

ESCUELA POLITÉCNICA DEL EJÉRCITO

Extensión Latacunga

DEPARTAMENTO DE ENERGÍA Y MECÁNICA

Carrera de Ingeniería Automotriz



"DISEÑO Y CONSTRUCCIÓN DE UN ROBOT DE COMBATE CON MOTOR DE COMBUSTION INTERNA"

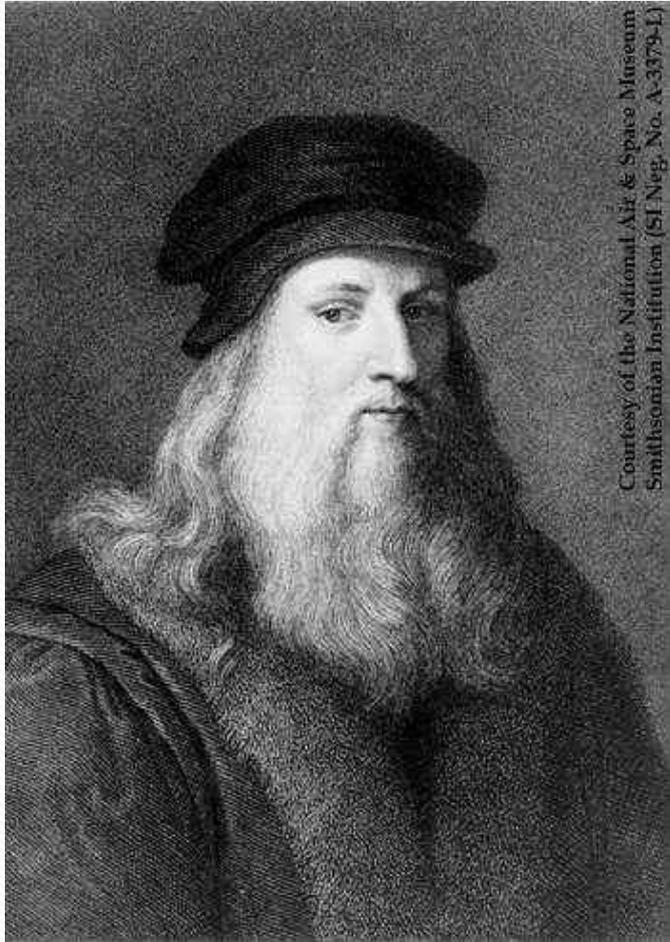
JORGE ANDRÉS CHACON GUZMÁN

Mayo 2012

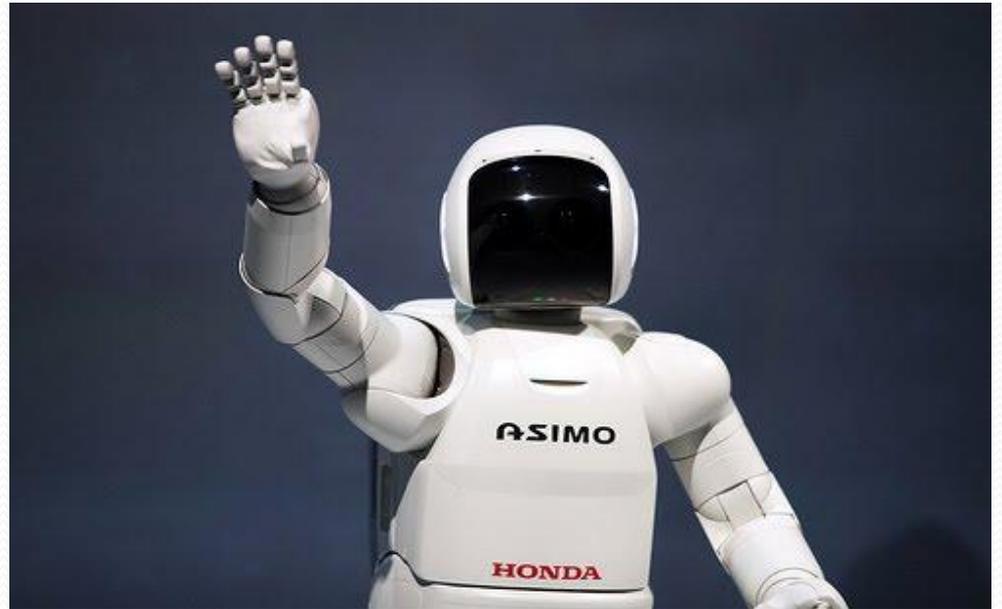
ANTECEDENTES

RESEÑA HISTÓRICA

La historia de la robótica va unida a la construcción de «artefactos», que trataban (en su mayoría) de materializar el deseo humano de crear seres a su semejanza y que los descargasen del trabajo, desde Leonardo da Vinci con la idea de un caballero mecánico en 1495 hasta la fecha con el robot humanoide ASIMO de la multinacional automotriz Honda, el hombre ha tenido en mente esta rama de la tecnología que se dedica al diseño, construcción, operación, disposición estructural, manufactura y aplicación de los robots.

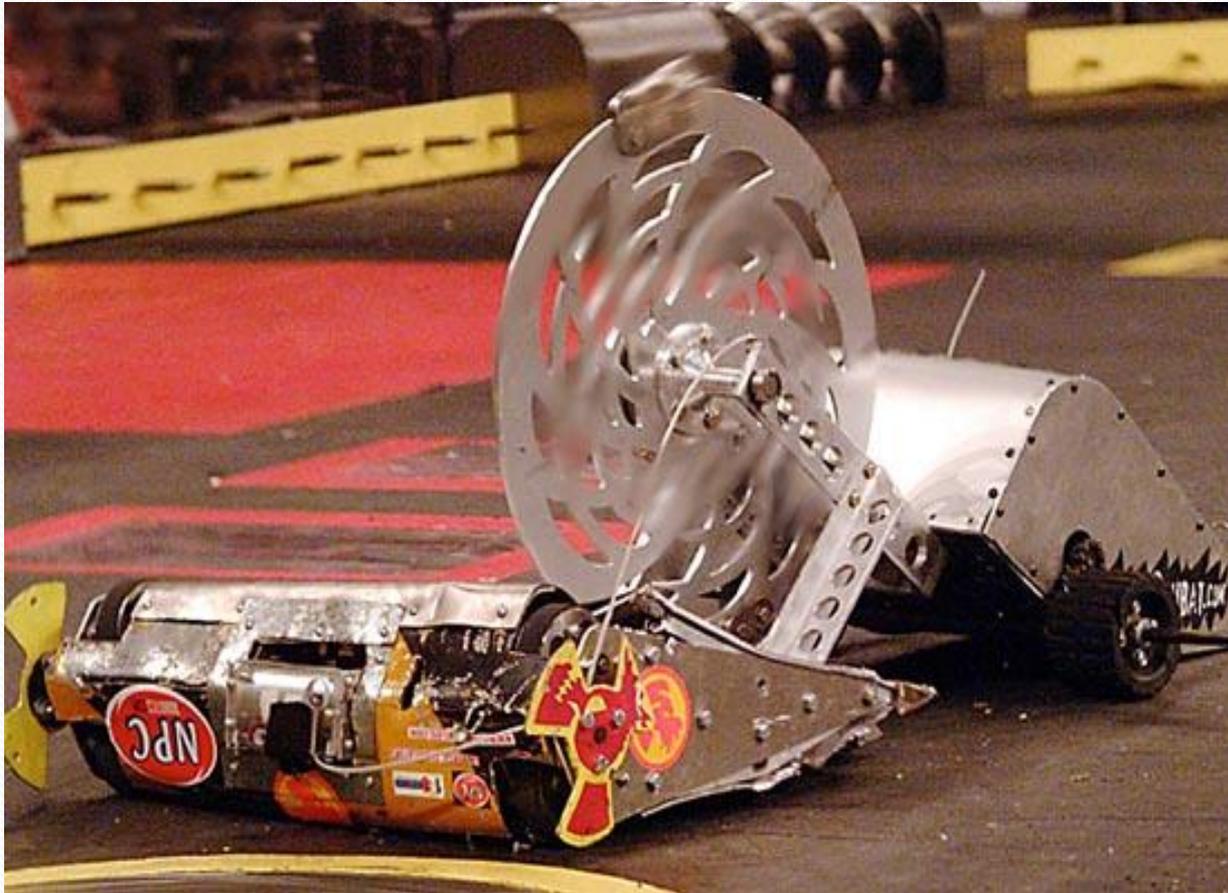


Courtesy of the National Air & Space Museum
Smithsonian Institution (SI Neg. No. A-3379-L)



El combate de robots

Es una afición o deporte en el que dos o más máquinas personalizadas utilizan diversos métodos para destruir o desactivar a los oponentes, en la mayoría de los casos estas máquinas son los vehículos a control remoto en lugar de robots autónomos, aunque hay excepciones particularmente en la categoría robot-sumo





Los combates entre robots disfrutaron de un período de popularidad cuando varios programas de televisión de difusión como BattleBots , Robot Wars , y Robótica los transmitían por televisión, aunque aún hay decenas de pequeñas competiciones en todo los EE.UU. y en otros países cada año, incluyendo Ecuador; localmente la competencia se la realiza periódicamente entre las universidades.

- 
- Los robots pueden variar en su diseño y complejidad, desde modificar los juguetes a control remoto que pesan menos de un kilo a otros de trescientas libras los más sofisticados y de electrónica avanzada.
 - Aunque la construcción de un robot de combate puede costar miles de dólares, cientos de horas de diseño y trabajo, algunas escuelas utilizan la construcción de robots de combate en sus cursos para enseñar el diseño mecánico y la tecnología.

EVOLUCIÓN DE LA ROBOTICA EN EL ECUADOR

El avance de la robótica en nuestro país es indudable desde hace ya algunos años atrás, su desarrollo se evidencia con la creación de carreras afines a ella, que mezclan la electrónica y demás ciencias; nuestra universidad no es la excepción ya que existe el club de robótica de la ESPE extensión Latacunga, que ha conseguido excelentes resultados en certámenes inter-institucionales.

- 
- Los objetivos de este tipo de eventos son; posicionar la robótica y tecnologías afines en la comunidad de jóvenes
 - Brindar a la comunidad un espectáculo único del cual puedan disfrutar y aprender además de fomentar el espíritu competitivo y de trabajo en equipo de quienes practican ésta actividad.
 - Reconocer las habilidades, conocimientos, destrezas y creatividad de todos los competidores.

Situación–problema que amerita el tema del proyecto de investigación

Determinar el objetivo central y los específicos, identificar las razones que señalan que la investigación propuesta ayudará en la solución de problemas y encauzar el proyecto de investigación con una hipótesis de trabajo.

PLANTEAMIENTO DEL PROBLEMA

Existen muchos robots de combate, la gran mayoría de estos son impulsados por medio de uno o varios motores eléctricos, a diferencia de ellos, el proyecto utiliza uno de combustión interna, el cual tiene muchas menos limitaciones en cuanto a su autonomía de energía, así como también más potencia y fuerza.



El reto es realizar las adaptaciones necesarias para que todos los mandos operados manualmente en un motor convencional, sean accionados por medio de servo motores y motores eléctricos controlados mediante radio control, supone un desafío de diseño e inventiva para que estos funcionen correctamente y sea posible operarlos a distancia.

JUSTIFICACIÓN E IMPORTANCIA.

Identificando el problema es evidente que el proyecto justifica su diseño y construcción; debido a que a través de él, se busca implementar una innovadora forma de construir y diseñar un robot, diferente a las tradicionales siendo el primero en su clase y sentando un precedente y marcando diferencia con el resto.

OBJETIVOS DE LA INVESTIGACION

- OBJETIVO GENERAL

Diseñar y construir un robot de combate propulsado con un motor de combustión interna que posea características únicas en cuanto a su movilidad y desarrollo tecnológico, que cuente con mecanismos de accionamiento diseñados en base a la electrónica, el diseño de elementos de máquinas, además del uso software de simulación para todos los componentes estructurales; que sea altamente competitivo así como ágil y potente.

Logo del Proyecto



OBJETIVOS ESPECIFICOS

- Diseñar y construir todos los mecanismos necesarios para su correcta operación y que estos puedan ser operados a distancia a través de Radio Control.
- Definir las características del diseño y construcción del sistema de Potencia.
- Programar con el uso de software los componentes electrónicos que se encargaran del accionamiento de todos y cada uno de los actuadores del robot.
- Crear el marco tubular y realizar la simulación a través de (SOLID WORKS); siendo esta una herramienta de diseño e ingeniería asistida por computador.
- Pronosticar posibles fallas y corregirlas a tiempo.

HIPÓTESIS

- Al culminar el presente trabajo de investigación el robot de combate diseñado y construido con motor de combustión interna tendría todos mecanismos de accionamiento controlados electrónicamente a través de radio control, tanto los encargados del direccionamiento, aceleración, seleccionamiento de marchas, encendido y apagado a distancia, etc.

COMPONENTES PRINCIPALES

- TREN MOTRIZ

Como elemento motriz del robot de combate se ha escogido un motor de cuatrimoto, de la Marca (Motor 1), modelo Rocket, con las siguientes especificaciones.

Especificaciones del motor Rocket

125 cc.

Cilindrada (cc.)	125
Potencia/r.p.m.	16.41kW /8000 min.-1
Refrigeración	Aire
Tipo	Monocilíndrico, 4 Tiempos
Válvulas	2
Distribución	Árbol de levas sencillo en cabeza (SOHC)
Transmisión	Semi-automática
Encendido	CDI(ignición por descarga de capacitor)
Lubricación	Por Bomba de Aceite y cárter mojado
Combustible	Gasolina
Alimentación	Carburador Keihin 20 mm de aguja.
Diam. x carrera	54.00 mm. x 54.00 mm.
Relación de Compresión	9,33:1
Arranque	Eléctrico
Peso	40 libras



DISEÑO Y CONSTRUCCIÓN DE LOS COMPONENTES ESTRUCTURALES

- ESTRUCTURA TUBULAR
- El armazón o chasis del robot es tubular, esta técnica utiliza como estructura una red de tubos metálicos soldados, recubierto después con fibra de vidrio, esta técnica consigue una carrocería de gran rigidez y resistencia con muy poco peso.

EMPLEO DE SOFTWARE

- Todo el diseño, modelado y simulación de la estructura así como sus componentes mecánicos y de suspensión se lo realizará con el uso del software SolidWorks que es un programa de diseño asistido por computadora para modelado mecánico.

Logo del Software SolidWorks.



DISEÑO ARMAZON PRINCIPAL.

