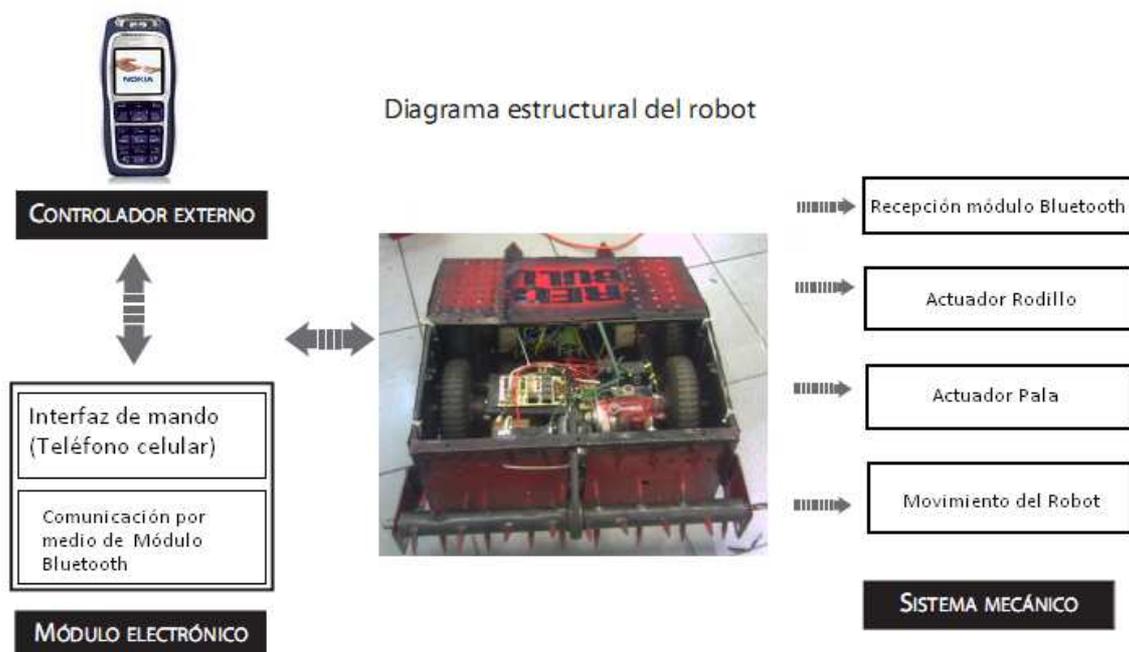
El Robot de Batalla consta de diferentes etapas lo que permite tener separado todo lo referente a control y comunicación de la etapa de potencia debido a los diferentes consumos de corriente y las corrientes inversas que soportan los circuitos de potencia:

- La primera etapa denominada de Control ya que se encarga de la etapa de potencia que es la encargada de generar el movimiento del robot y se encarga de recibir los datos provenientes del módulo Bluetooth y codificarlos de tal manera que pueda realizar solamente las instrucciones específicas que se le dan al robot de batalla,

además de realizar el control de las armas de que se encuentran como defensa y ataque del robot de batalla.

- La segunda etapa es la de potencia que se encarga de controlar cada uno de los movimientos del motor mediante la utilización de un puente H.
- Y la tercera etapa una de las más importantes la cual dará la movilidad de las armas del robot de batalla mediante la utilización de un puente H para controlar la Pala Eléctrica y un relé para el control del Rodillo Eléctrico.

La figura II.36 muestra un diagrama de la estructura mecánica acoplada con la parte electrónica, además de las armas que posee el Robot de Batalla:



*Fig. II.36.- Estructura del Robot de Batalla*

### 2.1.8.1. Diseño de la Etapa de Control de Potencia

Para realizar el diseño de la etapa de control de potencia se tomo en cuenta lo que se desea controlar con esta etapa, de qué manera se va hacer y de acuerdo a estos dos aspectos realizar los diseños debido a que está etapa define el movimiento del robot y el control de sus armas que son las etapas más importante que se debe tener en cuenta en un robot de batalla.

La figura II.37 muestra el diseño completo de la etapa de Control de Potencia, la cual se explica en sub-etapas debido a que este es un circuito electrónico muy completo que realiza tareas complejas para que el robot de batalla funcione de una manera óptima y muy fiable.

Diseño que en su totalidad fue realizado en PROTEUS ISIS PROFESIONAL debido a su facilidad de manejo y sus grandes prestaciones, ya que se encuentran todos los elementos electrónicos para el diseño previo:

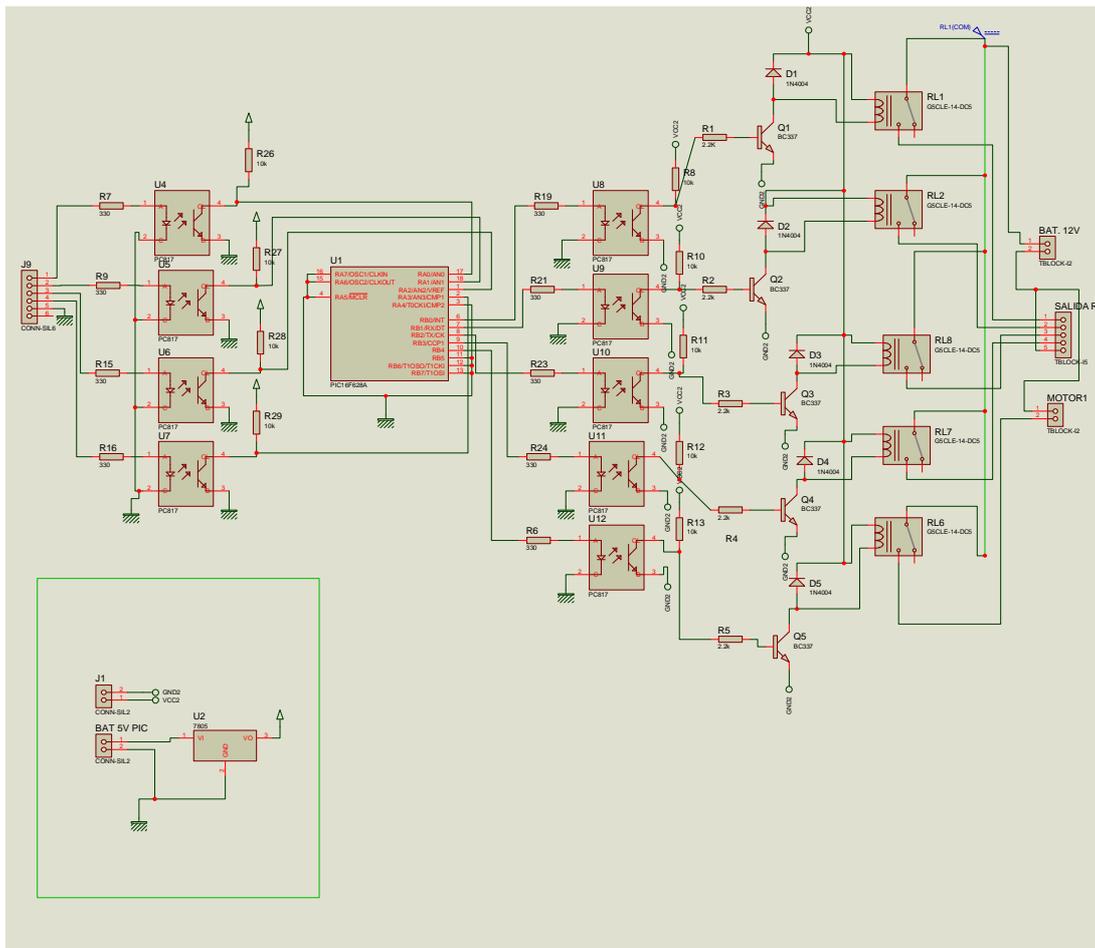


Fig. II.37. Diseño de la etapa de Control de Potencia

Los elementos electrónicos utilizados en esta etapa son los siguientes:

- Optoacopladores PC817
- Resistencias
- Sockets
- Pic 16F628A

- Regulador de 5V
- Transistores 2N3904
- Relés de 12V-1A

Lo primero que se tomo en cuenta es la construcción de una fuente para el Pic 16F628A el cual se alimenta con un voltaje entre 4.5V a 5.5V, debido a que su funcionamiento es primordial en el diseño de la etapa de control de potencia del robot de batalla y el cual debe tener un voltaje nominal de 5v aproximadamente para poder trabajar de una manera óptima para lo cual utiliza un regulador de 5V, para obtener el voltaje deseado.

La figura II.38 nos muestra el diseño de la fuente de 5V para el Pic 16F628A:

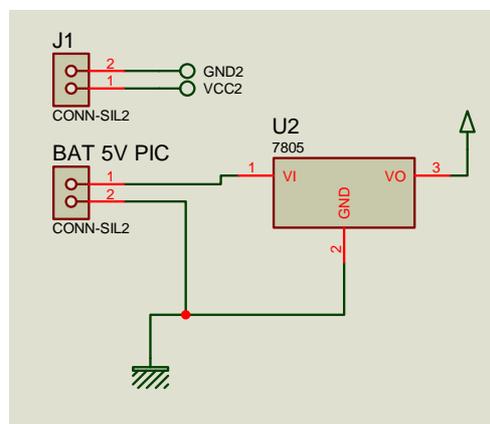


Fig. II.38.- Diseño de la fuente para el Pic 16F628A

La entrada del circuito de control de potencia estará aislada mediante cuatro Optoacopladores de la serie PC817 los cuales son dispositivos de emisión y recepción de luz que funciona como un interruptor excitado mediante la luz. La mencionada luz es emitida por un diodo LED que satura un componente optoelectrónico, normalmente en forma de fototransistor. De este modo se combinan en un solo dispositivo semiconductor, un fotoemisor y un fotorreceptor cuya conexión entre ambos es óptica. Los que se utilizan como medio de protección para dispositivos muy sensibles en este caso el Pic 16F628A.

La manera de protección se da si la tensión de entrada varía, la cantidad de luz también lo hará, lo que significa que la tensión de salida cambia de acuerdo con la tensión de entrada. De este modo el dispositivo puede acoplar una señal de entrada con el circuito de salida.

La ventaja fundamental de un optoacoplador es el aislamiento eléctrico entre los circuitos de entrada y salida. Mediante el optoacoplador, el único contacto entre ambos circuitos es un haz de luz. Esto se traduce en una resistencia de aislamiento entre los dos circuitos del orden de miles de  $M\Omega$ .

La figura II.39, nos muestra la disposición de los 4 Optoacopladores para la protección que se dará a las entradas del Pic:

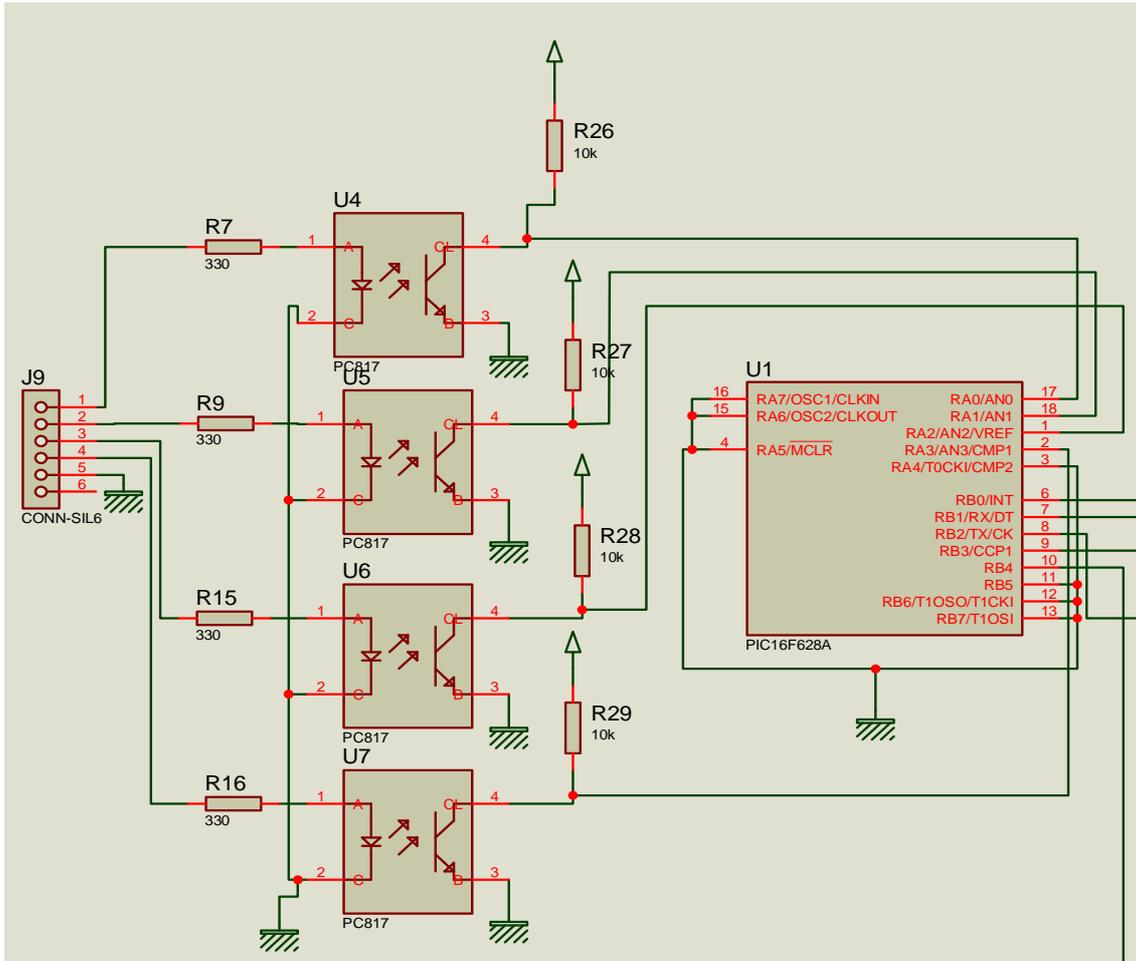


Fig. II.39.- Ubicación de Optoacopladores a la entrada del Pic 16F628A

De la misma forma se ubica cinco Optoacopladores a la salida del Pic con el fin de proteger de corrientes inversas producidas por la activación de los relés de potencia.