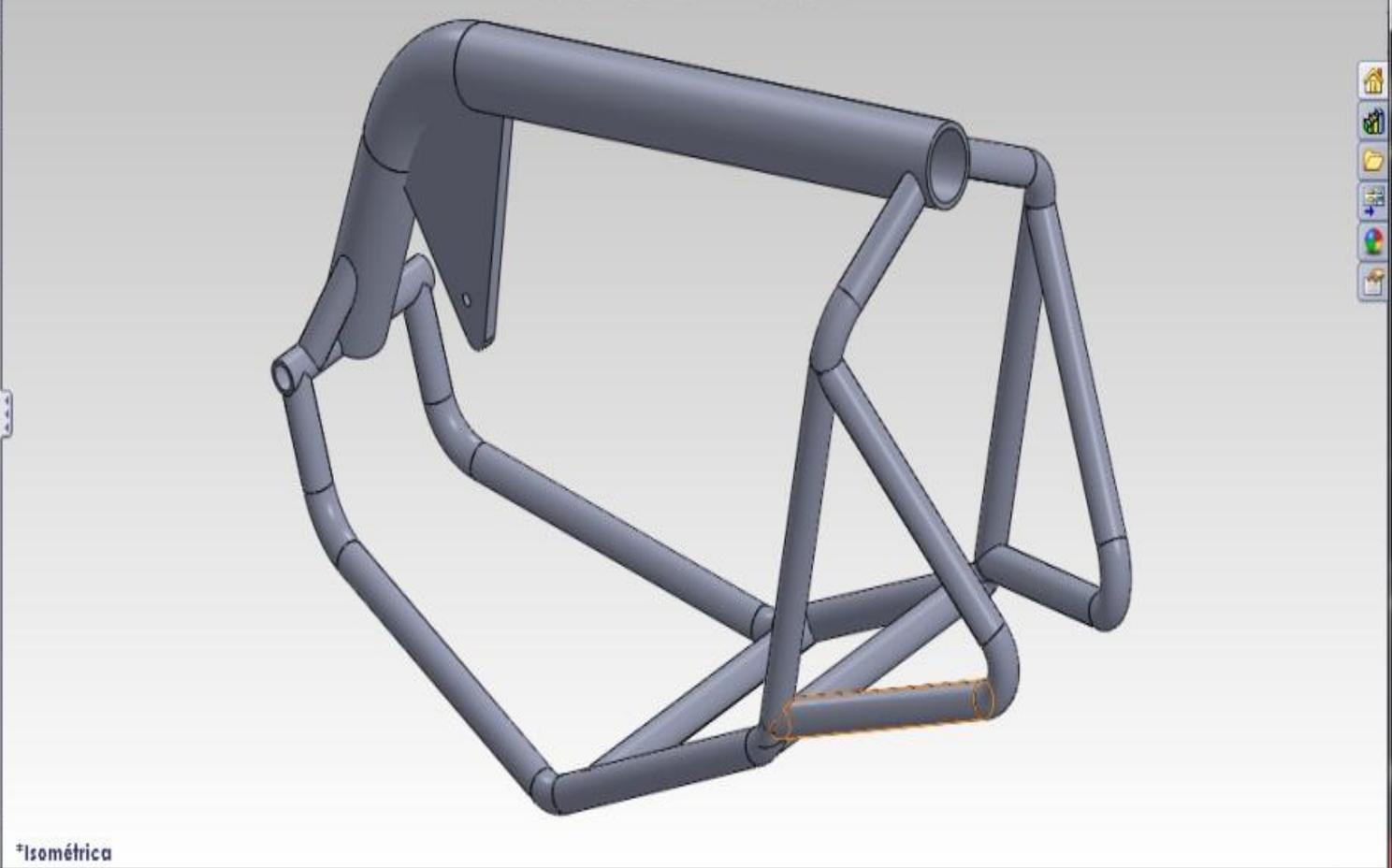
- El marco o armazón principal del robot de combate cumple varias funciones, principalmente soporta el peso del motor a través de sus dos bases por donde dos pernos sostienen al motor asegurándolo para que quede firme y seguro.



- Extruir superficie
- Revolución de superficie
- Barrer superficie
- Recubrir superficie
- Superficie limitante
- Rellenar superficie
- Forma libre
- Superficie plana
- Equidistandar superficie
- Superficie reglada
- Eliminar cara
- Reemplazar cara
- Extender superficie
- Recortar superficie
- Eliminar recorte
- Coser superficie
- Dar espesor
- Corte con espesor
- Cortar con superficie
- Geometría de refe...
- Curvas de refe...

Operaciones Croquis Superficies Piezas soldadas Calcular DimXpert Productos Office

- Sensores
- Anotaciones
- Lista de cortes(33)
- Material <sin especificar>
- Alzado
- Planta
- Vista lateral
- Origen
- Pieza soldada
- Miembro estructural4
- Miembro estructural5
- Miembro estructural6
- Miembro estructural7
- Miembro estructural8
- Miembro estructural9
- Miembro estructural10
- Miembro estructural13
- Miembro estructural14
- Miembro estructural15
- Croquis3D7
- radial ball bearing_68_iso
- Miembro estructural19
- Miembro estructural21
- Miembro estructural22
- (-) Croquis3D15

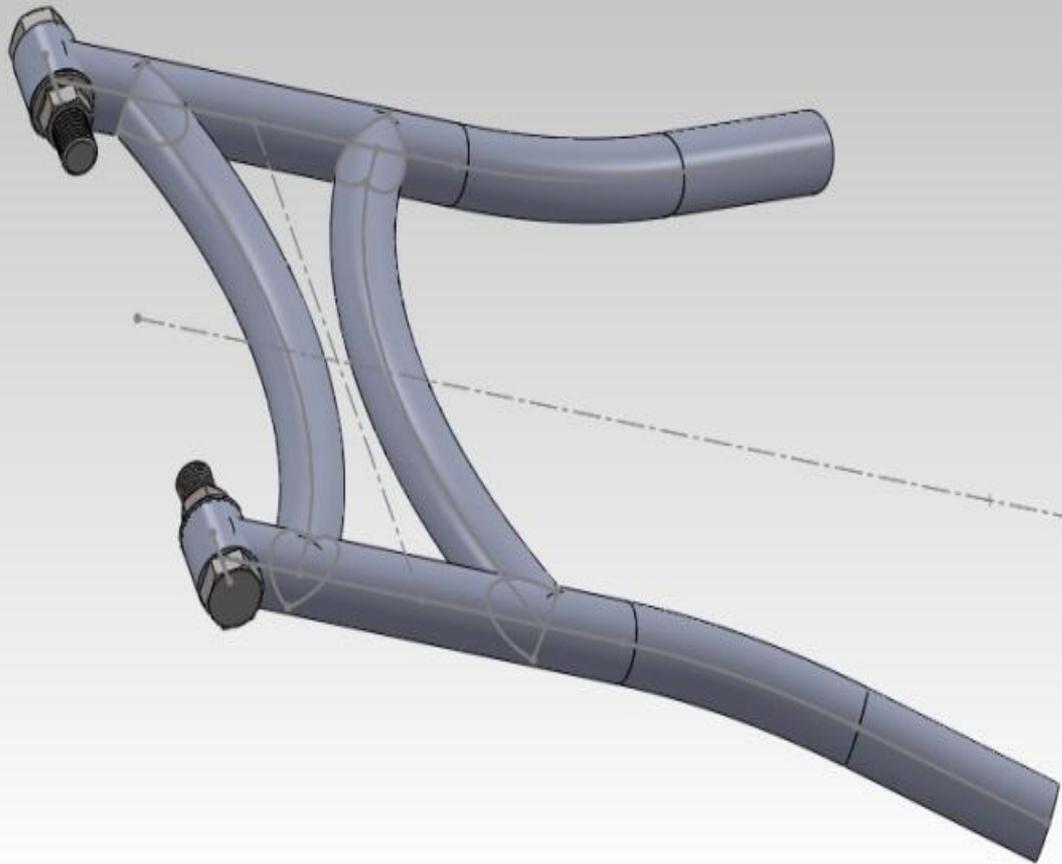


DISEÑO DEL MARCO FRONTAL

- Este componente estructural alberga tanto a las puntas del eje delantero donde se asientan las ruedas, como al mecanismo de dirección y dos bases inferiores que conectarán a través de un perno los amortiguadores del tipo MacPherson para la suspensión frontal.



- Ensamblaje largueros (Predete
- Sensores
- Anotaciones
- Alzado
- Planta
- Vista lateral
- Origen
- (f) LARGUEROS<1> (Predet
- (-) hex bolt gradeb_iso<1>
- (-) hex bolt gradeb_iso<3>
- (-) hex flange nut gradea_is
- (-) hex flange nut gradea_is
- Relaciones de posición



El robot con todos sus componentes estructurales





**TENSIONES O ESFUERZOS A LOS
QUE ESTAN SOMETIDOS LOS
COMPONENTES ESTRUCTURALES Y
MECANICOS DEL ROBOT.**

Conceptos básicos de SolidWorks Simulation

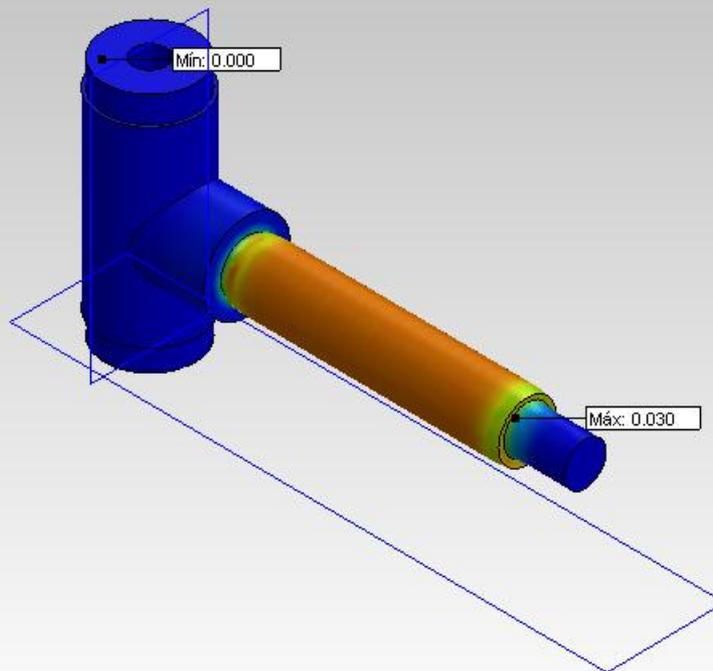
- SolidWorks® Simulation es un sistema de análisis de diseño que está completamente integrado con SolidWorks®. SolidWorks Simulation proporciona una solución de pantalla para los análisis de tensión, frecuencia, pandeo, térmicos y de optimización. Alimentado por programas de resolución de problemas rápidos (solvers), SolidWorks Simulation permite resolver grandes problemas de manera rápida utilizando su computadora personal. SolidWorks Simulation viene en varios paquetes para satisfacer sus necesidades de análisis.
- SolidWorks Simulation reduce el tiempo de salida al mercado ahorrando tiempo y esfuerzo en la búsqueda de resultados óptimos.

Ventajas del análisis

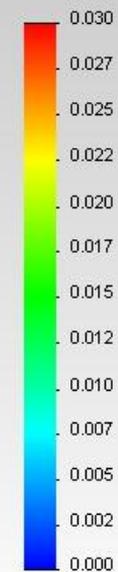
- Luego de construir su modelo, deberá asegurarse del eficaz funcionamiento de éste en el campo. Sin herramientas de análisis, esta tarea sólo se puede afrontar realizando ciclos de desarrollo de productos costosos y que insumen una gran cantidad de tiempo. Un ciclo de desarrollo de producto normalmente incluye los siguientes pasos:
- Construir el modelo.
- Construcción de un prototipo del diseño.
- Prueba del prototipo en el campo.
- Evaluación de los resultados de las pruebas de campo.

- Modificación del diseño basándose en los resultados de la prueba de campo.
- Este proceso continúa hasta que se haya alcanzado una solución satisfactoria. El análisis le puede ayudar a realizar las siguientes tareas:
- Reducir el costo simulando la prueba de su modelo en la computadora en lugar de realizar pruebas de campo costosas.
- Adelantar la comercialización del producto reduciendo el número de ciclos de desarrollo del mismo.
- Mejorar productos probando rápidamente múltiples conceptos y situaciones antes de tomar una decisión final, lo cual le proporciona más tiempo para idear nuevos diseños.

Nombre de modelo: terminal direccion
Nombre de estudio: Estudio_de_SimulationXpress
Tipo de resultado: Static tensión nodal Stress
Escala de deformación: 662083

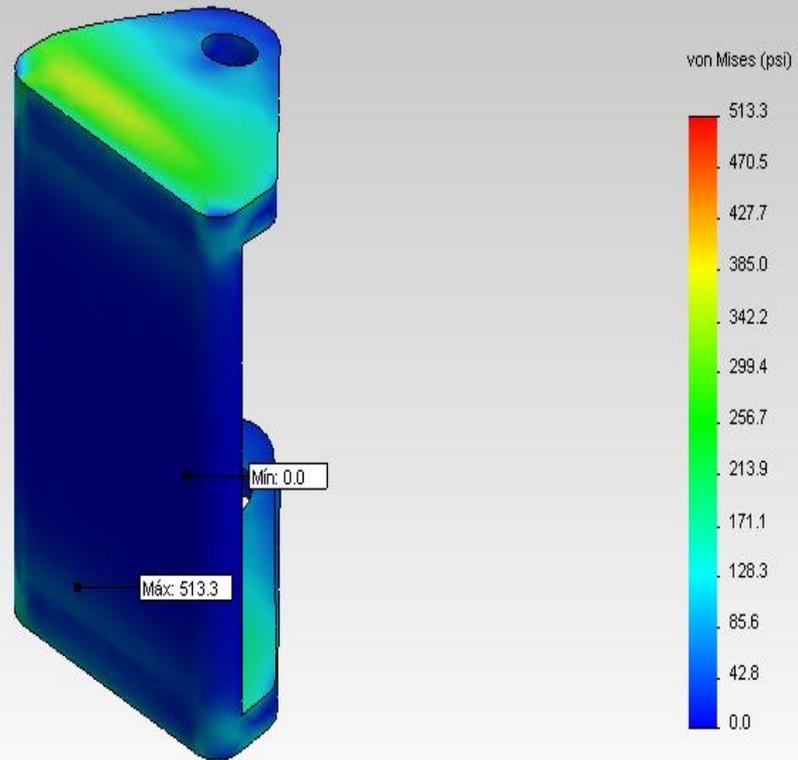


von Mises (N/mm² (MPa))



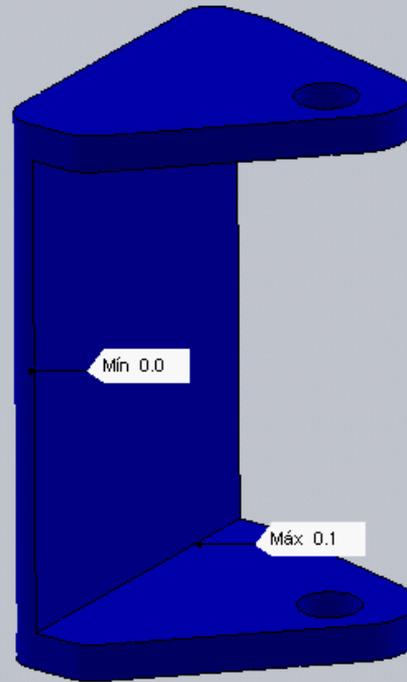
→ Límite elástico: 27.574

Nombre de modelo: Ensamblaje1
Nombre de estudio: Estudio 1
Tipo de resultado: Static tensión nodal Tensiones1
Escala de deformación: 1834.11

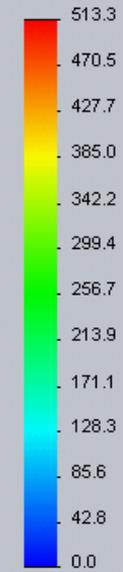


SIMULACION DE LAS TENSIONES

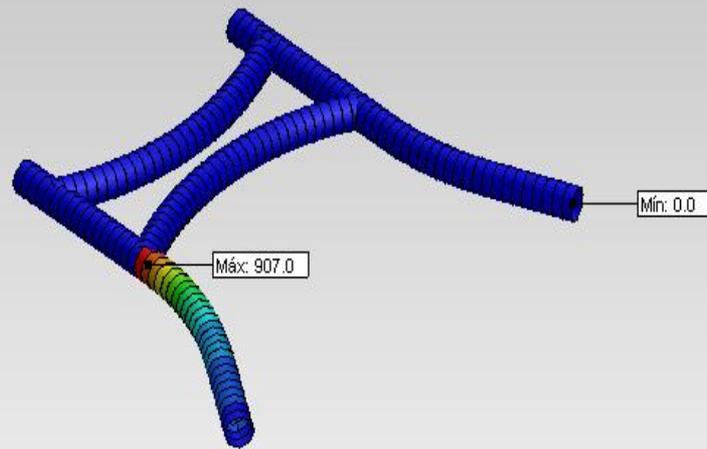
Nombre de modelo: Ensamblaje1
Nombre de estudio: Estudio 1
Tipo de resultado: Static tensión nodal Tensiones1
Escala de deformación: 1834.11



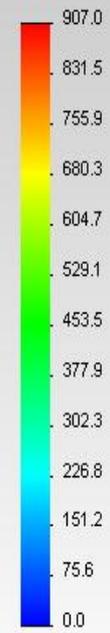
von Mises (psi)



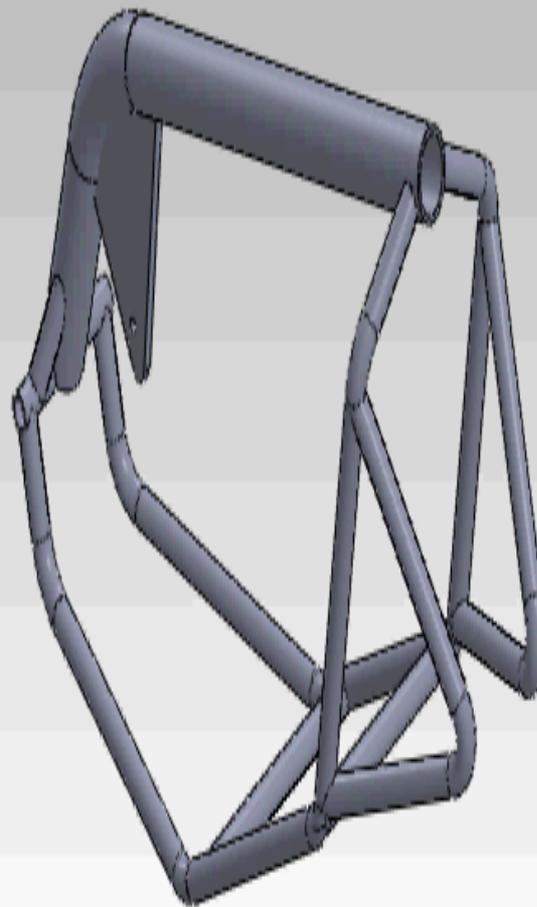
Nombre de modelo: LARGUEROS
Nombre de estudio: Estudio 1
Tipo de resultado: Tensión axial y de flexión más alta Tensiones1
Escala de deformación: 2052.35



Tensión axial y de flexión más alta (psi)



VISTA 3D DEL MARCO ESTRUCTURAL

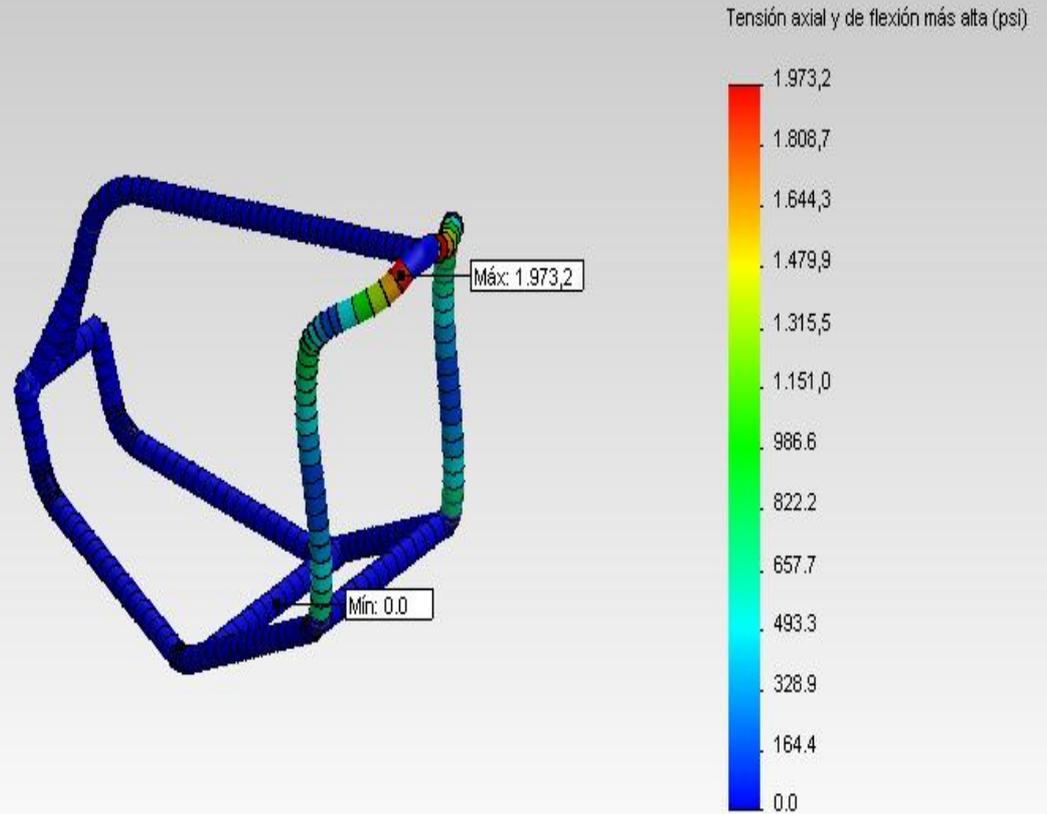


Nombre de modelo: marco estructural 2

Nombre de estudio: Estudio 2

Tipo de resultado: Tensión axial y de flexión más alta Tensiones1

Escala de deformación: 2857.47





**DISEÑO CONSTRUCCIÓN E
IMPLEMENTACIÓN DE
COMPONENTES ELÉCTRICOS,
ELECTRÓNICOS Y SOFTWARE DE
CONTROL.**

- 
- Las siguientes diapositivas se enfocarán en todos y cada uno de los componentes eléctricos y electrónicos con los que cuenta el robot, su diseño, características, y configuraciones; así como el desarrollo de software necesario para controlar los diferentes componentes del robot.

COMPONENTES ELÉCTRICOS Y ELECTRÓNICOS

- Algunos de ellos son: servomotores, motores eléctricos, componentes electrónicos de comunicación (X-BEE'S), de control (AVR'S) además de componentes presentes comúnmente en circuitos electrónicos como son: condensadores, transistores, diodos Zener, resistencias, etc.



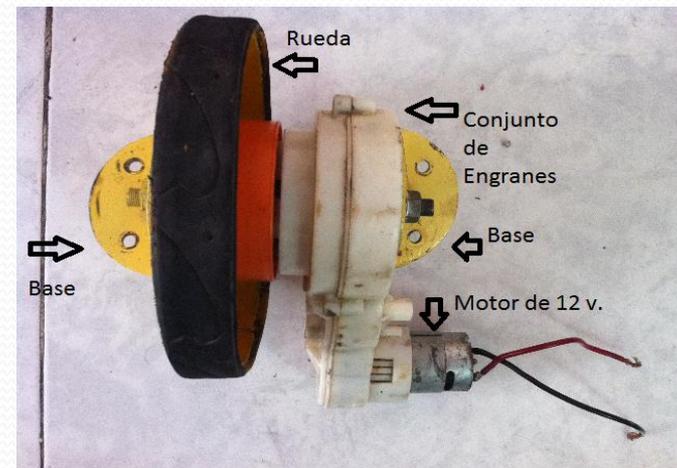
MOTOR USADO EN LA DIRECCIÓN.



XBEE



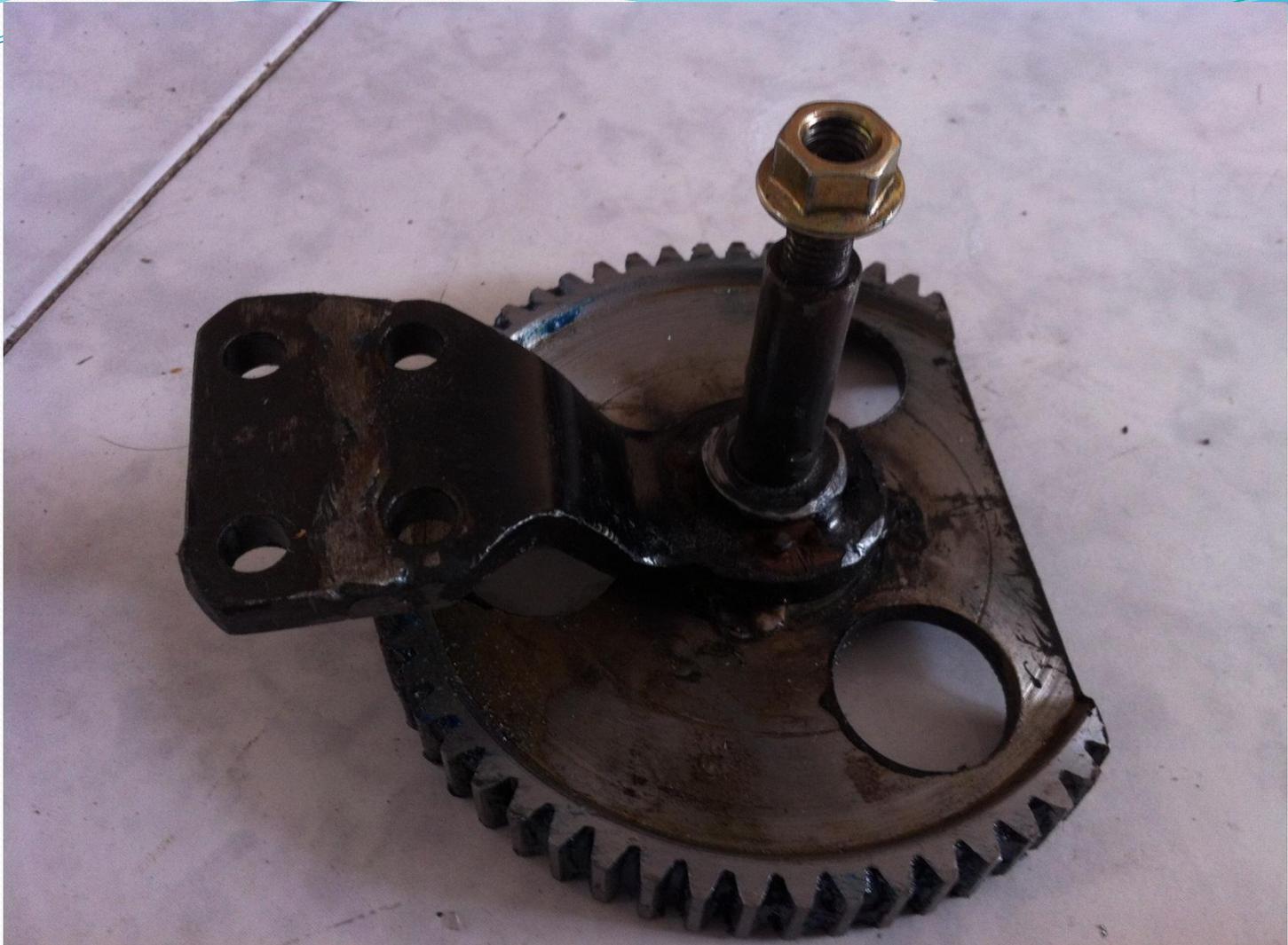
SERVO DE LA ACELERACIÓN



MOTOR SELECCIONADOR DE MARCHAS

SELECCIONA MIENTRO DEL MOTOR USADO EN LA DIRECCIÓN.

- Para el funcionamiento correcto de la dirección del robot se eligió un motor de plumas de camión de 12V. por sus características, ya que la dirección requiere un motor de torque alto y velocidad de giro baja, este posee en su interior un eje de tipo sin fin acoplado a un engranaje plástico, pese a ello no es suficiente para accionar la dirección, por lo cual fue necesario diseñar y fresar un engranaje recto adicional de diámetro 15 centímetros el cual sería instalado en el sistema de direccionamiento para su accionamiento final.



ENGRANAJE RECTO DE 15 CM DE DIAMETRO PARA LA DIRECCION

MOTOR ELÉCTRICO ENCARGADO DEL SELECCIONAMIENTO DE LAS MARCHAS.

Para el seleccionamiento de las marchas se utilizara un motor eléctrico de 12 V. el cual posee una particularidad, tiene adaptado a él, una serie de engranajes plásticos alineados de tal forma que permiten la multiplicación adecuada de la fuerza, y al final de estos engranajes una rueda de 18 cm. de diámetro acoplada, que en su diámetro exterior tiene soldado un cilindro de 1,5 cm de diámetro por 2.0 cm de altura, elemento final que accionara la palanca de cambios, cuando la rueda gire.



**PRINCIPIO DE FUNCIONAMIENTO
DEL MOTOR ELÉCTRICO
ENCARGADO DEL
SELECCIONAMIENTO DE LAS
MARCHAS.**

- El motor gira en ambos sentidos, es decir qué; cuando su polaridad es la normal, el (rojo) va al positivo de la batería y el (negro) va al negativo de la misma, este girará de forma horaria, y si se lo conecta inversamente es decir el cable (rojo) al negativo de la batería y el (negro) al positivo, este girará de manera anti horaria esto posible gracias al AVR encargado de hacerlo, que envía la señal para una u otra polaridad según el deseo del operador humano a través del control remoto.