La figura II.40, nos muestra la disposición de los 5 Optoacopladores para la protección que se dará a las salidas del Pic:

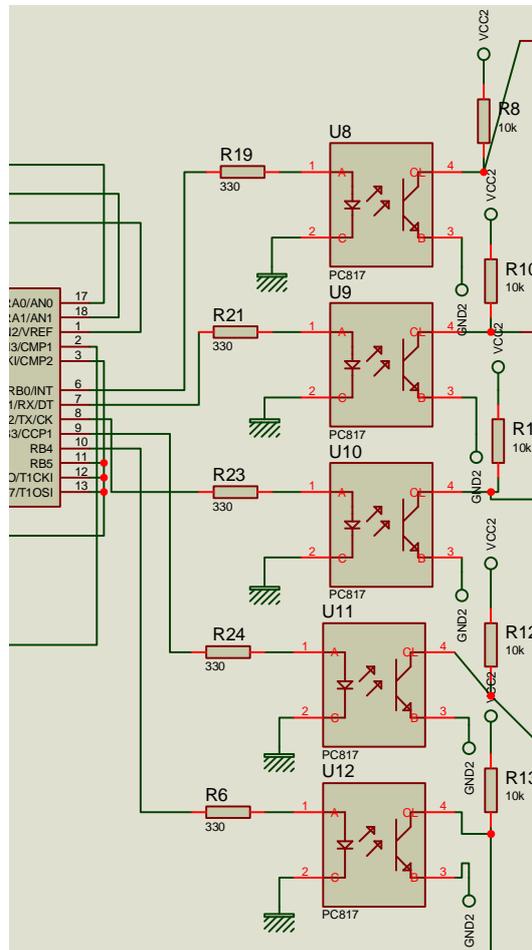


Fig. II.40.- Ubicación de Optoacopladores a la salida del Pic 16F628A

Las datos de las salidas son enviadas en forma de tabla de verdad de “1” (5V) y “0” (0V), pasando por los Optoacopladores hacia transistores 2N3904 que son los encargados de la activación de cinco relés (12V, 2A), que a su vez se encargan de la activación del puente H que posee relés de potencia (12V, 30A), los cuales se encargan de mover los motores de para el movimiento del robot de batalla.

Pero también mediante esta tabla de verdad se puede controlar el segundo Puente H que se encarga de mover el motor de arranque que da movimiento a la Pala eléctrica.

De una manera que no haya ningún tipo de interferencia entre el manejo de los movimientos del robot con el movimiento de la Pala Eléctrica.

La figura II.41 muestra el esquema final de los transistores que controlan la activación de los relés de la etapa de potencia:

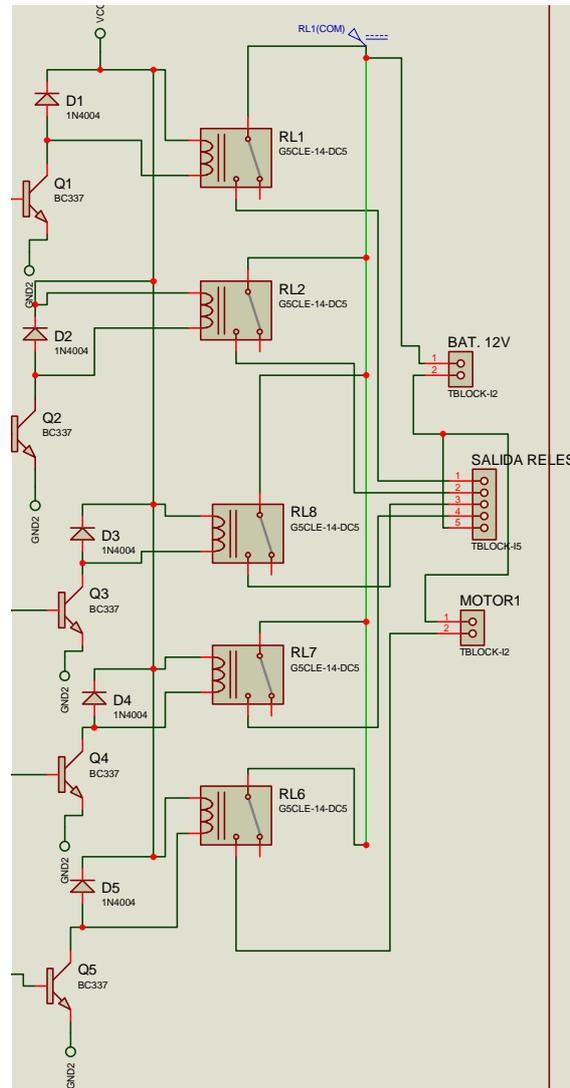


Fig. II.41.- Sub-Etapa de Activación de la Etapa de Potencia

### 2.1.8.2. Diseño de la Etapa de Potencia

Para poder mover los motores con cajas de engranajes se realizó un Puente H con relés de 12V a 30 A. Sabiendo que un Puente H es un circuito electrónico que permite a un motor eléctrico de DC girar en ambos sentidos, *avance* y *retroceso*. Los cuales son ampliamente usados en robótica y como convertidores de potencia. Los puentes H están disponibles como circuitos integrados, pero el Puente H encapsulado que soporta más corriente es de un máximo de 6A y debido al alto consumo de corriente que presentan los motores con caja de engranaje llegando a alcanzar consumos mayores a los 12 A estos puentes encapsulados no, nos serán de ayuda para el control de los motores del robot,

razón por la cual se diseñó un Puente H con relés que soportan un alto consumo de corriente llegando a soportar hasta un máximo de 30A.

El diseño se lo hará de tal manera que se lo pueda realizar en una baquelita, pero debido al alto consumo de corriente las conexiones serán reforzadas con cable de conexiones eléctricas número 10, ya que al producirse un alto consumo de corriente se eleva la temperatura del cable y si es muy delgada podría llegar a fundirse y con mucha facilidad producir un gran cortocircuito.

La placa estará diseñada para poder ubicar terminales eléctricos los cuales servirán para poder colocar los relés de 12v. La realización en la placa tiene como fin ayudar a la estética del robot permitiendo la ubicación más sencilla y de una manera más compacta dentro del robot de batalla.

La figura II.42 muestra el diseño en PROTEUS ISIS PROFESIONAL de la placa para poder ubicar los relés en las respectivas terminales:

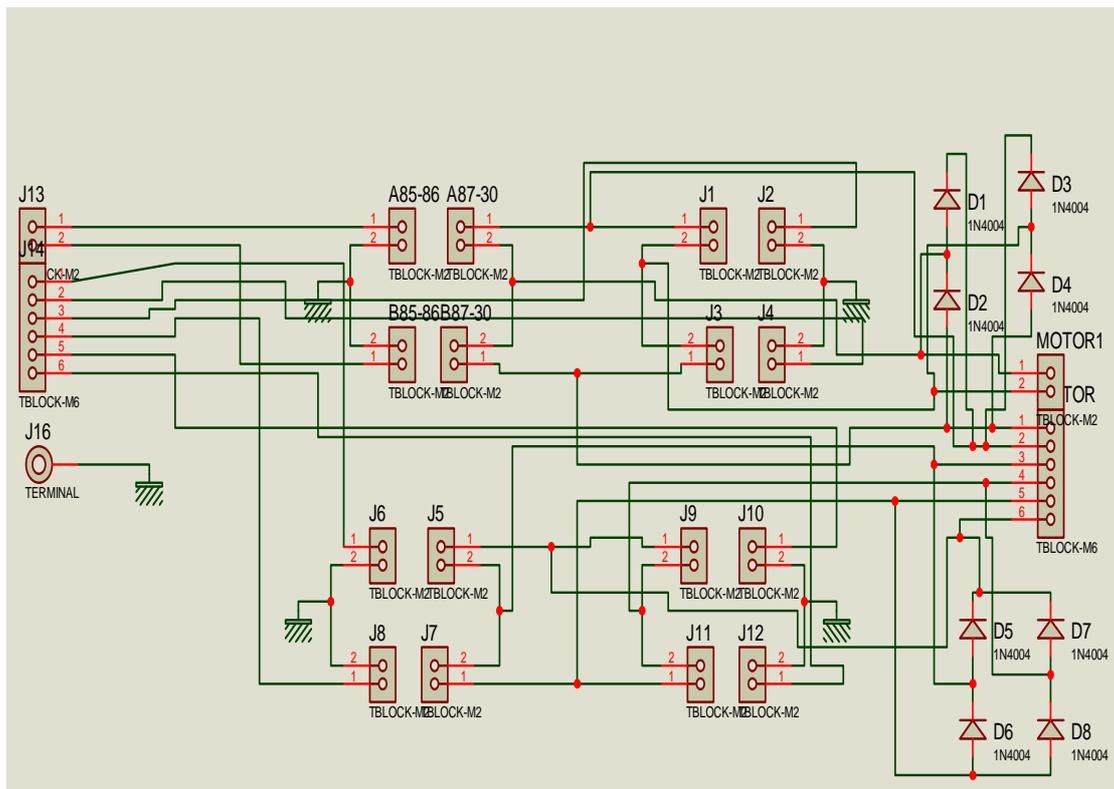


Fig.II.42.- Puente H

Los materiales a utilizar para la construcción del puente H son los siguientes:

- Relés de 12V-30A

- Terminales eléctricas
- Diodos 1N4004
- Sockets
- Cable de cobre número 10 de conexiones eléctricas

Además este Puente H tiene un circuito de protección en caso de cortocircuito el cual está diseñado con diodos los cuales en caso de existir algún cortocircuito los diodos abren el circuito y con esto no permiten el paso de corriente lo cual brinda la seguridad para que los relés no sufran ningún daño.

La figura II.43 muestra la etapa de protección diseñada en caso de cortocircuitos en base de diodos comunes 1N4004:

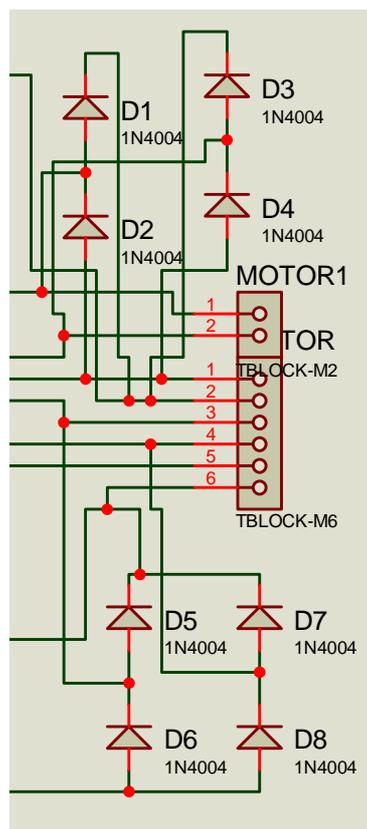


Fig. II.43.- Protección con diodos del Puente H

### 2.1.8.3. Diseño de la Etapa de Control de Armas

Para el diseño de la etapa de control de armas se tomo en cuenta el tipo de armas que se utilizarán que en nuestro caso serán dos la Pala Eléctrica y el Rodillo Eléctrico, los cuales para su operación tienen circuitos independientes debido a que su forma de trabajo es completamente independiente el uno del otro.

Razón por la cual se brinda la debida explicación del funcionamiento de cada una de estas armas y como fueron diseñados sus circuitos de control.