- Logrando así que las marchas bajen o suban, logrando con esto escoger cualquiera de las cinco marchas disponibles (R-N-1-2-3); por ser la caja del motor una del tipo secuencial y sin embrague manual, no existe control electrónico para este.



**SERVOMOTOR UTILIZADO EN
ACCIONAR LA ACELERACIÓN DEL
ROBOT DE COMBATE.**

- El elemento encargado de la aceleración del robot será un servo motor de la marca HITEC Modelo HS-755HB.
- La razón para escoger este servo análogo en particular es su elevado torque, su velocidad de operación, tamaño y peso detallados a continuación
- El servo Hitec HS-755HB es un servo de tamaño $\frac{1}{4}$ de Escala y que proporciona un torque elevado, los engranes son de Karbonite los cuales son 4 veces más resistentes que los engranes de nylon.

Características a 4.8 Volts:

-Velocidad: 0.28 seg/60 grados; Torque: 11 Kg-cm.



Este servo motor, posee tres cables, uno rojo voltaje, otro negro tierra, y uno amarillo señal

CONECTIVIDAD INALÁMBRICA

Esta característica fundamental con la que será posible la comunicación entre el control remoto y el robot estará a cargo de los módulos Xbee, de la marca MaxStream estos proveen muchas ventajas, pueden ser configurados desde la computadora utilizando el software de computadora X-CTU y pueden comunicarse en configuraciones: punto a punto, punto a multipunto o en una red mesh.....(cont.)



que son redes de complejidad que permiten acceder a un punto remoto utilizando módulos intermedios para llegar como routers, en nuestro caso específico no usaremos ese tipo de configuración sino, la más sencilla que es de punto a punto, desde el Zigbee que se encuentra en el control remoto, hasta el Zigbee que se encuentra en el robot de batalla.

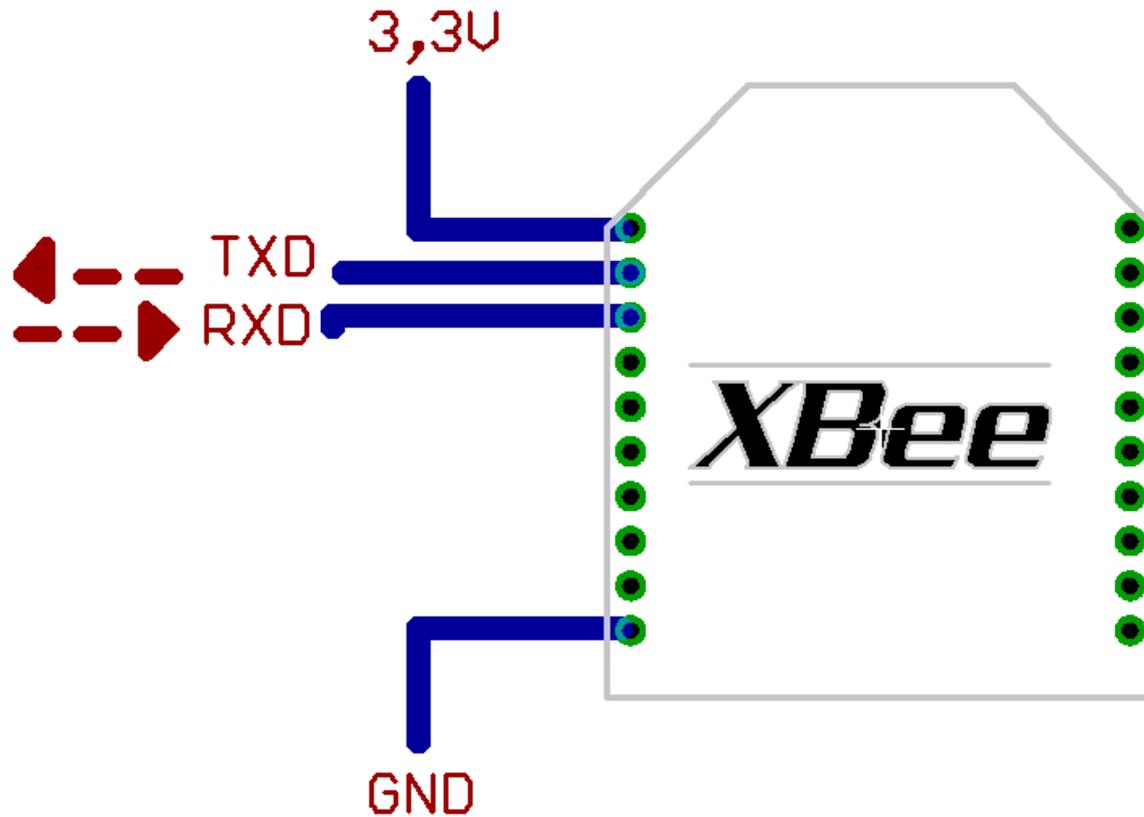
- Denominado también como "Zigbee" este es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE_802.15.4. creado por Zigbee Alliance, y permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas.

- La elección del módulo XBee correcto radica en escoger la potencia de transmisión, en el proyecto se utilizará los que poseen una potencia de 2mW para 91,44 m. de alcance aproximadamente en línea vista, y en interiores de unos 30 m. distancia más que necesaria para las aspiraciones del proyecto; las comunicaciones Zigbee se realizan en la banda libre de 2.4GHz y realiza las comunicaciones a través de una única frecuencia es decir de un canal, normalmente puede escogerse un canal de entre 16 posibles, la velocidad de transmisión de datos de una red Zigbee es de hasta 256kbps.

- Los módulos Zigbee cada uno de ellos tiene una dirección única de 64bits que viene grabada de fábrica; por otro lado la red Zigbee utiliza para sus algoritmos de ruteo direcciones de 16 bits para que cada vez que un dispositivo se asocia a una red Zigbee el coordinador al cual se asocia le asigna una dirección única en toda la red de 16 bits siendo el número máximo 65535 direcciones posibles.

CIRCUITO BÁSICO PARA EL XBEE

La Figura siguiente muestra las conexiones mínimas que necesita el módulo Xbee para poder ser utilizado, luego de esto se debe configurar según el modo de operación que se desea para la aplicación requerida por el usuario.



La primera conexión de la figura muestra el voltaje de alimentación de 3.3 V; la conexión a tierra y las dos líneas de transmisión de datos TXD (transmisión) y RXD (recepción) para comunicarse con un microcontrolador, en el caso del proyecto se comunicará con el AVR presente en el control remoto.

LA ELECTRÓNICA DE CONTROL.

- El cerebro del proyecto, el encargado del procesamiento de los datos de entrada y el que comanda las ordenes a realizarse, es el Micro Controlador ATMEGA48V -10PU de la marca ATMEL, a continuación se destacan algunas de sus principales características.

- Alto Performance, Low Power AVR® 8-Bit Micro controlador.
- Arquitectura Avanzada RISC.
- Programación no Volátil, memorias de almacenamiento de datos.
- Características Periféricas.
- Dos timers de 8 bits/contadores con pre escala separados y modo de comparación.
- Un timer de 16 bits/ contador con pre escala separados, modo de comparación y modo captura.
- Contador en tiempo real con oscilador separado.
- Seis Canales PWM.
- Ocho canales de 10 bits ADC en TQFP y paquete MLF.
- Seis Canales de 10 bits ADC en paquete PDIP.
- Programación serial USART.
- Interfaz serial SPI. Coordinador /Esclavo.
- Chip analógico Comparador.
- Características especiales del micro controlador.
 - Power-on Reset y detección programable Brown-out.

- Oscilador interno calibrado.
- Interrupción de recursos internos/externos.
- Cinco modos sleep.
- Pines de entrada y salida.
- 23 líneas de entrada y salida programables
- 28 pines PDIP.
- Voltaje de operación
- 1.8 - 5.5V
- Rango de Temperatura
- -40°C to 85°C
- Grado de Velocidad
- 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V
- Bajo consumo de Corriente
- Modo Activo: 1 MHz, 1.8V: 240µA
- 32 kHz, 1.8V: 15µA (incluido Oscilador)
- Modo de Bajo Consumo: 0.1µA at 1.8V

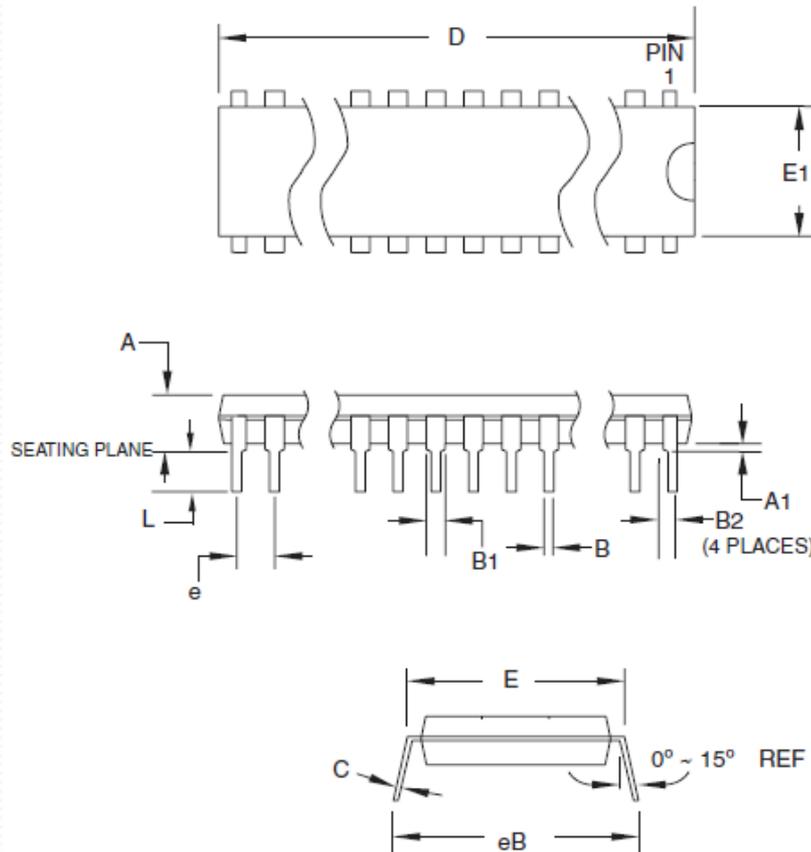
ATMEGA48V -10PU



Este micro procesador posee además las siguientes características:

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega88	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega168	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector

DIMENSIONES DEL MICRO PROCESADOR.

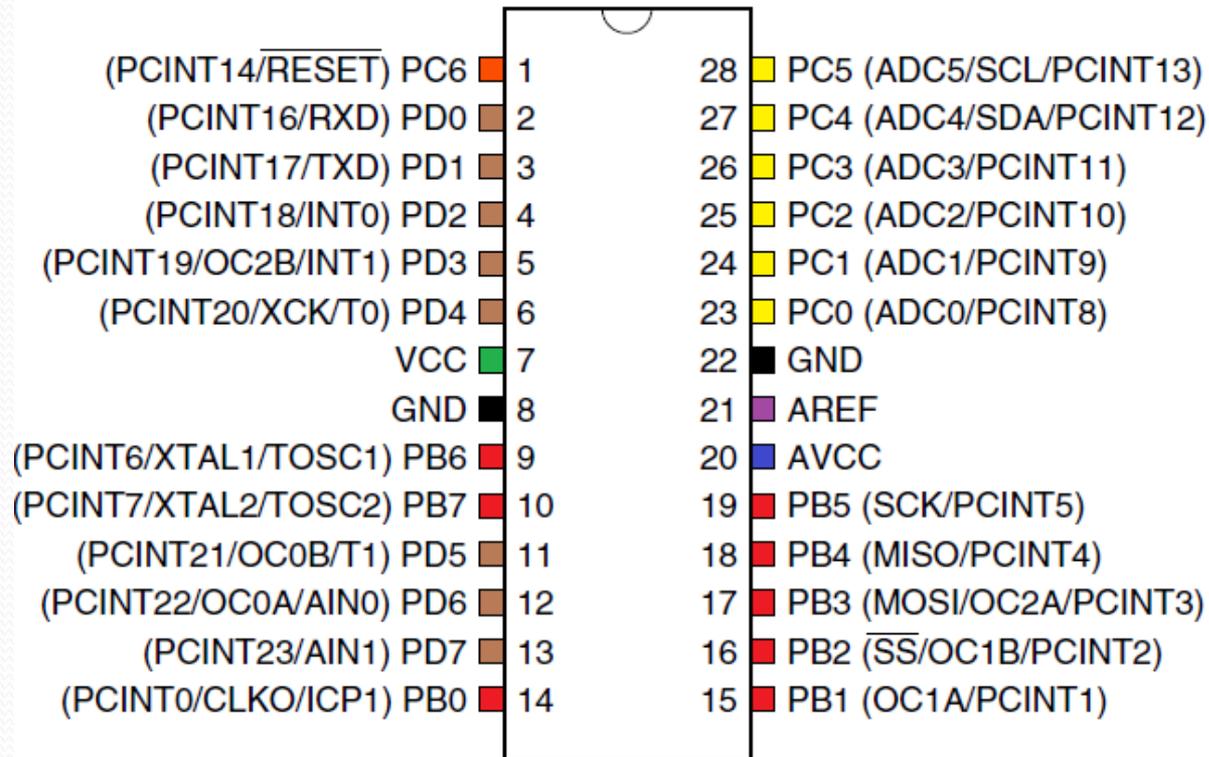


COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-	-	4.5724	
A1	0.508	-	-	
D	34.544	-	34.798	Note 1
E	7.620	-	8.255	
E1	7.112	-	7.493	Note 1
B	0.381	-	0.533	
B1	1.143	-	1.397	
B2	0.762	-	1.143	
L	3.175	-	3.429	
C	0.203	-	0.356	
eB	-	-	10.160	
e	2.540 TYP			

Disposición de Pines E/S

PDIP



PROGRAMACIÓN DE LOS MICRO CONTROLADORES.

- En esta parte del capítulo se detallará el diseño y la sintaxis de programación para los Micro Controladores, tanto del que se encuentra en el control remoto, como el que se halla en la tarjeta principal del robot de combate. Para ello es necesario usar el Software de programación para PC llamado Bascom AVR, el cual generará un archivo del tipo (*.bas), que al término de su diseño y luego de verificada su correcta compilación, producirá a su vez otro archivo del tipo (*.hex) que es el que finalmente irá al micro controlador, a través de una tarjeta programadora y la utilización de otro software que se detallará más adelante.

Logo del Software BASCOM AVR.



DISEÑO DEL ARCHIVO (*.BAS) PARA
EL MICROCONTROLADOR DEL
ATMEGA 48V -10PU UBICADO EN
EL CONTROL REMOTO A TRAVÉS
DEL SOFTWARE BASCOM AVR.