### 2.1.8.4. Pala

El funcionamiento de la Pala se lo produce mediante la activación de un motor de arranque (M1), el cual está conectado a un cilindro eléctrico y este a su vez a un mecanismo de levantamiento. Razón por la cual para un correcto funcionamiento de la Pala el motor de arranque debe ser accionado por un Puente H2 ya que este solamente controlará un motor de arranque (M2) y será solo para que el mecanismo pueda levantar y luego bajar la pala mediante dos pulsos recibidos de la placa de control de potencia.

Pero como en la placa anterior también estará diseñada con circuitos de activación y desactivación mediante la utilización de dos optoacopladores.

La figura II.44 muestra el diseño de la placa de control de la Pala:

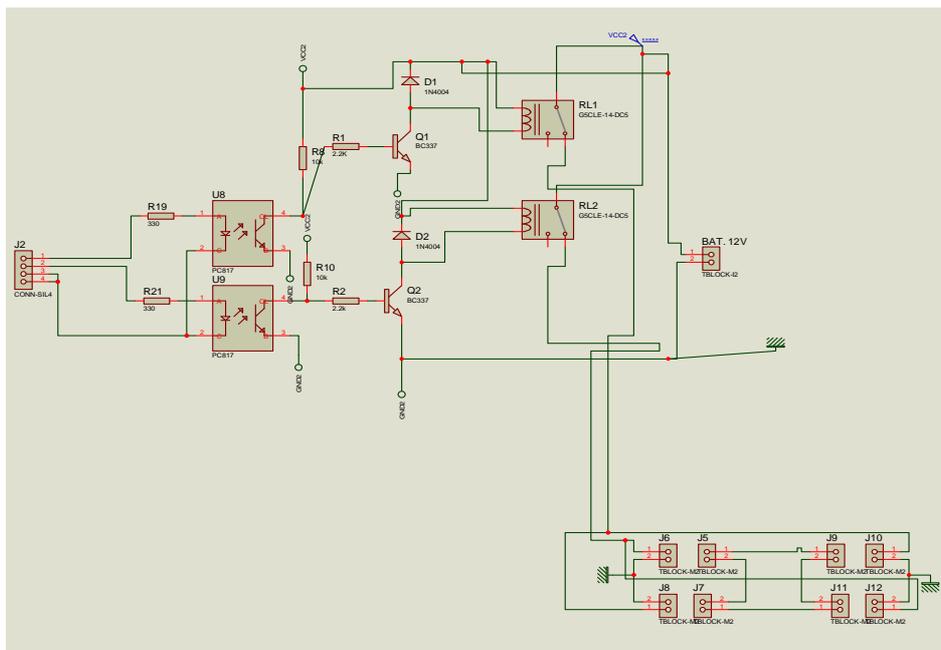


Fig. II.44.- Circuito de control de la Pala

Los materiales a utilizar para la construcción del circuito de control de la pala eléctrica son los siguientes:

- Optoacopladores PC817
- Resistencias
- Transistores 2N3904
- Relés de 12V-1A
- Relés de 12V-30A
- Sockets

### 2.1.8.5. Rodillo

El funcionamiento del rodillo se base en un motor de arranque M2, el cual al momento de encenderse mediante pulso enviado desde la placa de control de potencia consigue girar con la velocidad proporcionada por el motor. Razón por la cual no se necesita de un Puente H para activarse solamente un relé de 12V a 30A para activar el motor y el giro se hará de acuerdo a la necesidad cambiando su polaridad de forma manual.

Consiguiendo tener un circuito bastante sencillo el cual se ubica en la parte superior del Puente H de control de la Pala, como se muestra en la figura II.45:

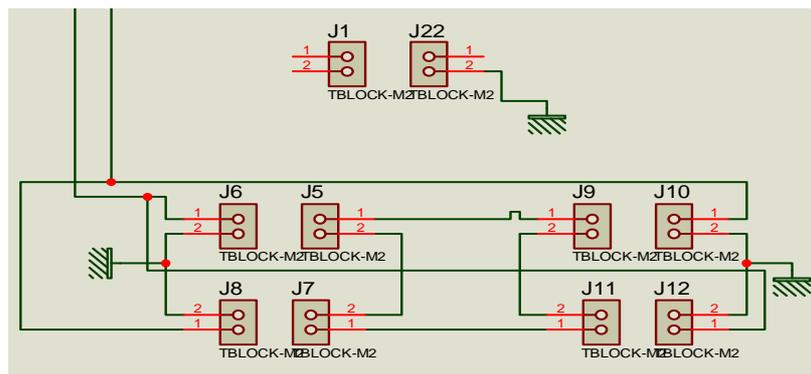


Fig. II.45.- Circuito de control del Rodillo

Para el diseño del circuito de control del rodillo se necesitaron los siguientes materiales:

- Relé de 12V-30A
- Optoacopladores PC817

- Transistores 2N3904
- Sockets

## 2.2. Diagrama de Flujo de Operación del PIC 16F628A

Para el funcionamiento del PIC se diseñó un diagrama de flujo de acuerdo con las necesidades del robot de batalla para poder realizar un programa que permita cumplir con las exigencias de este. La figura II.46 muestra el diagrama de flujo que será el que ayude a la programación del PIC de acuerdo a todas las instrucciones que deberá cumplir el robot de batalla:

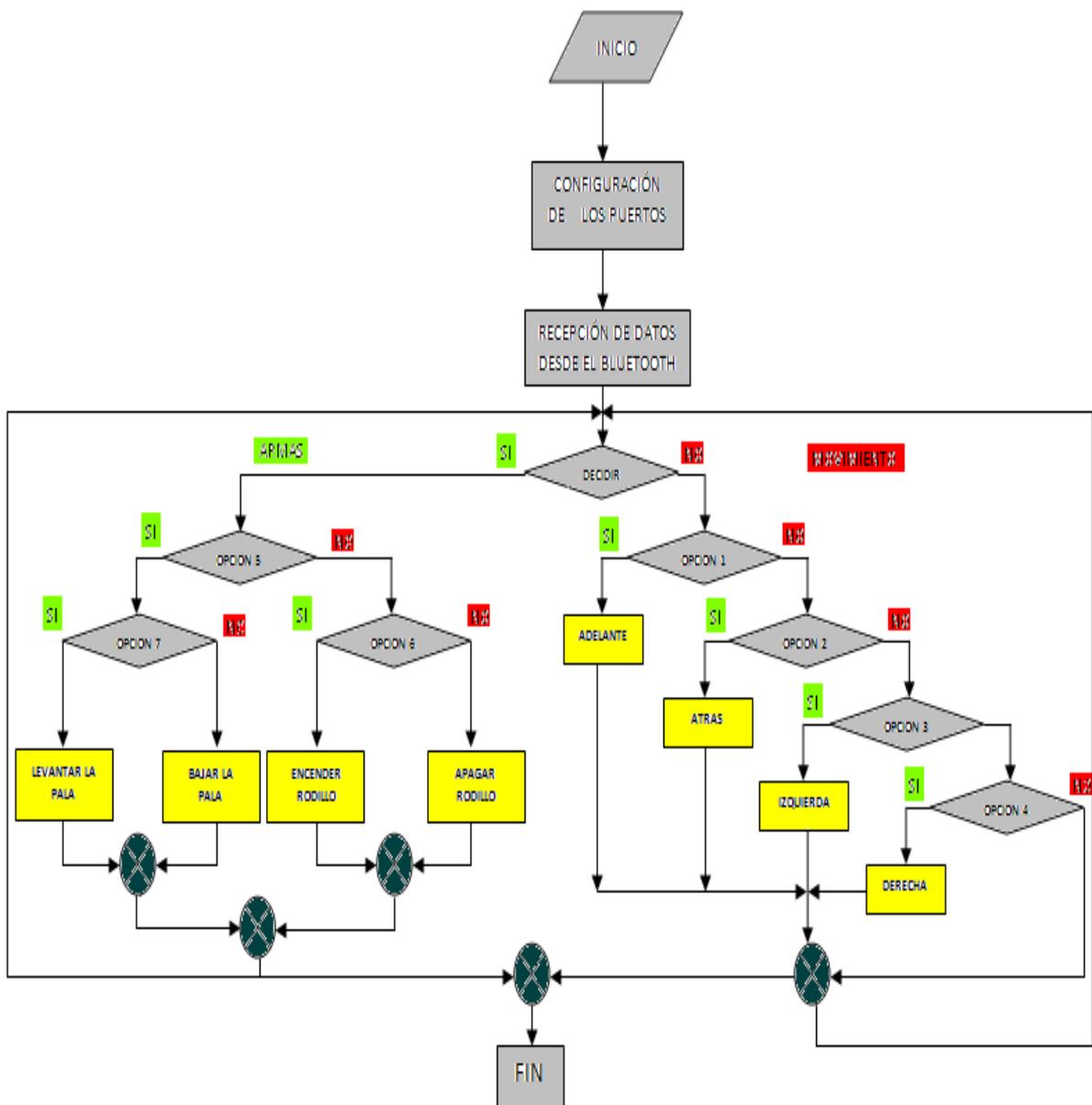


Fig. II.46.- Diagrama de flujo de funcionamiento del Pic 16F628A

Antes del funcionamiento del programa se debe configurar cada uno de los puertos del Pic debido a que hay que definir cuáles serán de entrada y de salida de acuerdo al diseño realizado anteriormente, para de esta manera poder recibir los datos que serán enviados desde el Bluetooth y de esta manera decidir qué es lo que hará el robot:

Para el movimiento hay que realizar las instrucciones para poder controlar cada uno de los motores para saber qué movimiento hay que realizar de acuerdo a una tabla de verdad de los datos recibidos desde el Bluetooth y de esta manera controlar los giros del robot de batalla dependiendo del pulso recibido se puede ir hacia adelante, atrás, izquierda o derecha, lo que facilita las maniobras de ataque o defensa.

Si la decisión son las armas sea esta el rodillo o la pala, siempre tratando de optimizar el uso de los recursos debido a que el consumo de corriente es bastante alto por lo que la descarga de las baterías puede ser muy rápida si no se tiene cuidado con la elección en el momento adecuado.

Cuando la elección sea la pala se puede elegir entre subir la pala mediante el accionamiento del motor de arranque M1, de tal forma que se pueda quedar elevado y poder realizar el movimiento del robot con la pala levantada o con un pulso inmediato dar reversa al motor y regresarlo a su posición original.

Cuando la elección sea el rodillo se puede tomar la decisión de encender y luego mover el robot de batalla o solo encender el rodillo y a su vez apagarlo independientemente de que este o no el robot en movimiento.

### **2.3. Programa para Control del PIC 16F628A**

Para el funcionamiento del circuito de Control de la Etapa de Potencia se realizara un programa en el lenguaje de programación Microcode Studio, el cual ayuda a controlar el correcto funcionamiento de la etapa de potencia, además del funcionamiento de la Pala Eléctrica.

Utilizando el código realizado en Microcode Studio para el funcionamiento de acuerdo con el diagrama de flujo de su funcionamiento es el siguiente:

```
LIST P=16F628A
RADIX HEX
ESTADO equ 0x03
PORTA equ 0x05
```

```

PORTB equ 0x06
CONFPA equ 0x05
CONFPB equ 0x06
CMCON equ 0x1F
PDel2 equ 0x20
PDel3 equ 0x21
PDel4 equ 0x22
VAR EQU 0X23
org 0x00
RESET
GOTO INI
org 0X05
INI
CLRF PORTA
MOVLW 0X07
MOVWF CMCON
BSF ESTADO,5
MOVLW .255
MOVWF CONFPA
MOVLW .128
MOVWF CONFPB
BCF ESTADO,5
CLRF PORTB
INIC
MOVLW B'00001111'; MÁSCARA
ANDWF PORTA,0
XORLW .5
BTFSK ESTADO,2
CALL ADELANTE
MOVLW B'00001111'; MÁSCARA
ANDWF PORTA,0
XORLW .9
BTFSK ESTADO,2
CALL IZQUIERDA
MOVLW B'00001111'; MÁSCARA
ANDWF PORTA,0
XORLW .6
BTFSK ESTADO,2
CALL DERECHA
MOVLW B'00001111'; MÁSCARA
ANDWF PORTA,0
XORLW .10
BTFSK ESTADO,2
CALL ATRAS
MOVLW B'00001111'; MÁSCARA
ANDWF PORTA,0
XORLW .1
BTFSK ESTADO,2
CALL SUBIR_PALA
MOVLW B'00001111'; MÁSCARA

```

```
ANDWF PORTA,0
XORLW .2
BTFSC ESTADO,2
CALL BAJAR_PALA
MOVLW B'00001111'; MÁSCARA
ANDWF PORTA,0
XORLW .11
BTFSC ESTADO,2
CALL ON_RODILLO
MOVLW B'00001111'; MÁSCARA
ANDWF PORTA,0
XORLW .7
BTFSC ESTADO,2
CALL OFF_RODILLO
MOVLW B'00001111'; MÁSCARA
ANDWF PORTA,0
XORLW .15
BTFSC ESTADO,2
CALL STOP
GOTO INIC
SUBIR_PALA
BSF PORTB,5
BCF PORTB,6
CALL RETARDO1
BSF PORTB,5
BSF PORTB,6
RETURN
BAJAR_PALA
BCF PORTB,5
BSF PORTB,6
CALL RETARDO1
BSF PORTB,5
BSF PORTB,6
RETURN
ON_RODILLO
BCF PORTB,4
RETURN
OFF_R
BSF PORTB,4
RETURN
ADELANTE
MOVF PORTB,0
MOVWF VAR
MOVLW 0X69;0X09
MOVWF PORTB
BTFSC VAR,4
BSF PORTB,4
RETURN
IZQUIERDA
MOVF PORTB,0
```

```

MOVWF VAR
MOVLW 0X6A;0X0A
MOVWF PORTB
BTFSC VAR,4
BSF PORTB,4
RETURN
DERECHA
MOVF PORTB,0
MOVWF VAR
MOVLW 0X65;0X05
MOVWF PORTB
BTFSC VAR,4
BSF PORTB,4
RETURN
ATRAS
MOVF PORTB,0
MOVWF VAR
MOVLW 0X66;0X06
MOVWF PORTB
BTFSC VAR,4
BSF PORTB,4
RETURN
STOP
MOVF PORTB,0
MOVWF VAR
MOVLW 0X6F
MOVWF PORTB
BTFSC VAR,4
BSF PORTB,4
RETURN
RETARDO1 movlw .22 ; 1 set numero de repetición (C)
movwf PDel2 ; 1 |
PLoop01 movlw .30 ; 1 set numero de repetición (B)
movwf PDel3 ; 1 |
PLoop11 movlw .200 ; 1 set numero de repetición (A)
movwf PDel4 ; 1 |
PLoop21 clrwdt ; 1 clear watchdog
decfsz PDel4, 1 ; 1 + (1) es el tiempo 0 ? (A)
goto PLoop21 ; 2 no, loop
decfsz PDel3, 1 ; 1 + (1) es el tiempo 0 ? (B)
goto PLoop11 ; 2 no, loop
decfsz PDel2, 1 ; 1 + (1) es el tiempo 0 ? (C)
goto PLoop01 ; 2 no, loop
PDelL1 goto PDelL2 ; 2 ciclos delay
PDelL2 clrwdt ; 1 ciclo delay
return
END

```

## 2.4. Implementación de la Etapa de Potencia

La etapa de potencia es la encargada de controlar el funcionamiento de los motores del movimiento del robot por lo que en este circuito se presenta un alto consumo de corriente por lo que su diseño en baquelita se lo realizo por fines estéticos, ya que, cada pista debe estar reforzada con alambre de cobre utilizado en las instalaciones eléctricas para su correcto funcionamiento sin peligro de recalentamiento de cables, ni cortocircuitos.

La figura II.47 muestra el diseño del circuito de potencia:

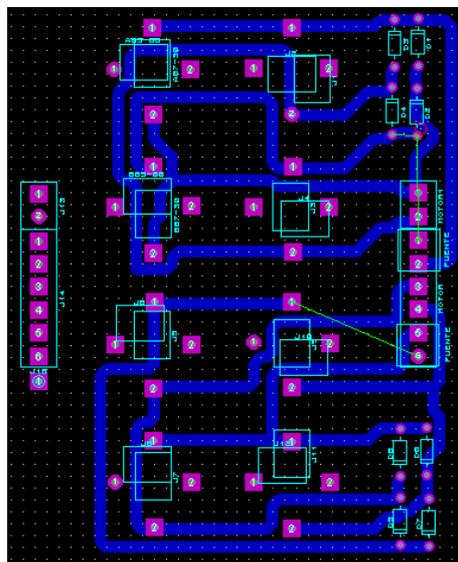


Fig. II.47.- Circuito de la etapa de Potencia

El diseño realizado en ISIS se muestra en la figura II.48:

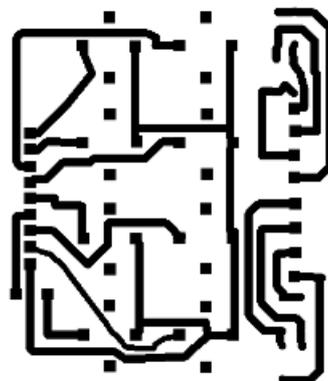


Fig. II.48.- Diseño del circuito de potencia

Una vez que se tienen los diseños en la baquelita se procede a soldar cada uno de los componentes como se muestra en la figura II.49, y además soldar terminales de tal manera que quede espacio para poder insertar los cables de cobre, el cual nos ayudara a soportar los recalentamientos que sufrirán los cables por el alto consumo de corriente.



Fig. II.49.- Placa de potencia

## 2.5. Implementación de la Etapa de Control de Armas

La placa para el control de las armas esta realizada de la manera más compacta y fiable posible y al igual que en la placa de potencia también se tiene que reforzar con alambre de cobre número de 10 de conexiones eléctricas debido al alto consumo de corriente que presentan los motores de arranque 1 y 2 su diseño se muestra a continuación en la figura II.50:

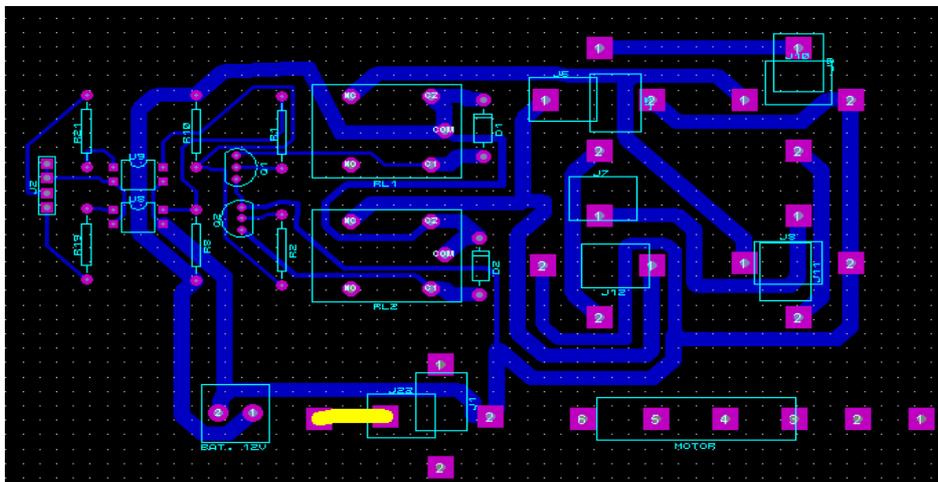
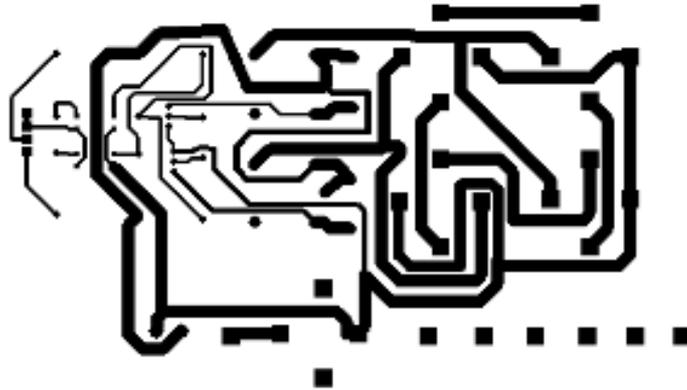


Fig. II.50.- Circuito de la etapa de Control de las armas

El diseño realizado en ISIS se muestra en la figura II.51:



*Fig. II.51.- Diseño del circuito de control de las armas*

Después de tener los diseños en la baquelita se procede a soldar cada uno de los componentes como se muestra en la figura II.52. Y además soldar terminales de tal manera que quede espacio para poder insertar los cables de cobre, el cual nos ayudara a soportar los recalentamientos que sufrirán los cables por el alto consumo de corriente



*Fig. II.52.- Placa de control de armas*

De esta manera se realiza el diseño y construcción de cada una de las partes que conforman el robot de batalla, tanto en la parte mecánica como en la parte eléctrica-electrónica. De la manera más sencilla, fiable, segura y tratando siempre de optimizar recursos debido a los altos costos de materiales para la construcción de robots de batalla.

Permitiendo tener un robot de batalla con un alto grado de competencia el cual será difícil de vencer y con el cual se tendrá grandes posibilidades de ganar eventos no solo a nivel local sino también a nivel internacional.

## 2.6. Diseño del circuito de transmisión del módulo Bluetooth

Para su diseño se utilizará el módulo BLUESMIRF-RN-V1, el cual recibirá las señales del teléfono celular con tecnología Bluetooth. El diseño realizado en ISIS se muestra en la Figura II.53:

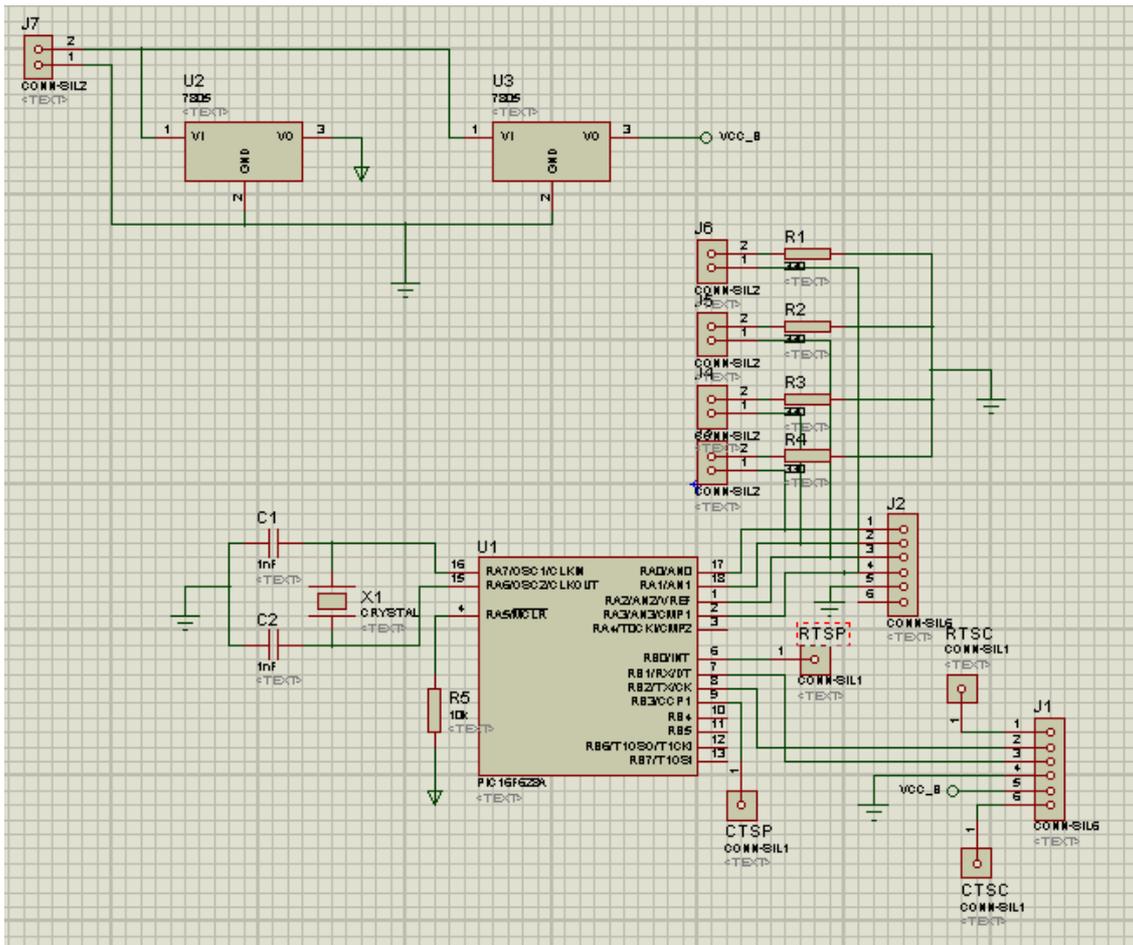


Fig. II.53.- Circuito del modulo Bluetooth en ISIS

El diseño realizado en ARES se muestra en la figura II.54:

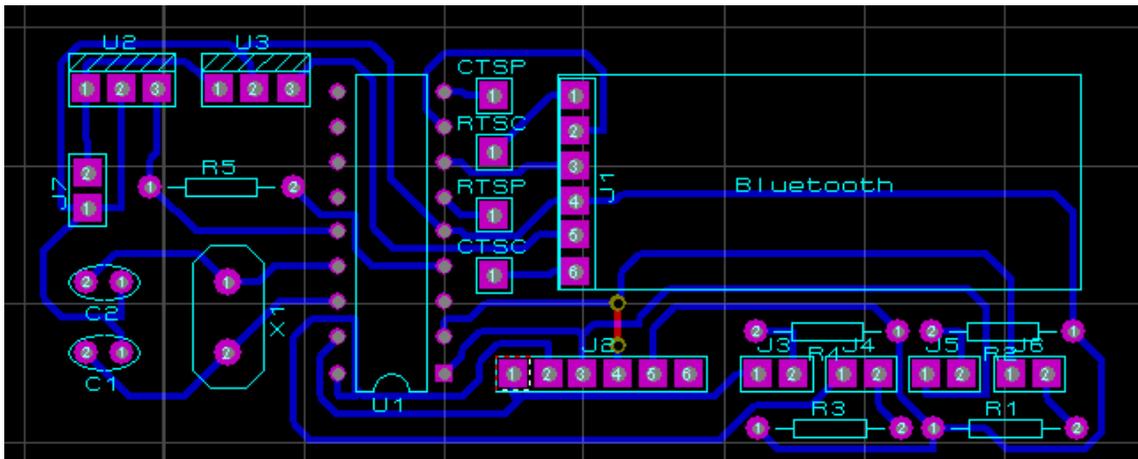


Fig. II.54.- Circuito del modulo Bluetooth en ARES

## **CAPITULO III**

### **PRUEBAS Y RESULTADOS**

#### **3.1. ETAPA DE CONTROL**

Las pruebas se realizaron de manera independiente para cada una de las etapas, de esta manera se obtuvieron los mejores resultados.