- Como primer paso damos inicio al programa y a continuación escogemos nuevo archivo como en cualquier otro software, y lo denominamos como nosotros queramos en el caso particular del proyecto se lo ha denominado control.bas luego de eso lo primero que se detalla gráficamente en la captura de pantalla en la parte inferior, es agregar la librería o los registros del micro para que el programa sepa con qué modelo en particular vamos a trabajar, ya que existen muchos tipos de micro controladores y diferentes números de serie para cada uno de ellos, colocamos el comando `$regfile="m48def.dat"` que equivaldría al ATMEGA 48V -10PU, luego de esto determinamos la velocidad del oscilador que serían 4MHz además de la velocidad de transmisión serial que serían 9600.

BASCOM-AVR IDE [2.0.7.1] - [C:\Users\PC1\Desktop\Robot batallador\Receptor\control.bas]

File Edit View Program Tools Options Window Help

remoto.bas control.bas

Sub Label

```
'configuro registros del microcontrolador
$regfile = "m48def.dat"

'la velocidad del oscilador
$crystal = 4000000

' velocidad de transmision serial
$baud = 9600

'declaracion de variables
```

- A continuación en la siguiente captura de pantalla se declara todas las variables, las primeras variables C, N, Aceleración, Izq (izquierda), Der (derecha), Subem (para subir la marcha), Bajam (para bajar la marcha) todas ellas se declaran como enteras; la variable S como una variable del tipo String (este tipo de variable puede almacenar letras o caracteres máximo 20); y la ultima variable B se configura como Byte.

'declaracion de variables

Dim C **As** Integer

Dim N **As** Integer

Dim Aceleracion **As** Integer

Dim Izq **As** Integer

Dim Der **As** Integer

Dim Subem **As** Integer

Dim Bajam **As** Integer

Dim S **As** String * 20

Dim B **As** Byte

- Se procede a configurar los pines del micro ubicado en el control remoto y se determina si estos son de entrada o salida, los pines del puerto C han sido asignados para la activación de los relés de la dirección y marchas, son pines de salida, al igual que lo los pines b0 y b1 que han sido designados para apagar o prender el motor del robot también esto a través de relés pero mucho más pequeños que los anteriores y los cuales se encuentran en la placa principal del robot; el puerto D7 asignado como de salida y es para el servo de la aceleración; el único dato entrante es el pin D5 que capta la señal del sensor electromagnético de las marchas ubicado en el robot, este sirve para saber cuándo la rueda que se encarga de los cambios ha dado una revolución completa, efectuando así un cambio ya sea este hacia arriba o hacia abajo, evitando así que gire infinito número de veces.

```
'configuro pines del microcontrolador
```

```
'*****salidas
```

```
Config PORTC = Output
```

```
Config PORTE = Output
```

```
Config PORTD.7 = Output
```

```
Config PORTD.5 = Input
```

```
'direccion y marchas
```

```
'reles b0=apagar y b1=prender
```

```
'servo acelerador
```

```
'sensor magnetico
```

```
c - ""
```

A continuación se escribe `S = ""` para inicializar variables en blanco y para almacenar la cadena de datos desde el puerto serial.

```
S = ""  
C = 0  
Aceleracion = 0  
Izq = 0  
Der = 0  
Subem = 0  
Bajam = 0
```



Seguido a esto se habilita las interrupciones; la interrupción es una técnica que coloca al programa temporalmente en suspenso mientras el microcontrolador ejecuta otro conjunto de instrucciones en respuesta a un suceso; en el caso específico del proyecto cuando se envíe un dato al micro

-este va a estar atento para recibir los datos de las interrupciones; a continuación se habilita la interrupción con el comando : **On Urxc Rec_isr** esto hace que cuando se envíe un dato desde el control automáticamente ingrese a dicha interrupción, y finalmente se habilita la recepción de datos por el puerto serial digitando el comando: **Enable Urxc**

```
Enable Interrupts  
On Urxc Rec_isr  
Enable Urxc
```

- Asigno los números que quiero que correspondan a cada una de las acciones a realizarse, antecediendo un apostrofe para indicar que no es parte de la programación sino solamente con medios informativos; por ejemplo cuando se presione el pulso del encendido del motor el micro controlador enviará un número cuatro y así como sucesivamente como esta en la captura de pantalla para las de más acciones.

```
'4 encendido pulso  
'3 prender y apagar  
'6 bajo marcha  
'5 subo marcha  
'2 iz  
'1 derecha
```

Al iniciar la programación a través del comando **Reset** coloco en cero las salidas de la dirección (puertos C5 y C4) además de las marchas (C3 y C2).

```
Inicio:
```

```
'*****'
```

```
' seteo a cero las salidas de direccion y marchar
```

```
'direccion 2 puertos
```

```
Reset PORTC.5
```

```
Reset PORTC.4
```

```
'marchas 2 puertos
```

```
Reset PORTC.3
```

```
Reset PORTC.2
```

ACELERACION

Para la aceleración creo un lazo del tipo **FOR**, que es un ciclo repetitivo, y le doy valores de uno hasta quinientos en intervalos de uno, además le doy un comando **Pulseout** que es prácticamente un generador de onda y le asigno el puerto D7 junto con la variable **Aceleración**, para finalizar le doy un tiempo de espera de 10ms y cierro el lazo externo **If**.

```
'*****  
If Aceleracion > 100 Then  
    For N = 1 To 500 Step 1  
        Pulseout PORTD , 7 , Aceleracion  
        Waitus 10  
    Next  
End If
```

DIRECCIONAMIENTO

- Para activar el relé de la dirección cuando quiera girar a la izquierda, hago que la variable **Izq** tenga el valor de uno y active el puerto C2, espere 150 ms y si el valor es cero no realice acción alguna, finalizo el lazo **If**.

```
If Izq = 1 Then
    'activo 1 rele de direccion
    PORTC.2 = 1
    Waitms 150
    PORTC.2 = 0
    Izq = 0
End If
```

- De idéntica forma cuando quiera que gire a la derecha hare que el valor sea uno apretando el botón del control, haciendo que la variable **Der** tenga el valor de uno y active el puerto C3 espere 150 ms y si el valor es cero no realice acción alguna, finalizo el lazo **If**.

```
If Der = 1 Then
    PORTC.3 = 1
    Waitms 150
    PORTC.3 = 0
    Der = 0
End If
```

CAMBIO DE MARCHAS

- Las siguientes líneas de programación corresponden a la activación del relé para subir las marchas, cuando la variable denominada **Subem** tiene un valor de uno, se activa el puerto C4 y espera quinientos mili segundos entonces ingresa al bucle **Loop Until** que es una instrucción que se ejecuta hasta que el comando de prueba se ejecuta correctamente, que en este caso sería hasta que la señal del sensor electromagnético llegue a través del puerto D5, luego de eso seteo a cero la salida del puerto C4 y también lo hago con la variable **Subem**, cierro el lazo con el **End If**.

```
If Subem = 1 Then
    'activo un rele de marcha para subir
    PORTC.4 = 1
    Waitms 500
    Do
        'espera q de una vuelta completa hasta que el sensor magnetico sea uno
    Loop Until PIND.5 = 1
    PORTC.4 = 0 'seteo la salida a cero
    Subem = 0 'la variable para subir marcha seteo a cero
End If
```

- De manera análoga líneas de programación corresponden a la activación del relé para bajar las marchas, cuando la variable denominada **Bajam** tiene un valor de uno, se activa el puerto C5 y espera quinientos mili segundos entonces ingresa al bucle **Loop Until** que es una instrucción que se ejecuta hasta que el comando de prueba se ejecuta correctamente, que en este caso sería hasta que la señal del sensor electromagnético llegue a través del puerto D5, luego de eso seteo a cero la salida del puerto C5 y hago lo mismo con la variable **Bajam**, cierro el lazo con el **End If** y envío a Inicio.

```
If Bajam = 1 Then
    'activo un rele de marcha para bajar
    PORTC.5 = 1
    Waitms 500
    Do
        'espera q de una vuelta completa hasta que el sensor magnetico sea uno
    Loop Until PIND.5 = 1
    'seteo la salida a cero
    PORTC.5 = 0
    'la variable para bajar marcha seteo a cero
    Bajam = 0
End If
```

```
Goto Inicio
```

INTERRUPCIÓN

- Continúo diseñando la interrupción al inicio mencionada, almacenando en B los datos recibidos por el puerto serial cuando envío datos desde el control remoto con el comando Rec_isr ; además de agregar la instrucción que lee los datos del puerto serial y los almacena en la variable B.

BASCOM-AVR IDE [2.0.7.1] - [C:\Users\PC1\Desktop\Robot batallador\Receptor\control.bas]

File Edit View Program Tools Options Window Help



remoto.bas

control.bas

Sub

Label

```
Goto Inicio
```

```
Rec_isr:
```

```
' almaceno en B los datos que recibe x el puerto serial cuando envio datos del control remoto
```

```
B = UDR
```

```
If B = 10 Or B = 13 Then ' comando que lee los datos del puerto serial y almaceno en la variable B
```

- Datos que pueden ser 10 ó 13 (siendo estos códigos ASCII decimales) los cuales corresponden a un fin de línea (line feed) y salto de línea (carriage return) respectivamente.

Char	Oct	Dec	Hex	Control-Key	Control Action
NUL	0	0	0	^@	NUL character
SOH	1	1	1	^A	Start Of Heading
STX	2	2	2	^B	Start of TeXt
ETX	3	3	3	^C	End of TeXt
EOT	4	4	4	^D	End Of Transmission
ENQ	5	5	5	^E	ENQuiry
ACK	6	6	6	^F	ACKnowledge
BEL	7	7	7	^G	BELI, rings terminal bell
BS	10	8	8	^H	BackSpace (non-destructive)
HT	11	9	9	^I	Horizontal Tab (move to next tab position)
LF	12	10	a	^J	Line Feed
VT	13	11	b	^K	Vertical Tab
FF	14	12	c	^L	Form Feed
CR	15	13	d	^M	Carriage Return

- Dentro de la interrupción asignamos varios casos, el primero el caso 4 es el encargado de apagar el motor, cuando el puerto Bo se activa o es igual a uno, se envía la señal a la placa principal del robot, para que se enclave el relé encargado de apagar el motor, luego de ello se le asigna un tiempo de espera de 1000 ms. y si el puerto Bo no recibe nada pues no realice ninguna acción, por ultimo se setea la variable en blanco con el comando $S = ""$

```
Select Case S
  Case "4":
    ' activo puerto b0 para apagar
    PORTB.0 = 1
    Waitms 1000
    PORTB.0 = 0
    'seteo la variable s en blanco
    S = ""
  Case "3":
    ' activo puerto b1 para encender
    PORTB.1 = 1
    Waitms 1000
    PORTB.1 = 0
    S = ""
  Case "1":
    Izq = 1
    S = ""
  Case "2":
    Der = 1
    S = ""
  Case "5":
    Subem = 1
    S = ""
  Case "6":
    Bajam = 1
    S = ""
```

- A continuación para el siguiente caso, el caso 3 que es el encargado del encendido del motor que usa el puerto B1 para enclavar un relé de la placa principal ubicada en el robot se hace exactamente lo mismo que para apagarlo como en el caso anterior cambiando nada más que el puerto.
- Para el siguiente caso el caso 1 que equivaldría a girar a la izquierda, lo único que se hace es enviar la orden si se requiere de esta acción al

- 
- presionar el botón correspondiente, así como para cuando se quiera igualmente girar a la derecha (caso 2), subir marchas (caso 5) y bajar marchas(caso 6).

INTERVALOS PARA EL ACELERADOR ANALÓGICO

- Los siguientes y últimas líneas de programación del micro controlador son para el control remoto, que por ser analógico posee un potenciómetro interno que lo que hace es captar la posición de aceleración e ir variando la señal así pues cuando el caso sea el caso A, la señal de aceleración será la mínima con un valor de 350, pasando por los demás casos hasta llegar a la máxima que es la del caso E con un valor de 770. Se le asigna además un valor para cuando el estado sea cero, para cuando no exista aceleración alguna.

```
Case "A":  
    Aceleracion = 350  
    '350  
    S = ""  
Case "B":  
    Aceleracion = 450  
    S = ""  
Case "C":  
    Aceleracion = 550  
    S = ""  
Case "D":  
    Aceleracion = 650  
    S = ""  
Case "E":  
    Aceleracion = 770  
    S = ""  
Case "0":  
    Aceleracion = 0  
    S = ""  
Case Else:  
    S = ""  
End Select  
Else  
    S = S + Chr(b)  
End If  
Return  
  
End
```

COMPILACION DEL ARCHIVO

- Una vez terminado el programa se compila para verificar errores presionando (F7) y se guarda el archivo en cualquier parte de la computadora para luego utilizarlo en la programación final del micro controlador.

BASCOM-AVR IDE [2.0.7.1] - [C:\Users\PC1\Desktop\Robot batallador\Receptor\control.bas]

File Edit View Program Tools Options Window Help

remoto.bas control.bas  Compile program (F7)

Sub Label

```
'configuro registros del microcontrolador
$regfile = "m48def.dat"

'la velocidad del oscilador
$crystal = 4000000

' velocidad de transmision serial
$baud = 9600
'declaracion de variables
Dim C As Integer
Dim N As Integer
Dim Aceleracion As Integer
Dim Izq As Integer
Dim Der As Integer
```

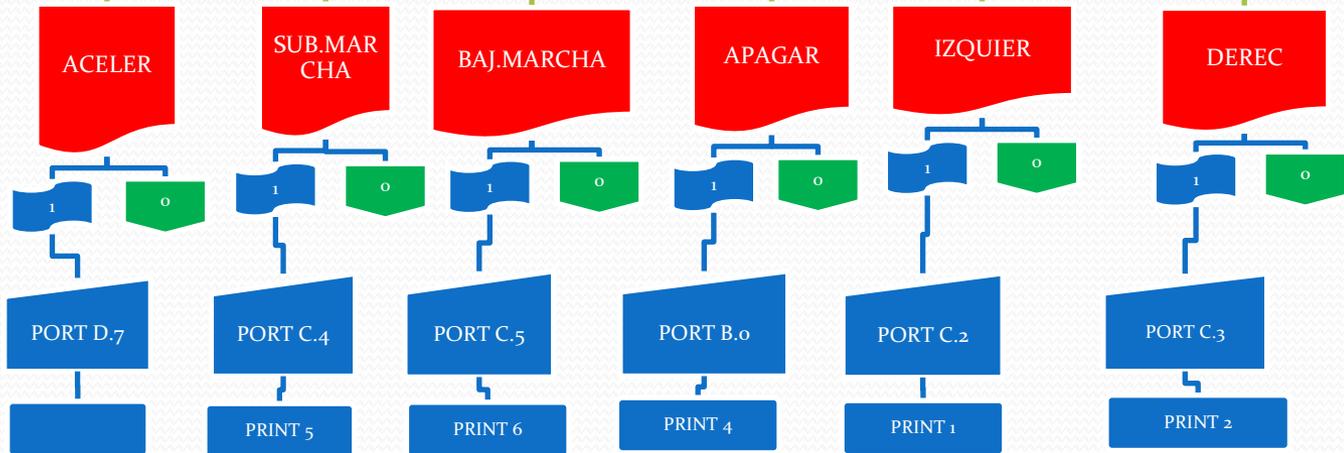
DISEÑO DEL ARCHIVO (*.BAS) PARA
EL MICROCONTROLADOR DEL
ATMEGA 48V -10PU UBICADO EN
LA **PLACA PRINCIPAL DEL ROBOT** A
TRAVÉS DEL SOFTWARE BASCOM
AVR.