La etapa de control es la encargada de gestionar los movimientos del robot así la activación y desactivación de sus armas. El control de robot se realiza mediante la inserción de pulsos “0” y “1” de esta manera se ingresan códigos al micro controlador PIC16F628A lo que permite realizar una comparación efectiva para ejecutar las instrucciones.

## TABLA DE VERDAD

S1	S2	S3	S4	S5	S6	S7	S8	INSTRUCCIÓN
1	0	0	0	0	0	0	0	ADELANTE
0	1	0	0	0	0	0	0	ATRÁS
0	0	1	0	0	0	0	0	IZQUIERDA
0	0	0	1	0	0	0	0	DERECHA
X	X	X	X	1	0	0	0	ENCENDER RODILLO
X	X	X	X	0	1	0	0	APAGAR RODILLO
1	0	0	0	0	1	0	0	SUBIR PALA
0	1	0	0	1	0	0	0	BAJAR PALA

*Tabla III.III.- Tabla de verdad*

La tabla de verdad indica los códigos que se ingresaron al microcontrolador PIC18F628A, byte por byte los mismos que son comparados y ejecutados para activación de los relés de 12v -1Amp los cuales controlan el funcionamiento del puente H de movimiento del robot, puente H de la pala y el relé de encendido y apagado del rodillo.

La placa de la etapa de control funciona con una batería de 12v-4.5A. El PIC16F628A es alimentado por una batería de 6V controlada por un regulador de voltaje 7805 para su óptimo desempeño.

### 3.2. ETAPA DEL PUENTE H

Una vez comprobada de la etapa de control, se conecta al puente H para verificar su funcionamiento para comprobar los máximos y mínimos consumos de corriente, de esta forma podemos saber el tiempo máximo de funcionamiento dependiendo del tipo de consumo que este realice. Sabiendo que el consumo de un motor con su respectiva caja de engranaje es de:

<b>Motor con caja de Engranaje</b>	<b>Consumo mínimo (A)</b>	<b>Consumo nominal (A)</b>	<b>Consumo máximo (A)</b>
	4	7	12

*Tabla III.IV.- Consumo de corriente del motor de CC*

La alimentación del puente H de control de movimiento del robot y armas se lo realiza con una batería de 12v -12A, por lo cual mediante pruebas realizadas se obtuvo diferentes tiempos de funcionamiento, además que se adjunta un tiempo de batalla de prueba, el consumo de las baterías se lo puede expresar por tiempos cortos, medios o prolongados como se muestra a continuación

	<b>Carga de batería (A)</b>	<b>Consumo(A)</b>	<b>Minutos</b>
Consumo Mínimo	12	4	25
Consumo Nominal	12	7	15
Consumo Máximo	12	12	10
Consumo en batalla	12	4 - 12	12

*Tabla III.V.- Tiempo de funcionamiento del robot*

La tabla anterior mediante pruebas realizadas muestra consumos de corriente en un determinado tiempo, lo que evita que se den posibles esfuerzos inútiles debido a que el consumo de corriente produce un recalentamiento en los cables de conexión lo cual incrementa el riesgo de corto circuito.

### **3.3. Circuitos de Armas**

Las pruebas realizadas con los circuitos de control de armas son similares a las del puente H con la diferencia que el motor utilizado para el funcionamiento de dichas armas, es un motor de arranque de moto de 12V con un máximo consumo de corriente de 12A, los cuales son alimentados por una batería independiente de 12V – 12A.

### 3.3.1. Pala

<b>Motor de Arranque</b>	<b>Consumo mínimo(A)</b>	<b>Consumo Máximo(A)</b>
	4	12

*Tabla III.VI.- Consumo de corriente del motor de arranque*

El funcionamiento estará dado siempre por el consumo mínimo con el cual arrancara el motor, dando paso inmediatamente a un consumo máximo de corriente debido que por su mecanismo deberá utilizar una fuerza máxima cuando trata de suspender un peso de 15Kg el cual se lo estableció en el diseño de la plataforma de levantamiento, por lo cual el tiempo de utilización de la batería no es muy largo ya que el consumo de corriente es bien alto. Al combinarse el uso del rodillo reduce su utilidad de la pala en un máximo de 10 levantamientos por batalla.

### 3.3.2. Rodillo

El rodillo funciona con el segundo motor de arranque, pero su uso es de una manera diferente y la tabla de consumo de corriente se presenta a continuación:

<b>Motor de arranque</b>	<b>Consumo Mínimo (A)</b>	<b>Consumo Nominal (A)</b>	<b>Consumo Máximo (A)</b>
	4	7	12

*Tabla III.VII.- Consumo de corriente del Rodillo*

El rodillo tiene un consumo nominal de 7A mientras se busca la búsqueda de su objetivo, al encontrarlo y golpearlo su consumo se vuelve máximo, por lo cual hay que tener mucho cuidado debido a que si el motor se detiene bruscamente podría fundirse debido al alto consumo de corriente.

Cuando se realiza la combinación de las dos armas el consumo de corriente se incrementa reduciendo la duración de la batería aproximadamente a 6 minutos. Razón por la cual la utilización de las armas debe ser de una manera controlada, además los motores de arranque no deben estar encendidos por más de 5 minutos debido a que tienden a quemarse porque no están contruidos para trabajos que conllevan un tiempo prolongado de funcionamiento.

### 3.4. Bluetooth

En las pruebas realizadas para la transmisión de datos al robot de batalla mediante el módulo Bluetooth, debido a lo establecido en concursos para este tipo de robot y por seguridad se hicieron pruebas inalámbricamente con el robot a una distancia aproximada de 4 – 6 metros. Las cuales aprobó satisfactoriamente evitando el común cruce de radiofrecuencias que se dan en estos tipos de eventos de batalla de robot.

Para lo cual se muestra una tabla de consumo de corrientes y voltajes del módulo Bluetooth:

<b>Módulo Bluetooth</b>	<b>Consumo Mínimo</b>	<b>Consumo Máximo</b>
Consumo de Voltaje (V)	3.3	5.5
Consumo de Corriente (mA)	25	450

*Tabla III.VIII.- Consumo de corriente y voltaje del Bluetooth*

## CONCLUSIONES

- La necesidad de tener un robot de batalla que sea lo suficientemente competitivo llevo al planteamiento, diseño y construcción de “RED BULL”, como tema de tesis de grado, un robot de batalla que está dotado de armas con la capacidad de causar graves daños en los robots contrarios, con capacidad de reacción y defensa.
- Una de las pruebas se realizo en el Concurso Ecuatoriano de Robótica CER 2009, donde el diseño realizado se desempeño de una manera óptima con lo que respecta a la movilidad y funcionalidad de las armas planteadas así como el funcionamiento de los circuitos eléctricos y electrónicos.
- El robot tiene un confiable sistema de protección ya que las armas funcionan también como defensa al proteger la pala las fuentes de energía (baterías) y el rodillo la parte de control (circuitos) que son indispensables para el funcionamiento del robot.
- El diseño de los circuitos electrónicos brinda la posibilidad de tener cada una de las etapas protegidas contra los diferentes niveles de voltaje y consumos de corriente. Y por sobre todo las tareas que se ejecutan en cada una de ellas.
- El circuito de la etapa de control de potencia el cual posee un PIC16F628A se convierte en el cerebro del robot de batalla ya que este es el encargado de controlar cada una de las partes del robot, tanto en su movimiento como en el control de las armas, debido a esto tiene asignada protección mediante la utilización de Optoacopladores PC817 los que se encargan de protegerlo de corrientes inversas producidas por los diversos cambios de consumo que sufren debido a la utilización de motores de CC.
- El robot gracias al diseño compacto brinda la posibilidad de ubicar las baterías y circuitos en lugares estratégicos que dan la facilidad para poder realizar cualquier sustitución de relés.
- Como en la etapa de movimiento dentro de la etapa de control de armas obtuvimos un esquema confiable y seguro nuevamente por la utilización de Optoacopladores PC817 y además del uso de relés de 12V-30A, los que tienen la capacidad de soportar el alto consumo de corriente.
- Se concluye que la prueba realizada en el Concurso Ecuatoriano de Robótica realizado en la ciudad de Ibarra en la Universidad Técnica del Norte en el 2009, fue confiable ya que mostró un correcto desempeño en su movimiento y control de armas.

- El robot esta dotado de seguridad y facilidad para su manejo de esta manera brindando la capacidad tener una buena presentación no importa el lugar ni el nivel que tengan los eventos en los que pueda presentarse.
- La segunda prueba de “RED BULL”, fue en el Primer Torneo Internacional de Robótica realizado en el Centro de Convenciones “Simón Bolívar” de la ciudad de Guayaquil, donde nuevamente se pudo comprobar que los diseños de movimiento y armas son lo suficientemente confiables, ya que el uso de las armas en especial el rodillo eléctrico brindo la posibilidad de ganar la mayoría de las batallas por inhabilitación de los robots rivales.

## RECOMENDACIONES

- Cuando se realice cualquier tipo de diseño lo más recomendable es fijarse muy bien en los reglamentos de estos tipos de eventos para no sobrepasar los límites establecidos ya sean estos de dimensiones o peso que son las consideraciones más importantes a tomar dentro de la construcción de robots de batalla.
- Para la construcción se recomienda la utilización de materiales reciclados los cuales son muy fáciles de conseguir como por ejemplo: para la estructura, motores de CC y cajas de engranajes que se encuentran en carros para niños, los cuales abaratan los costos de la construcción debido al alto costo de motores específicamente para el movimiento.
- En lo que tiene que ver con los circuitos electrónicos se recomienda tratar en lo posible separar el sistema. Para de esta manera saber en qué parte del circuito estamos trabajando y si existe alguna avería tener una forma más fácil para darle solución, además nos ayuda a ubicar los circuitos de una manera más fácil dentro de la estructura del robot.
- Es recomendable diseñar la etapa de potencia con relés de 12V-30A para brinda una mayor seguridad al circuito de potencia ya que los consumos de los motores funden con mucha facilidad los relés comunes de 10A lo cual siempre ha sido un problema al momento del diseño de robot de batalla por el alto costo y escasez de este tipo de relés.
- En cuanto tiene que ver con la construcción de los circuitos de cada una de las etapas que se planificaron para este robot lo más importante por recomendar tiene que ver con la verificación de cada una de las pistas después de la corrosión con el cloruro férrico debido a que si no se realiza una verificación correcta podemos causar cortocircuitos.
- Antes de poner en funcionamiento el robot verificar que los elementos estén conectados y ubicados de la manera correcta.
- Es recomendable revisar las conexiones de entrada y salida del PIC16F628A mediante los relés de activación para de esta manera tener los pulsos en los lugares que son los correctos para la activación tanto de los movimientos como de la activación de las armas del robot de batalla. De la misma forma se recomienda verificar las conexiones de las baterías y su conexión hacia los motores para que estos tengan el giro coordinado para su movimiento.
- Se recomienda siempre verificar el funcionamiento del robot y cada una de sus etapas de manera individual para no tener desperfectos durante la batalla.
- Además se recomienda realizar un cambio de cables regularmente debido a que los cables de las baterías sufren un recalentamiento excesivo.

## RESUMEN

Se realizó el DISEÑO Y CONSTRUCCIÓN DE UN ROBOT DE BATALLA CONTROLADO MEDIANTE DISPOSITIVO BLUETOOTH, manipulado desde un teléfono celular.

La estructura mecánica del robot está constituida en su totalidad de hierro reciclado, el sistema de armas consta de 2 motores de arranque a 12V – 6A los cuales controlan un rodillo construido con un tubo de hierro que recubre 4 rodamientos de  $\frac{3}{4}$ " que facilitan su movimiento el rodillo consta de 32 puntas de hierro, el otro motor controla un cilindro eléctrico que se acopla mediante un cable de acero recubierto por un resorte acerado a la medida permitiendo al cilindro desplazar un pala metálica en forma vertical. El sistema motriz consta de 4 Motores de CC con caja de engranaje los cuales operan a 12V – 7A.

El control de potencia de armas y movimiento está diseñado mediante un puente H casero constituido por 13 relés Bosch 12V – 30A y 13 diodos rectificadores 1N4004, circuito de control eléctrico y electrónico 12 relés 12V – 1A, resistencias, Reguladores 5V, optoacopladores PC817, transistores 2n3904, 1 PIC16F628A, circuito de transmisión inalámbrica mediante Bluetooth, 1 Bluetooth BLUESMIRF-RN-V1 a 5V – 450mA, un micro controlador PIC16F628A, 2 reguladores de Voltaje 7805, resistencias, cristal 16 Mhz, capacitores 1nF y diodos led. Para la alimentación se utilizaron 2 baterías de 12V – 12A, 1 batería de 12V – 4.5A, dos baterías a 6V – 500mA.

Se obtuvo un robot pesa 47kg, con una altura de 28cm, con armas extendidas 97cm de largo y ancho 56.5cm, es controlado inalámbricamente mediante Bluetooth a un rango de frecuencia 2.4~2.524 GHz y con una Modulación GFSK, 79 canales en intervalos de 1 MHz, generadas por el modulo y un teléfono celular con tecnología bluetooth que es recibida por el robot y comparada en los micro controladores los cuales generan los códigos necesarios para el movimiento del robot y el control de sus respectivas armas a una distancia de 6 – 20m.

Se recomienda que en este tipo de proyectos se involucre las áreas de Mecánica, Electricidad, Electrónica, Control y Telecomunicaciones, para mayor aplicación de la ciencia robótica.

## SUMMARY

It realize out the DESIGN AND CONSTRUCTION OF A BATTLE DROID CONTROLLED BY MEANS OF BLUETOOTH, manipulated from a cell phone.

The robot's mechanical structure is made entirely of recycled iron, the weapon system consists of 2 starters to 12V - 6A which control a roller constructed with an iron pipe lining bearings 4 ¾ "to facilitate their movement The roll consists of 32 points of iron, the other controls an electric cylinder engine that is coupled by a steel cable coated with a spring steel to the extent allowing the cylinder to move a metal blade vertically. The drive system consists of 4 DC motors with gearbox which operate at 12V - 7A.

The power of arms control and movement is designed using a homemade H-bridge relay comprising 13 Bosch 12V - 30A and 13 diodes 1N4004 rectifier circuit of electrical and electronic control 12 relays 12V - 1A, resistors, 5V regulator, opt coupled PC817, transistors 2N3904, 1 PIC16F628A circuit using Bluetooth wireless transmission, 1 Bluetooth BLUESMIRF-RN-V1 to 5V - 450mA, a PIC16F628A micro controller, 2 voltage regulators 7805, resistors, crystal 16 MHz, 1nF capacitors and LEDs. Were used for food 2 batteries 12V - 12A, 1 battery 12V - 4.5A, two batteries of 6V - 500mA.

The result was a robot weighs 47kg with a height of 28cm, with outstretched arms 97cm long and 56.5cm wide, is controlled wirelessly via Bluetooth to a frequency range 2.4 ~ 2.524 GHz and a modulation GFSK, 79 channels at intervals of 1 MHz, generated by the module and a cell phone with Bluetooth technology that is received by the robot and compared in micro drivers which generate the necessary code for the robot motion and control of their weapons to a distance of 6 - 20m.

It is recommended that such projects will involve the areas of Mechanical, Electrical, Electronic, Control and Telecommunications, for greater application of science robotics.

# BIBLIOGRAFÍA

- BARNES, JD y KÖLLING, M. Programación orientada a objetos con JAVA. Ohio- Estados Unidos. Editorial Pearson Education. 2007. pp. 68-91.
- GARCÍA, J. Aprende Java. Tercera Edición. San Sebastián-España. Editorial Tapa Blanca. 2000. pp- 1-125.
- HILARIO, A.- CASTRO, M. Simulación Y Electrónica Analógica. Prácticas Y Problemas, Segunda Edición. Madrid-España. Editorial Ra-ma. 2003. pp. 54-127.
- HOPKINS, B y RANJITH, A. Bluetooth for Java. Estados Unidos. Editorial APRESS. 2003. pp. 35-98.
- NATHAU, J y Vuelapluma, M. Tecnología Bluetooth. Segunda Edición. España. Editorial McGRAW-HILL/INTERAMERICANA DE ESPAÑA. 2002. pp. 35-67.
- PRIETO, Mj. Curso de J2ME. Segunda Edición. Reino Unido. Editorial AlfaOmega. 2005. pp. 6-60.
- REYES, A CARLOS. Manual de Microcontrolador Studio. Segunda Edición. Quito- Ecuador. Editorial RISPERGRAF. 2006. pp. 135-172.
- SING, L. y KNUDSEN J. Beginning J2ME.Tercera Edición. Estados Unidos. Editorial Springer. 2005. pp. 1-258.

## PÁGINAS DE INTERNET

Descarga de Microcode

- [www.mecanique.co.uk](http://www.mecanique.co.uk)

23-10-2009

Descarga de IC-prog105C.zip

- [www.IC-prog.com](http://www.IC-prog.com)

29-10-2009

Teoría de PIC 16F628A

➤ [www.todopic.com.ar](http://www.todopic.com.ar)

10-11-2009

#### Módulo Bluetooth

➤ <http://www.sparkfun.com/datasheets/RF/RN-41.pdf>

15-01-2010

➤ <http://feemo.in/store/bluetooth%C2%AE-modem-bluesmirf-gold/>

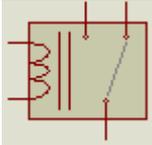
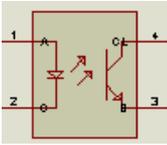
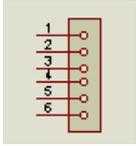
23-01-2010

➤ [http://www.sparkfun.com/datasheets/RF/BlueSMiRF\\_v1.pdf](http://www.sparkfun.com/datasheets/RF/BlueSMiRF_v1.pdf)

02-02-2010

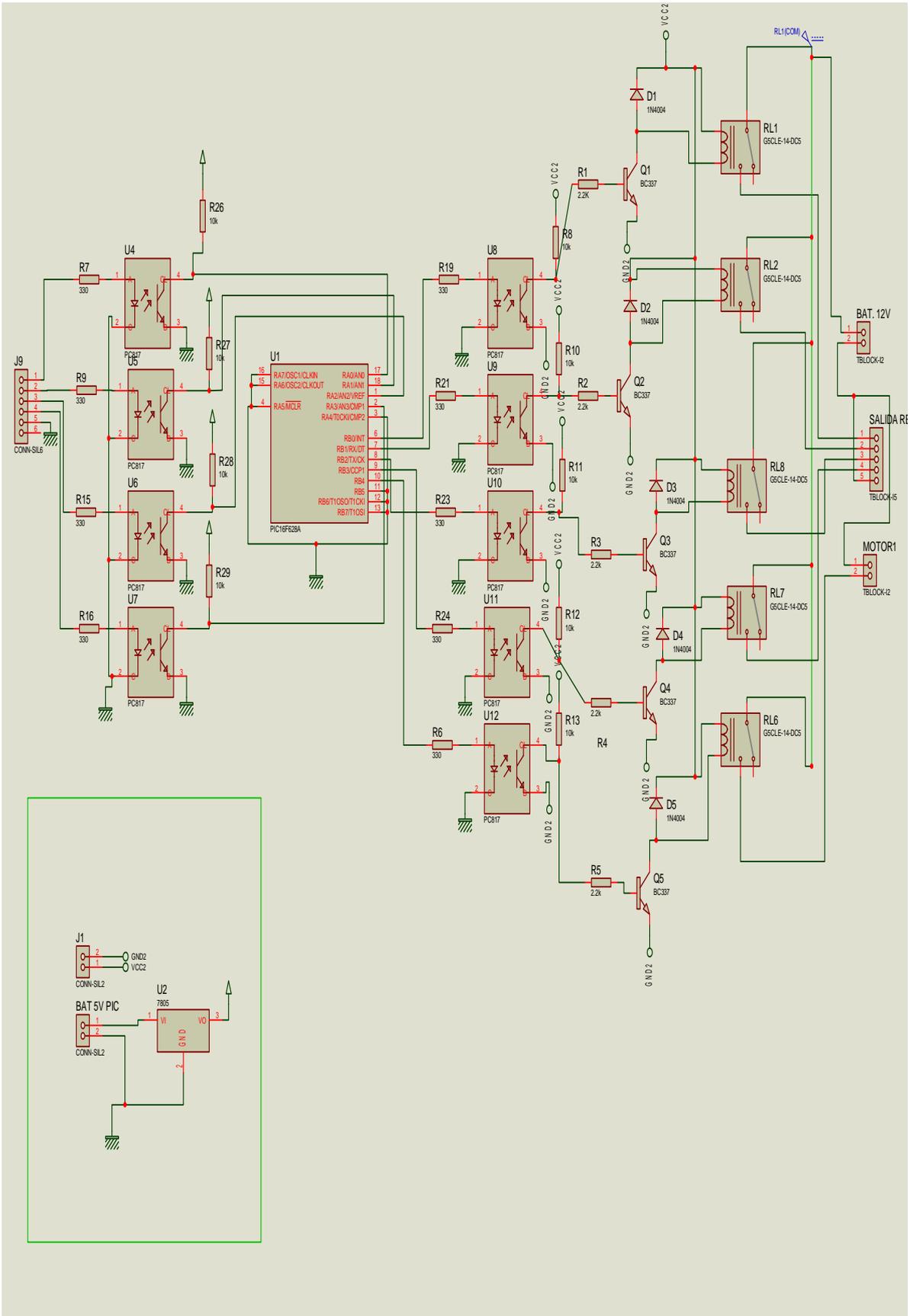
**ANEXO**

## SIMBOLOGÍA DE ISIS

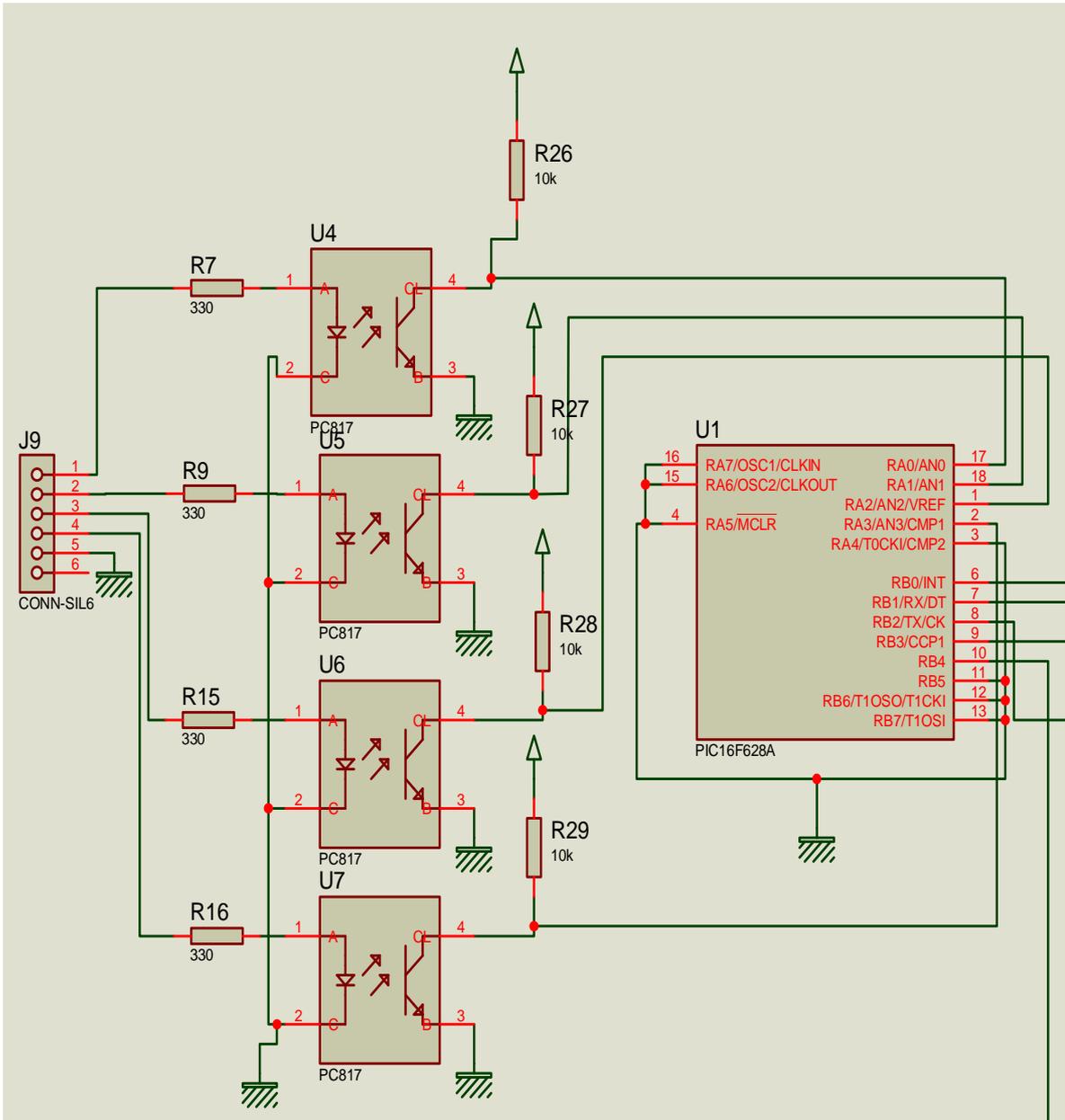
	<p>Transistor bipolar NPN tiene tres terminales: Base, Colector y Emisor</p>
	<p>Resistencia tiene dos terminales sin polaridad</p>
	<p>Regular de voltaje tiene tres terminales: Entrada, Tierra y Salida</p>
	<p>Relé tiene cinco terminales: tres entre las cuales están los contactos de abierto y cerrado y los dos restantes son la bobina</p>
	<p>Optoacoplador tiene cuatro terminales dos de entrada que activan a los dos de salida mediante un haz de luz interno</p>
	<p>Sockets utilizado para la ubicación de los terminales de baterías</p>
	<p>Diodo común tiene dos terminales ánodo y cátodo</p>
	<p>Socket utilizado para diferentes funciones entre ellos para el control de motores</p>
	<p>Símbolo de puesta a Tierra</p>