DIAGRAMA DE FLUJO DEL PROGRAMA

INICIO

INGRESO DE
REQUERIMIENTOS
A TRAVES DEL CONTROL
REMOTO



- Como primer paso damos inicio al programa y a continuación escogemos nuevo archivo como en cualquier otro software, y lo denominamos como nosotros queramos en el caso particular del proyecto se lo ha denominado remoto.bas luego de eso lo primero que se detalla gráficamente en la captura de pantalla en la parte inferior, es agregar la librería o los registros del micro para que el programa sepa con qué modelo en particular vamos a trabajar, ya que existen muchos tipos de micro controladores y diferentes números de serie para cada uno de ellos, colocamos el comando `$regfile="m48def.dat"` que equivaldría al ATMEGA 48V -10PU, luego de esto determinamos la velocidad del oscilador (cristal) que serían 4MHz además de la velocidad de transmisión serial que serían 9600.

BASCOM-AVR IDE [2.0.7.1] - [C:\Users\PC1\Desktop\Robot batallador\Emisor\remoto.bas]

File Edit View Program Tools Options Window Help



remoto.bas

Sub

Label

```
$regfile = "m48def.dat"  
'la velocidad del oscilador para que vaya de linea en linea  
$crystal = 4000000  
  
' velocidad de transmision serial  
$baud = 9600
```

Seguido a esto declaro las variables y determino su tipo dándole a la variable B el tipo de Byte y a las variables C y Aux como enteros; a continuación configuro los puertos como puertos de entrada que van desde el puerto D2 al D7.

```
'declaro variables
Dim B As Byte
Dim C As Integer
Dim Aux As Integer

' configuro puertos
Config PORTD.2 = Input
Config PORTD.3 = Input
Config PORTD.4 = Input
Config PORTD.5 = Input
Config PORTD.6 = Input
Config PORTD.7 = Input
```

- Habilito la conversión analógica-digital y las interrupciones; además de ello con el comando **Start ADC** doy inicio a la conversión antes mencionada, y defino a Lec como una variable del tipo Word que ocupa un espacio de memoria más grande como la encargada de almacenar los datos leídos a través del puerto analógico, datos correspondientes a la aceleración desde el control remoto.
- Le doy valores iniciales a la variable Lec y Aux =0 para que permita solo el envío de un solo dato cuando el potenciómetro esta accionado, para complementar le asigno un valor de $C = 300$ como antirebote.

```
'  
Config ADC = Single , Prescaler = Auto
```

```
Enable Interrupts
```

```
'comienzo la conversión analógica digital
```

```
Start ADC
```

```
'variable en la que se almacena los datos leídos por el puerto analógico
```

```
Dim Lec As Word
```

```
'seteo valores iniciales de las variables
```

```
    Lec = 0
```

```
    'variable para enviar solo un dato cuando este posicionado el potenciómetro
```

```
    Aux = 0
```

```
' antirebote
```

```
C = 300
```

- La aceleración analógica del robot está dividida en cinco intervalos, lo que equivale a que cada intervalo le corresponda un 20% de la aceleración total.
- Al inicializarse el programa lo primero que se hace es leer que valor existe en el potenciómetro y luego almacenarlo en la variable **lec**, si la lectura esta entre 0 y es menor a 550 (0% de aceleración) lo que correspondería al inicio de la aceleración, se envía por el puerto serial un "0", además no se le asigna ningún tiempo de espera y se termina el lazo If.

Inicio:

```
'*****  
'leo el potenciómetro y almaceno en la variable lec  
Lec = Getadc(5)  
  
If Lec < 550 Then  
    If Aux = 1 Then  
        'comando para enviar datos por el puertop serial  
        Print "0"  
        Waitms 10  
        Aux = 0  
    End If  
End If
```

- En caso de que la lectura o variable (lec) del acelerador esté entre los valores comprendidos desde los 550 hasta los 650 se envía una "A" (20% de aceleración) y se espera 100 ms.

```
If Lec > 550 AND Lec < 650 Then
  Print "A"
  Waitms 100
  'variable donde me indica que fue enviado un dato
  Aux = 1
End If
```

- Si la lectura o variable (lec) del acelerador esté entre los valores comprendidos desde los 650 hasta los 750 se envía una "B" (40% de aceleración) y se espera de igual manera 100 ms.

```
If Lec > 650 AND Lec < 750 Then  
    Print "B"  
    Waitms 100  
    AUX = 1  
End If
```

- Si la lectura o variable (lec) del acelerador esté entre los valores comprendidos desde los 750 hasta los 850 se envía una "C" (60% de aceleración) y se espera de igual manera 100 ms.

```
End If
If Lec > 750 AND Lec < 850 Then
  Print "C"
  Waitms 100
  Aux = 1
End If
```

- Si la lectura o variable (lec) del acelerador esté entre los valores comprendidos desde los 850 hasta los 950 se envía una "D" (80% de aceleración) y se espera de igual manera 100 ms.

```
If Lec > 850 AND Lec < 950 Then  
    Print "D"  
    Waitms 100  
    Aux = 1  
End If
```

- Si la lectura o variable (lec) del acelerador esté entre los valores comprendidos desde los 950 hasta los 1040 se envía una "E" (100% de aceleración) y se espera de igual manera 100 ms.

```
If Lec > 950 AND Lec < 1024 Then  
    Print "E"  
    Waitms 100  
    Aux = 1  
End If
```

- Para que sea posible leer la información proveniente desde el control remoto, en cuanto a solicitudes de direccionalidad (izquierda y derecha); seleccionamiento de marchas (subir y bajar marchas); encendido y apagado de motor, seis lecturas en total es necesario configurar los pines del micro controlador ubicado en la placa principal desde el D2 hasta el D7 de la manera que se detalla a continuación.

```
'lectura de pulsadores del control remoto
```

```
  If PIND.2 = 1 Then
```

```
    Waitms 100
```

```
    Print "1"
```

```
  End If
```

```
  If PIND.3 = 1 Then
```

```
    Waitms 100
```

```
    Print "2"
```

```
  End If
```

```
  If PIND.4 = 1 Then
```

```
    Waitms C
```

```
    Print "3"
```

```
  End If
```

```
  If PIND.5 = 1 Then
```

```
    Waitms C
```

```
    Print "4"
```

```
  End If
```

```
  If PIND.6 = 1 Then
```

```
    Waitms C
```

```
    Print "5"
```

```
  End If
```

```
  If PIND.7 = 1 Then
```

```
    Waitms C
```

```
    Print "6"
```

```
  End If
```

```
Goto Inicio
```

```
End
```

COMPILACION DEL ARCHIVO

- Una vez terminado el programa se compila para verificar errores presionando (F7) y se guarda el archivo en cualquier parte de la computadora para luego utilizarlo en la programación final del micro controlador.

BASCOM-AVR IDE [2.0.7.1] - [C:\Users\PC1\Desktop\Robot batallador\Emisor\remoto.bas]

File Edit View Program Tools Options Window Help

remoto.bas

Compile program (F7)

Sub Label

```
$regfile = "m48def.dat"  
'la velocidad del oscilador para que vaya de linea en linea  
$crystal = 4000000  
  
' velocidad de transmision serial  
$baud = 9600
```

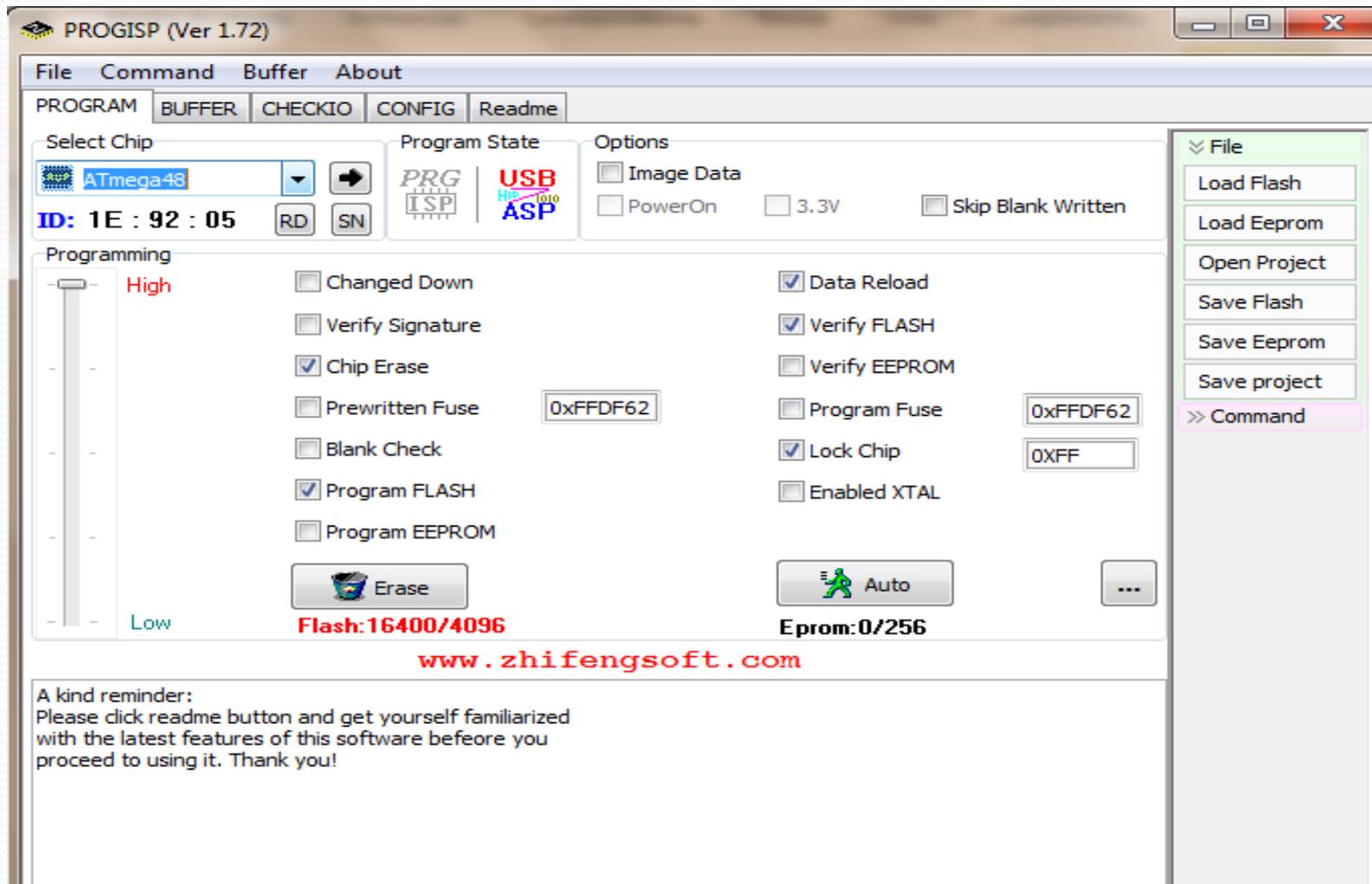
ENVÍO DEL ARCHIVO COMPILADO
(* .HEX) A TRAVÉS DEL SOFTWARE
DE PC PROGISP AL MICRO
CONTROLADOR CON LA
UTILIZACIÓN DE UNA TARJETA
PROGRAMADORA DE AVR'S CON
CONEXIÓN USB.

- Una vez que se finaliza con la programación de los dos micros, necesitamos pasar el archivo de tipo (*.hex) a los mismos para su programación final, este archivo se creó automáticamente cuando terminamos la programación anterior.
- La manera de pasar estos archivos es conectando los micros a una programadora de AVR's similar a la que se puede ver a continuación conectada al robot y de fácil adquisición.