<p>PIC16F628A</p>	<p>Pic 16F628A</p>
	<p>Positivo</p>

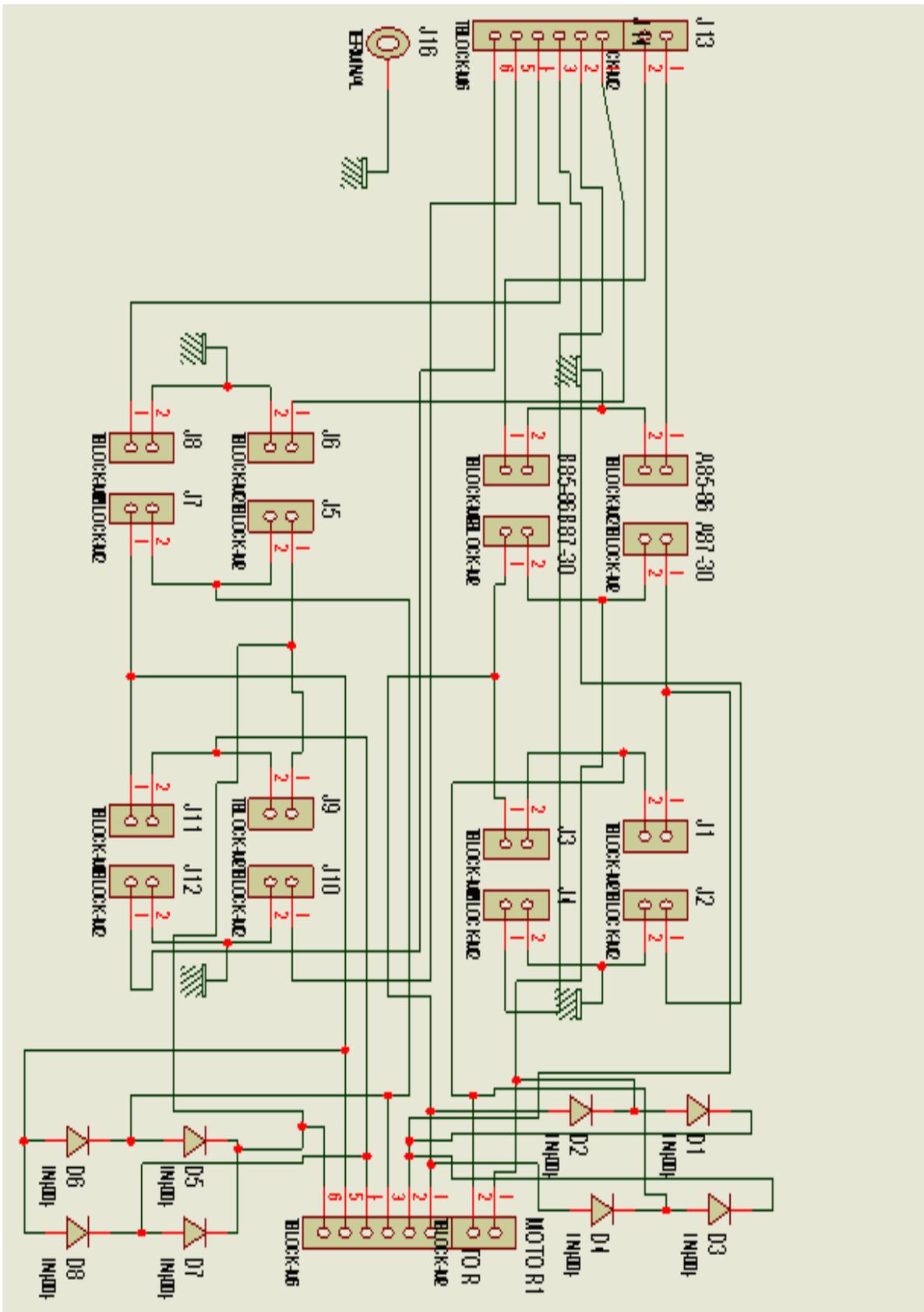
# Diseño de la etapa de Control de Potencia



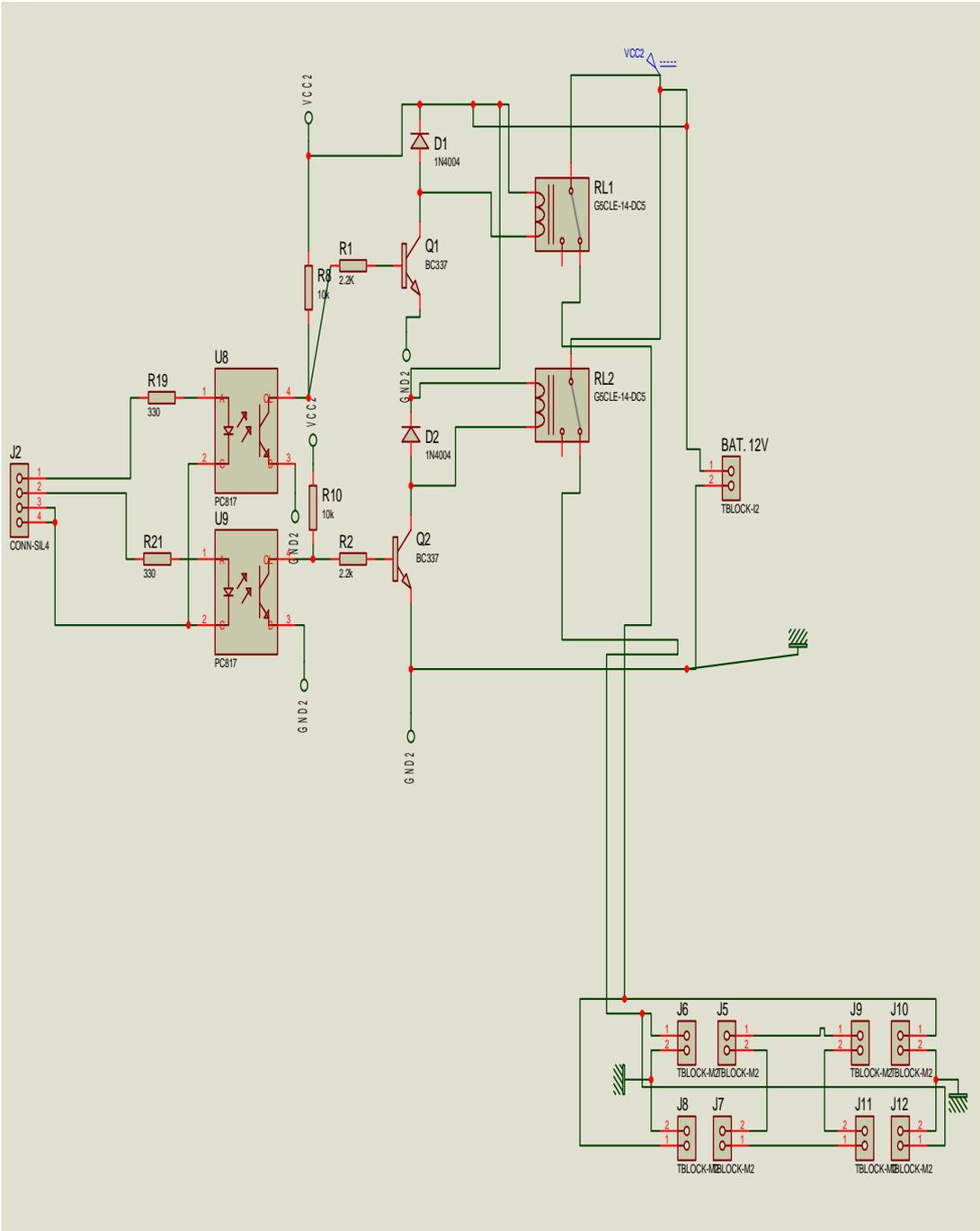
## Ubicación de Optoacopladores a la entrada del Pic 16F628A