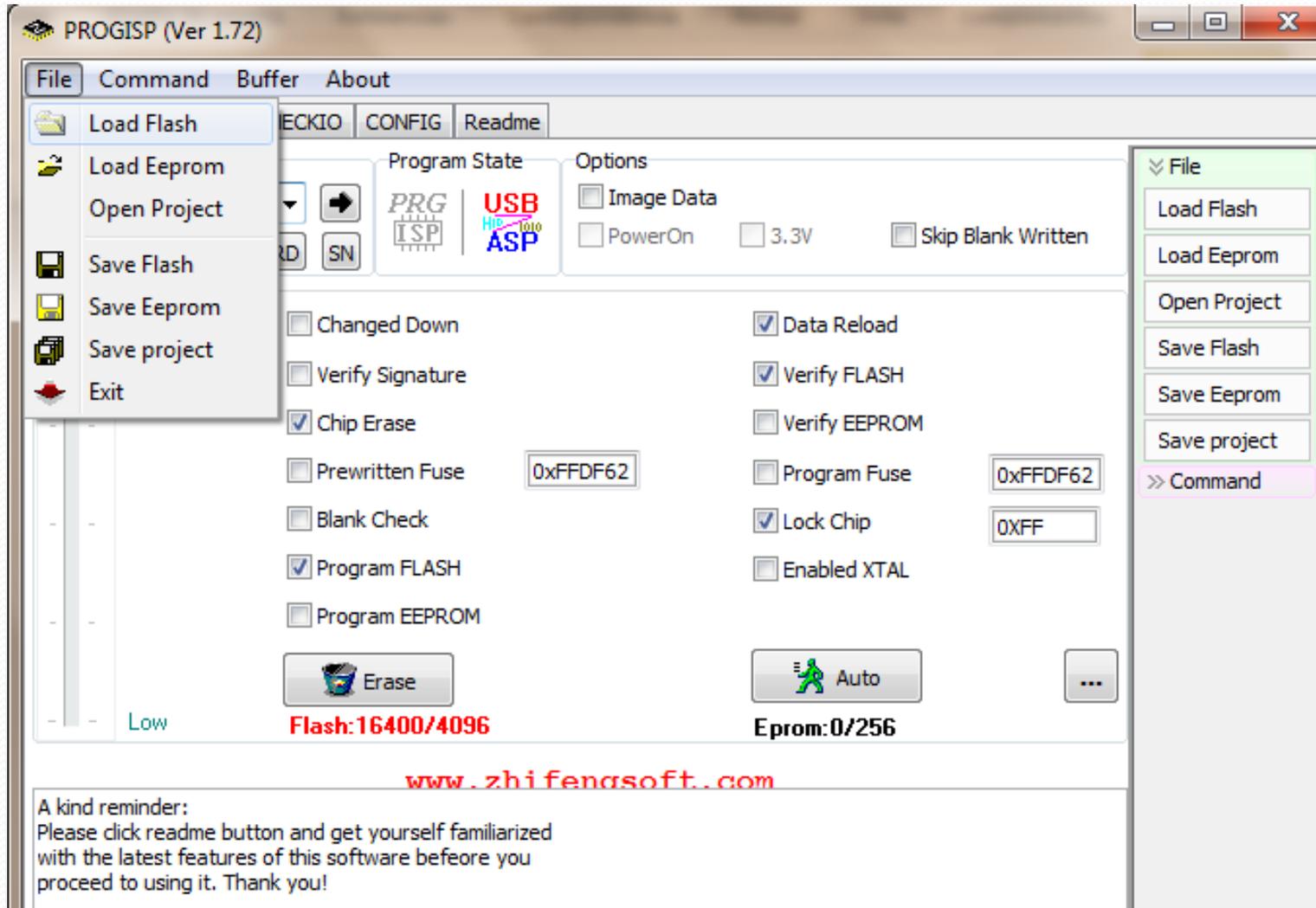


Tarjeta programadora de AVR's conectada a la placa principal

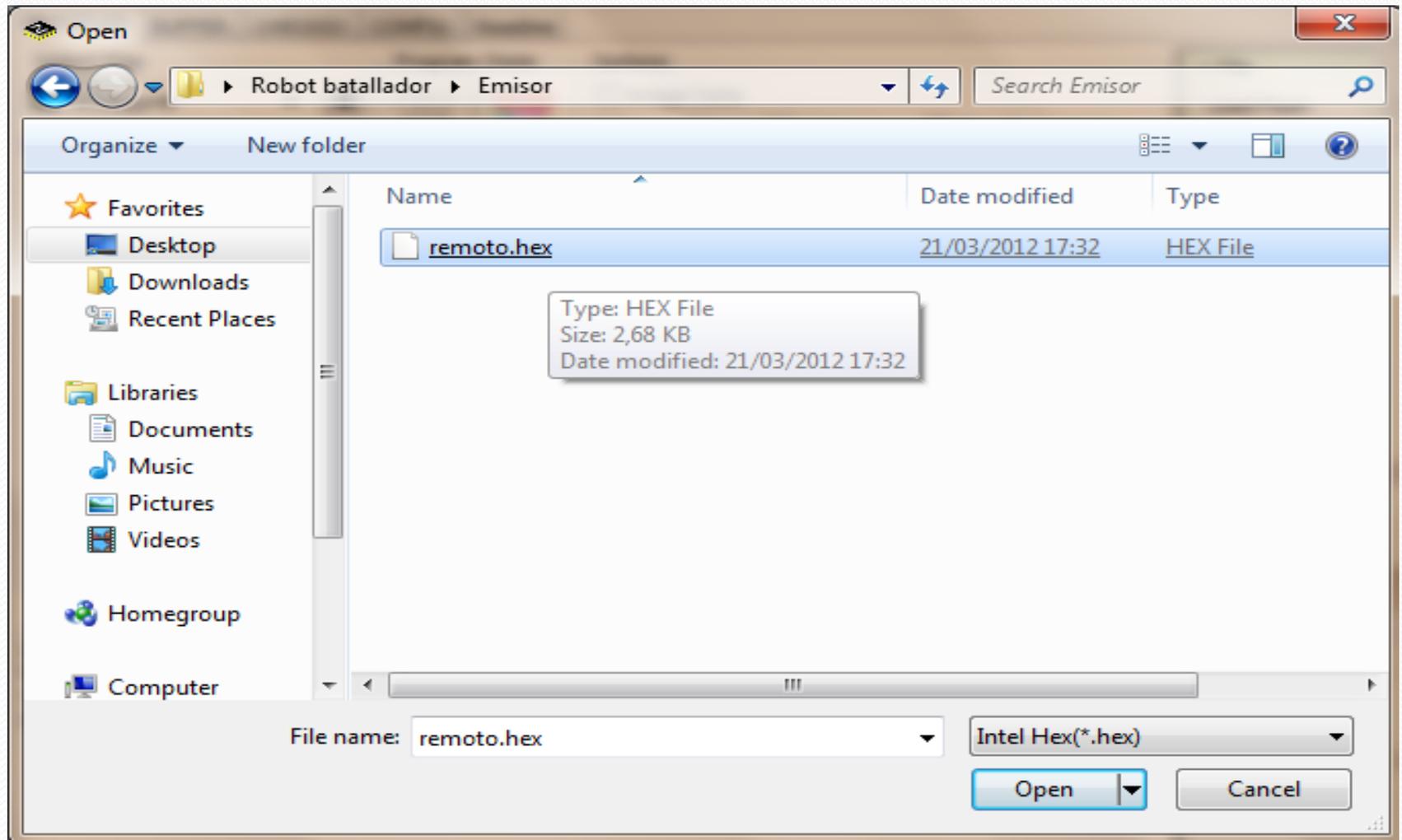
- Primero se abre el PROGISP y se selecciona el MICRO; en este caso el ATmega48 como en la siguiente captura de pantalla.



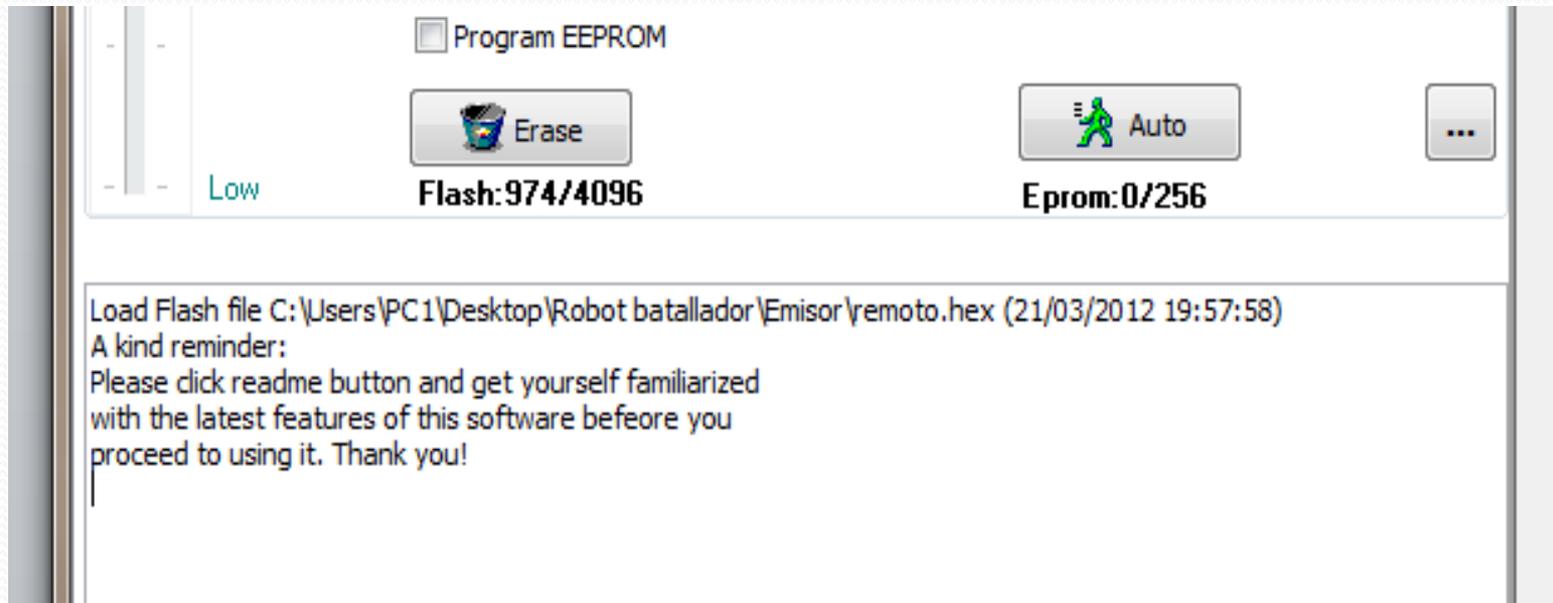
- A continuación se abre el menú File y se selecciona la opción Load Flash.



- Se abrirá el siguiente menú donde seleccionaremos la ubicación del respectivo archivo *.hex



- Una vez que hagamos correctamente la programación del micro saldrá el siguiente mensaje:



Luego Seleccionamos la opción Auto para programar el micro controlador y listo el AVR estará programado correctamente.

También existe la opción de borrar el micro, leerlo, etc.



DISEÑO Y FABRICACIÓN DE LAS PLACAS ELECTRÓNICAS CON USO DEL SOFTWARE EAGLE.

- Una vez que hemos programado los micro controladores AVR con los archivos *.hex están listo para ponerlos en las placas electrónicas junto con los demás elementos.

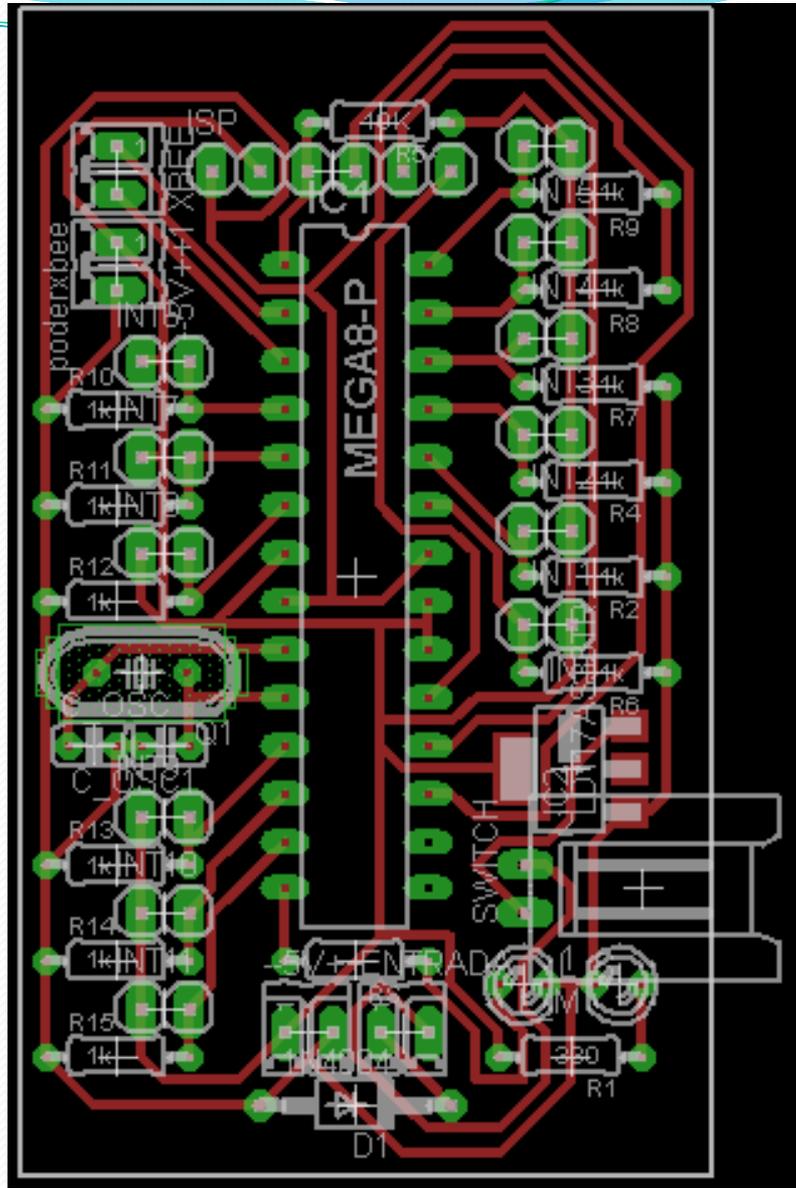
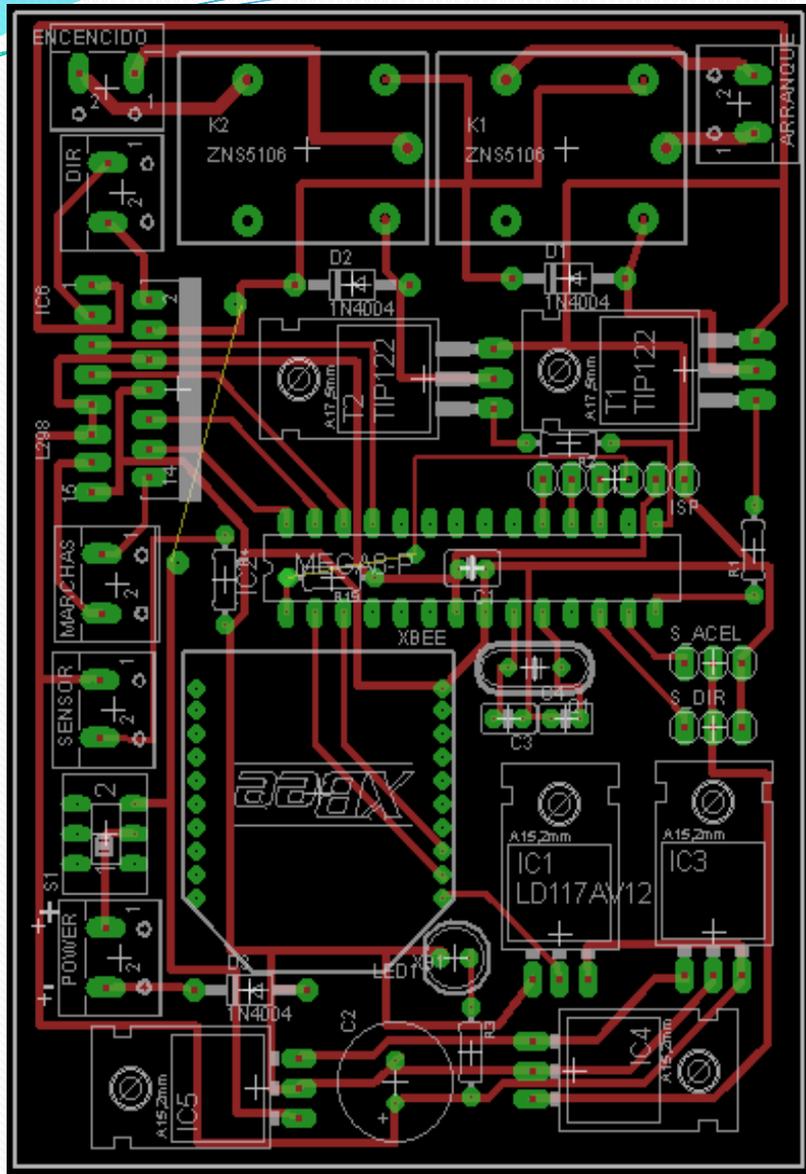


- 
- Para ello existe este software que nos permite seleccionar todos y cada uno de los componentes de manera virtual.