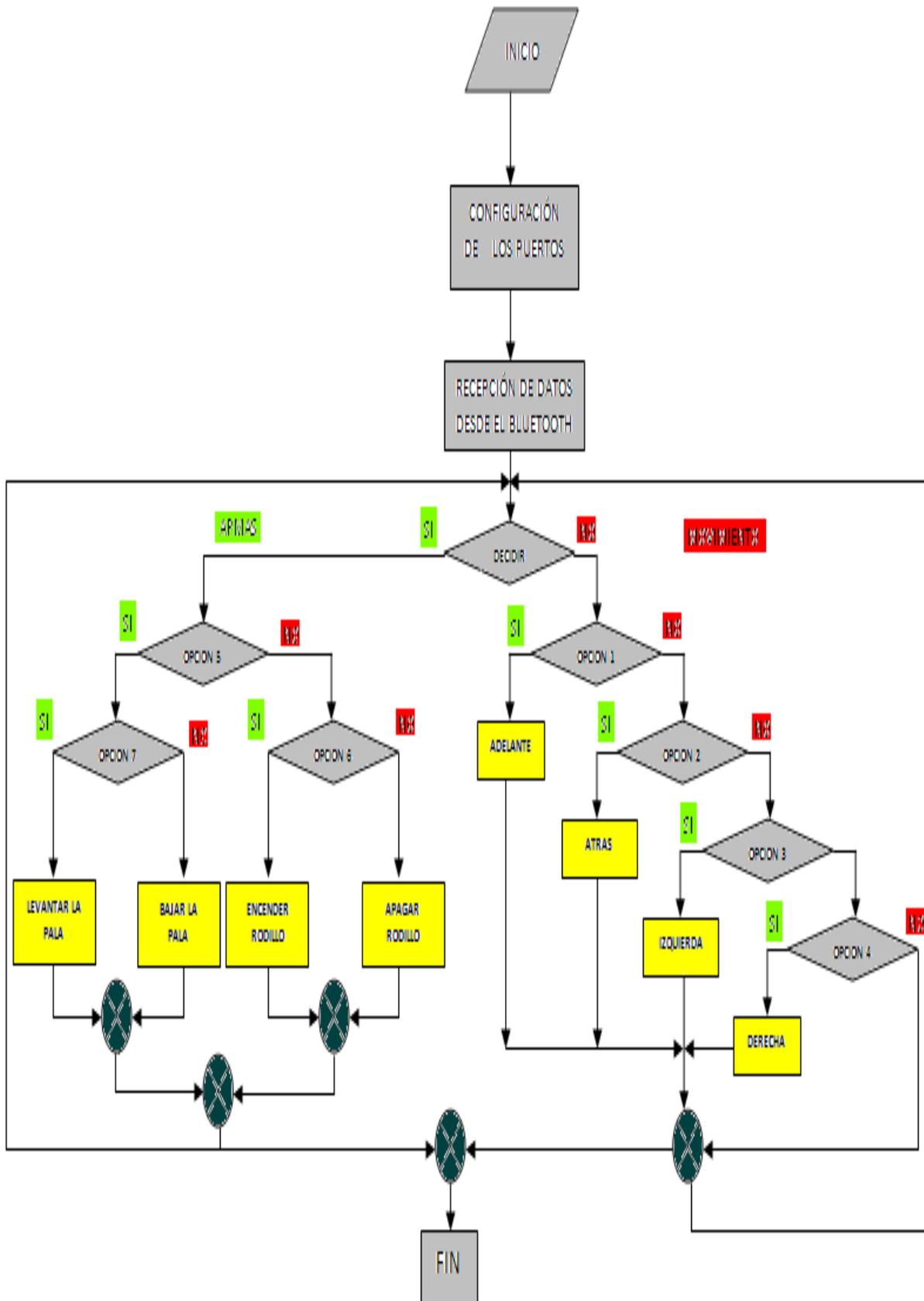
# Puente H



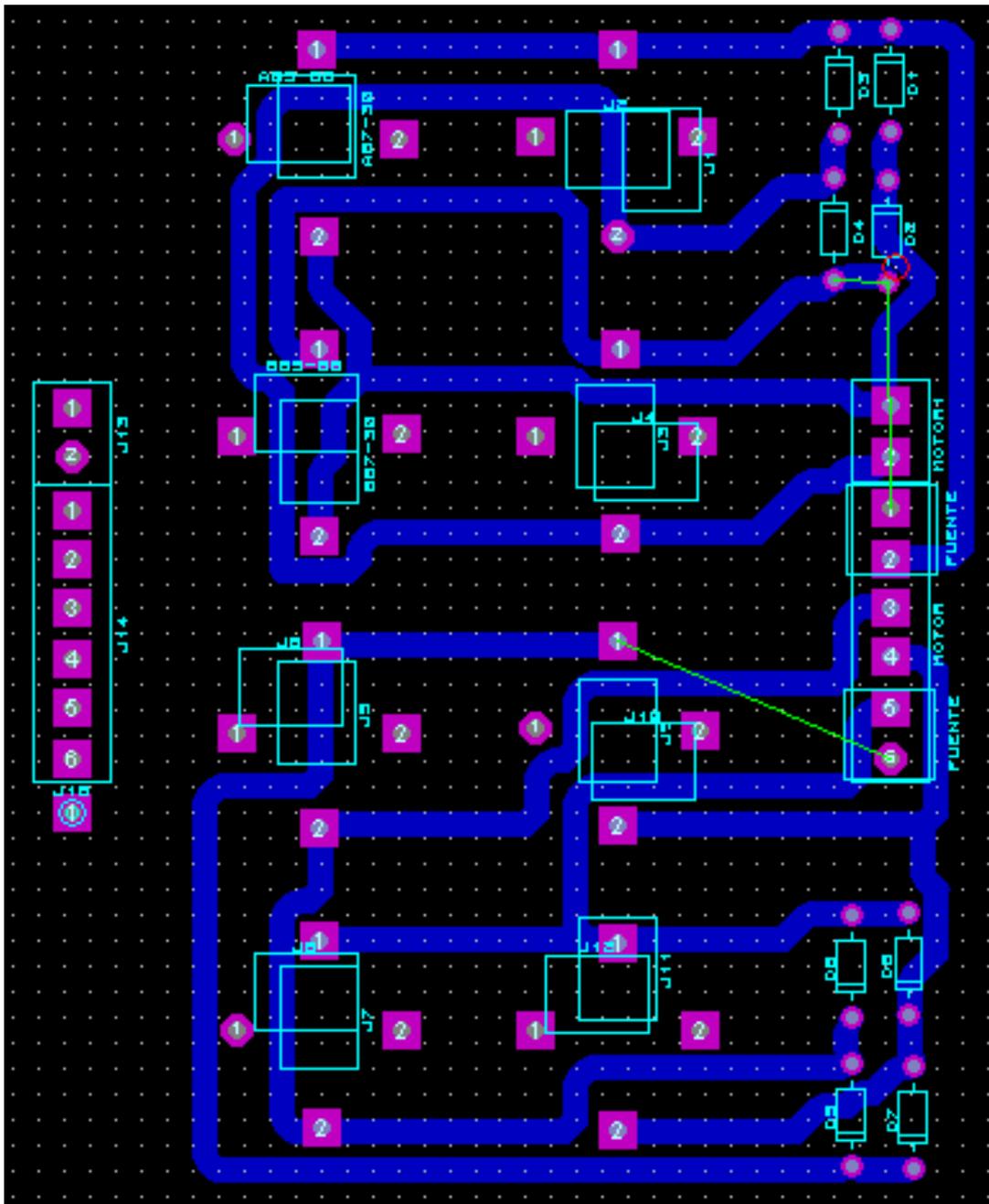
# Circuito de control de la Pala



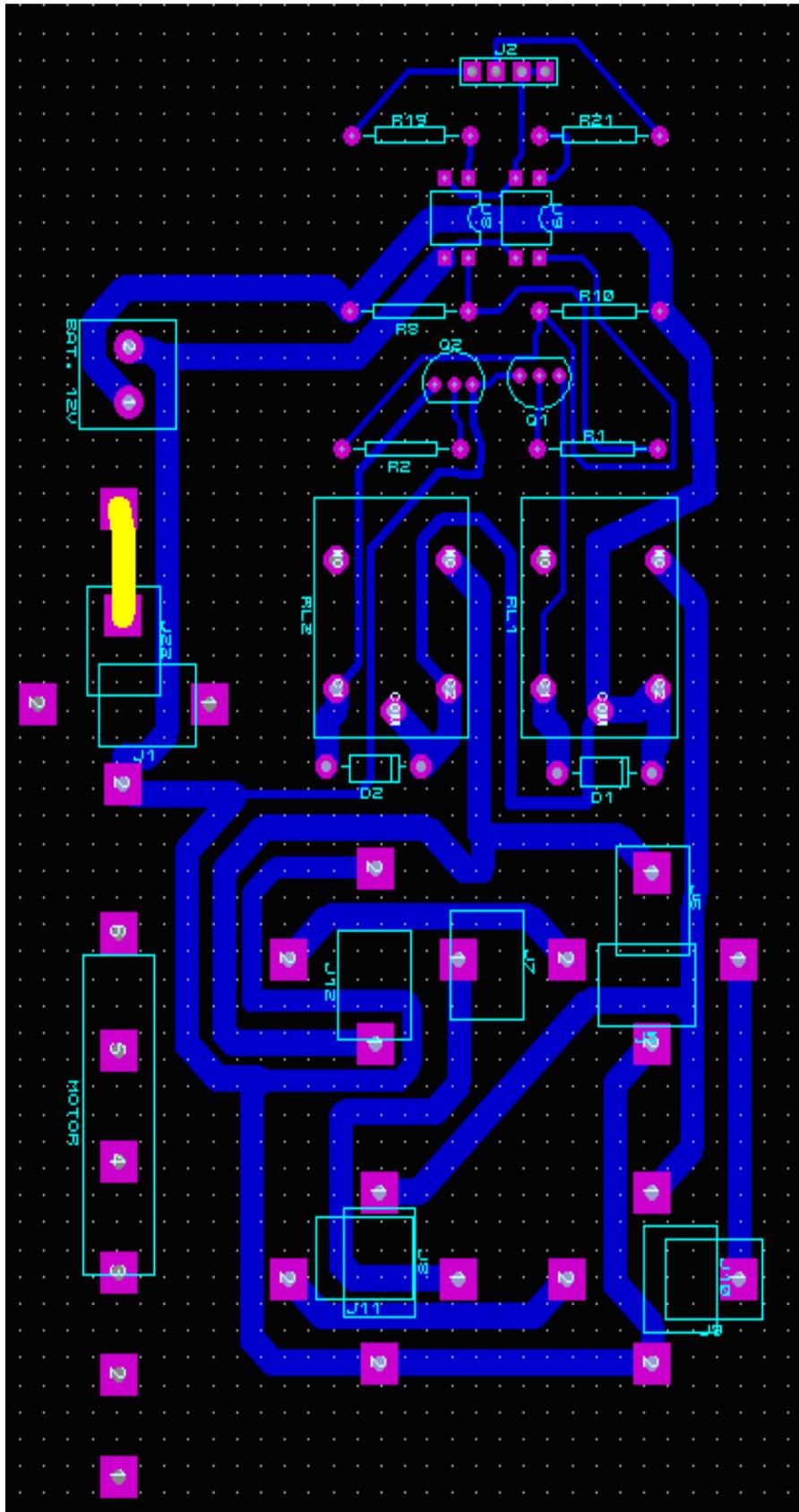
## Diagrama de flujo de funcionamiento del Pic 16F628A



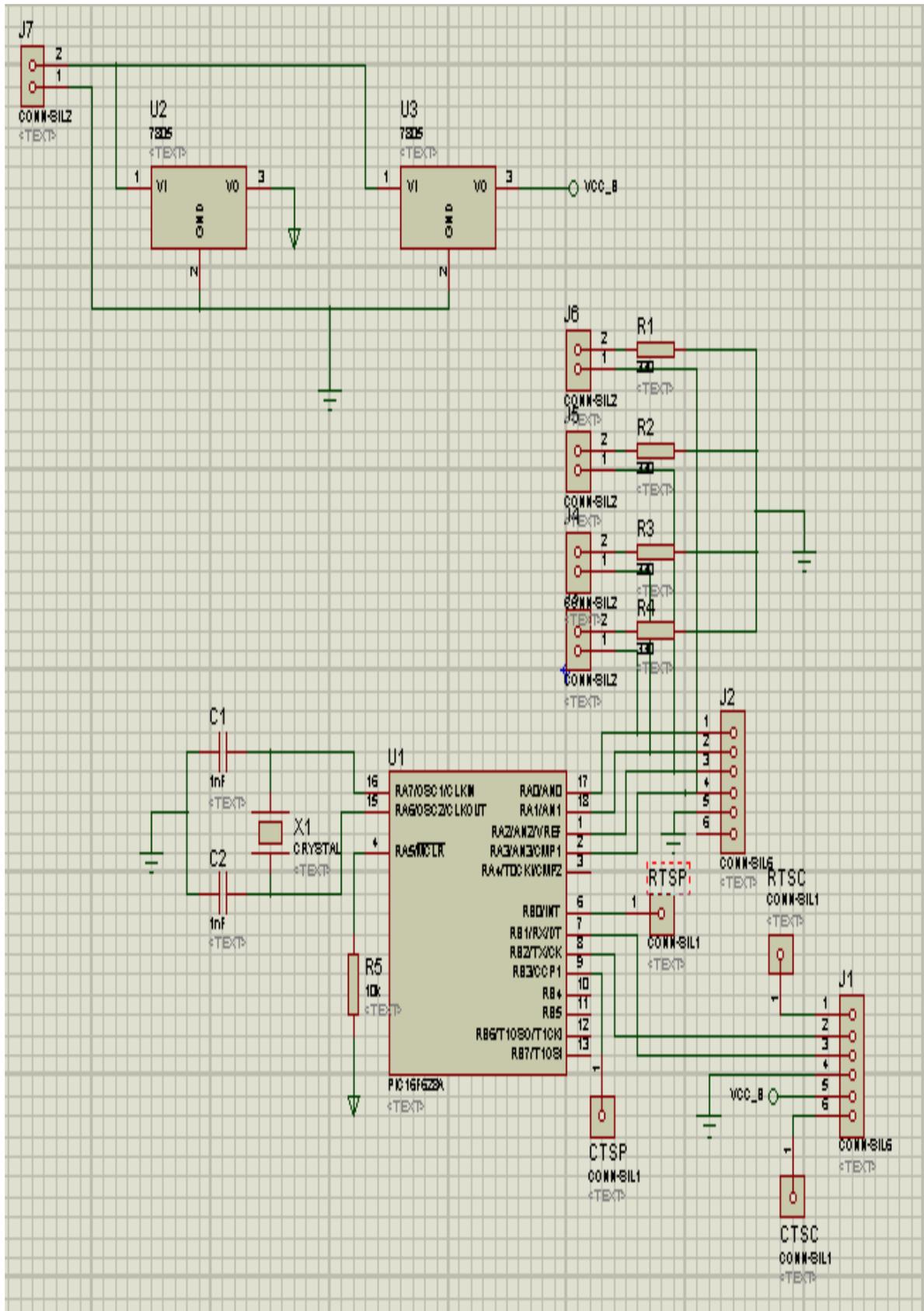
# Circuito de la etapa de Potencia



### Circuito de la etapa de Control de las armas



## Circuito del modulo Bluetooth en ISIS



### Circuito del modulo Bluetooth en ARES