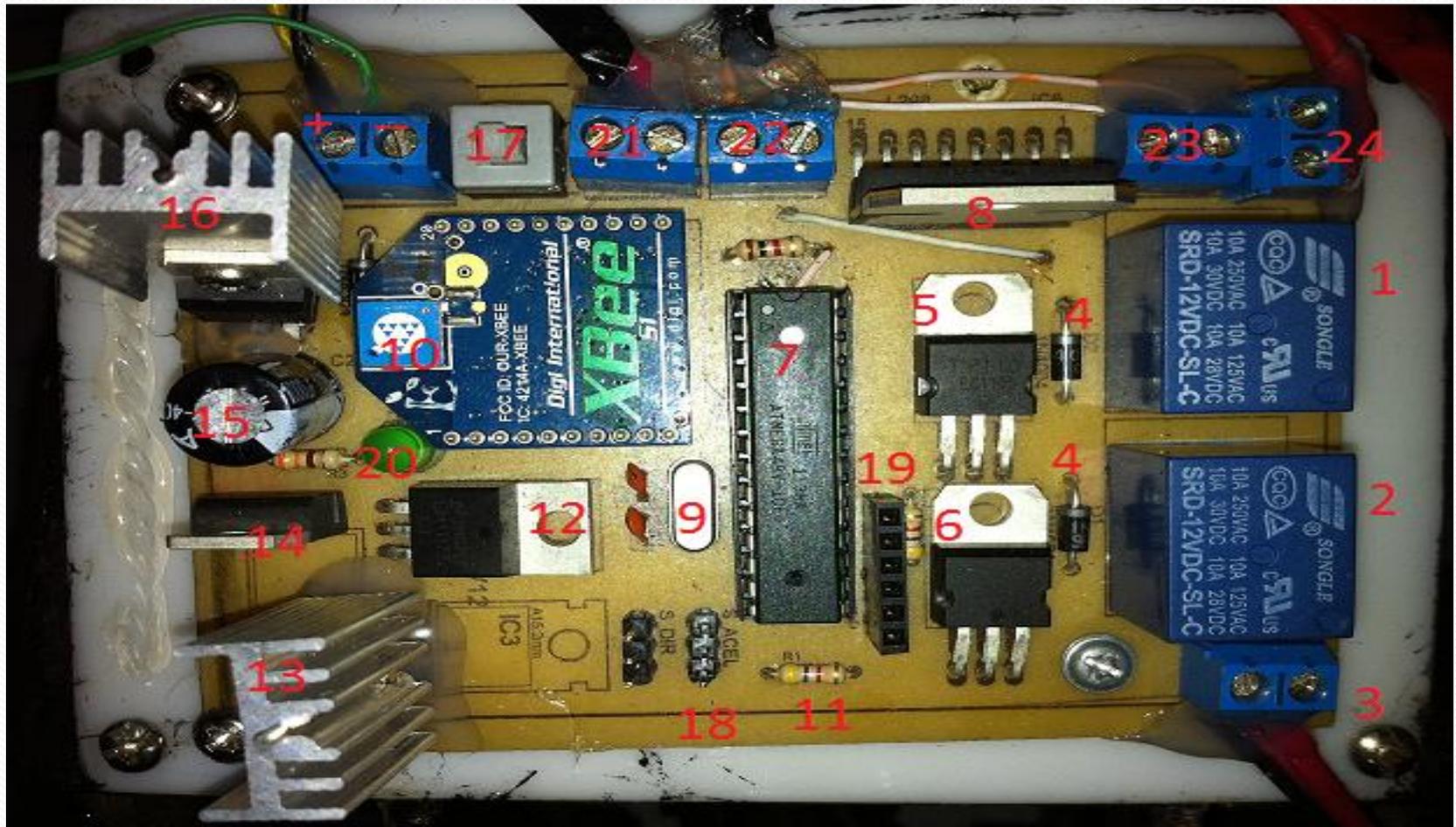


PROCESO PARA LA FABRICACIÓN DE LAS PISTAS EN LA BAQUELITA

- Una vez que hayamos hecho esto el programa ruteará las pistas y nos diseñará las mismas para luego imprimirlas en papel fotográfico con una impresora láser, después de ello con la ayuda de una plancha casera a través del calor de la misma hacer que las pistas por donde va la corriente queden impregnadas, luego de esto con la ayuda del ácido férrico producir una reacción química terminando el proceso de las pistas.



COMPONENTES DE LA PLACA PRINCIPAL.



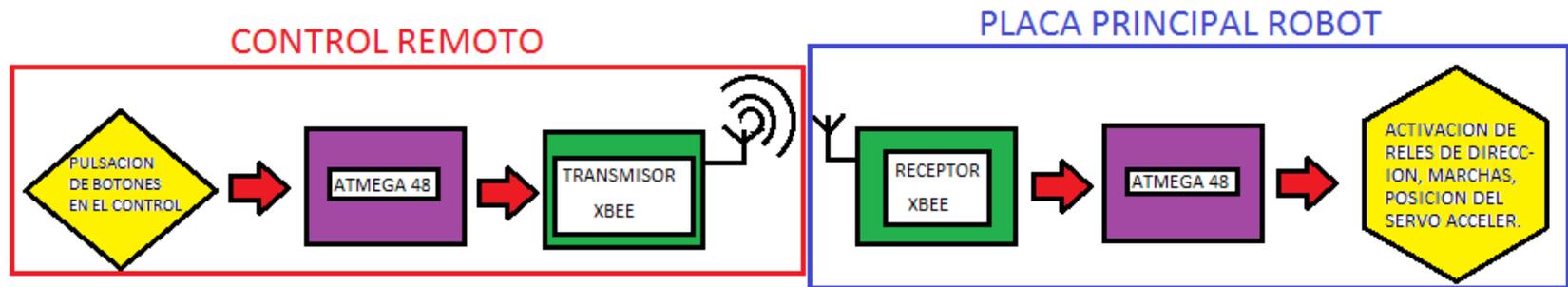
- 1) Mini relé encargado del encendido del motor.
- 2) Mini relé encargado del apagado del motor.
- 3) Bornera de conexión del relé del apagado.
- 4) Dos diodos en paralelo con las dos bobinas de los relés 1 y 2.
- 5) y 6) Transistores de Potencia
- 7) Micro controlador ATMEGA48V -10PU.
- 8) Puente en H; L298
- 9) Cristal Externo que da estabilidad al circuito.
- 10) Xbee Receptor.
- 11) Resistencia de 4,7 k.Ohmios.
- 12) Regulador de Voltaje LM117.
- 13) y 14) Regulador de voltaje Positivo 7805
- 15) Condensador de 1000 μf
- 16) Regulador de Voltaje 7812
- 17) Pulsador de encendido/apagado de la tarjeta principal.
- 18) Socket del servo de la aceleración.
- 19) Socket de programación para el ATMEGA48V -10PU.
- 20) Led indicador de estado ON/OFF.
- 21) Bornera de conexión del sensor electromagnético.
- 22) Bornera de conexión de los cambios.
- 23) Bornera de conexión de la dirección.
- 24) Bornera de conexión del relé del encendido.

DISEÑO DEL EQUIPO (HARDWARE DE CONTROL).

- **CONTROL REMOTO**

- Para el control remoto se utiliza una palanca de Playstation debido a que es la más idónea para los propósitos del proyecto, esta posee los suficientes botones o pulsadores para los diferentes comandos a enviar al robot, además de una palanca analógica propicia para la aceleración; a continuación se muestra el diagrama básico del procesamiento de los datos, desde que el operario del robot pulsa cualquier botón del control, hasta la orden o accionamiento de los diferentes elementos finales.

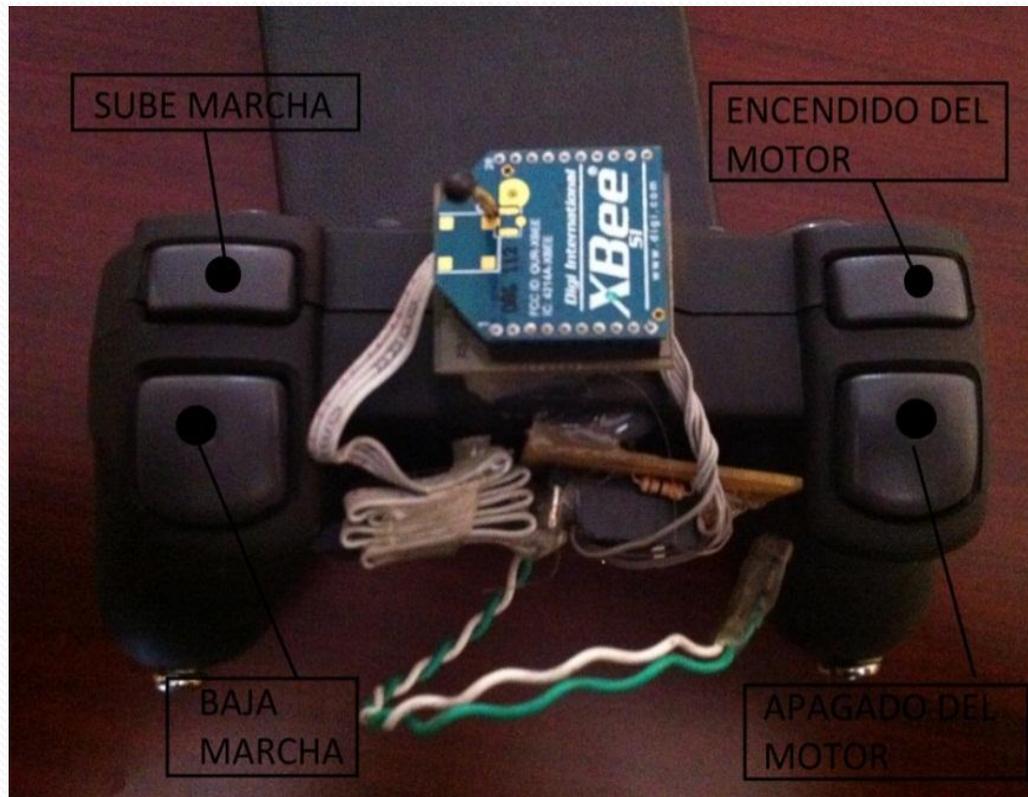
Diagrama del procesamiento de datos.

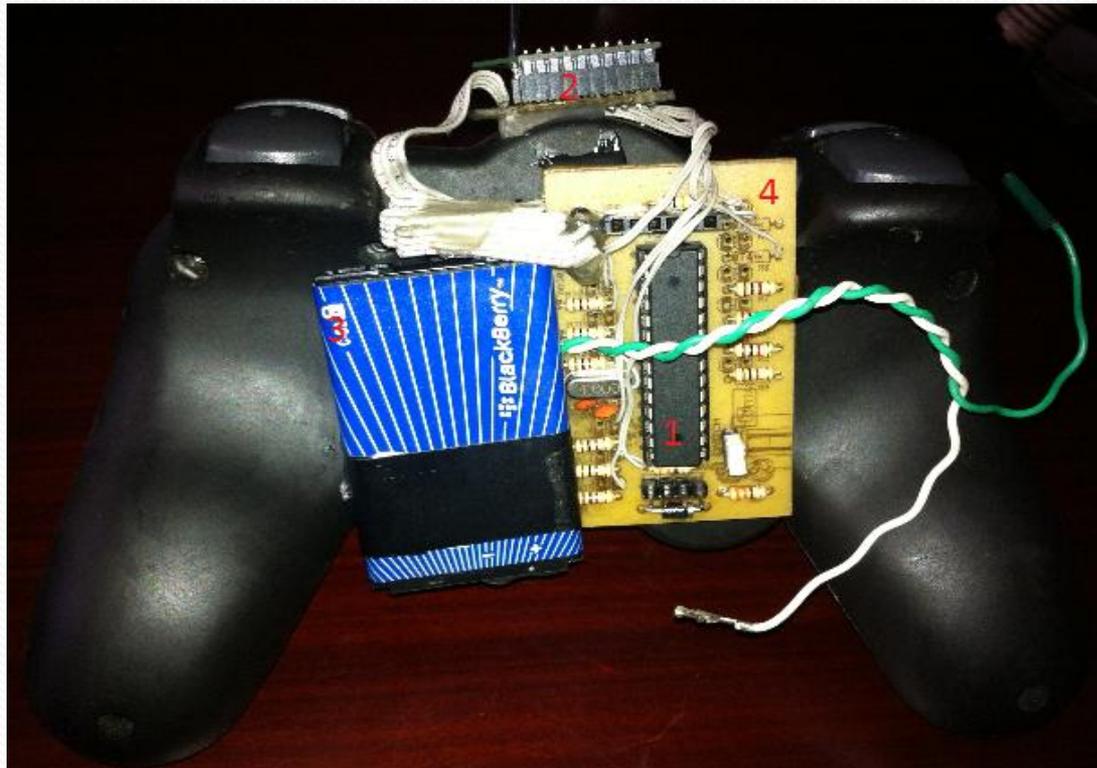


- El control posee seis botones activos, y una palanca analógica para la aceleración; la cual posee un potenciómetro internamente para determinar las necesidades de aceleración.



- Los seis botones o pulsadores corresponden a: Encendido y apagado del motor; ascenso y descenso de marchas; giro de dirección a la izquierda y giro de dirección a la derecha.





- 1) ATmega 48
- 2) Xbee transmisor
- 3) Batería 3.3V
- 4) Placa del control remoto.



DISEÑO Y CONSTRUCCION DE LA CARROCERIA DEL ROBOT

- 
- El objetivo primordial de la carrocería es proteger a los diferentes elementos, tanto eléctricos como electrónicos del robot, ante posibles ataques externos además de proveer protección contra el viento, polvo, lluvia, etc.
 - Las características de dicha estructura se detallan a continuación:

- 
- Liviana
 - Resistente
 - Fácil de Remover
 - Debe estar diseñada y construida en una sola pieza
 - Fácil de fabricar
 - De bajo costo
 - Simplicidad de construcción

- 
- Es por esto que se eligió a la fibra de vidrio como material para su construcción ya que posee todas las características anteriormente mencionadas.

LA FIBRA DE VIDRIO

- Material de muy fácil acceso comercial y sobretodo usado muy comúnmente en nuestro medio, con las ventajas antes mencionadas, la fibra de vidrio, se endurece con la aplicación de una resina que provoca una adhesión fuerte y rápida, es desde hace mucho tiempo usada para varias aplicaciones, que van desde reemplazar al acero como carrocería, hasta la creación de molde de casi cualquier forma. Esta fibra se produce en fábricas grandes a partir de un proceso no muy complicado de producción pero siendo de gran aceptación entre artesanos, productores de carrocerías y afines por sus bondades. Aunque su resistencia es menor a la de otras fibras, su ventaja costo-beneficio es realmente muy buena; es por eso que se optó en utilizarla en este proyecto en particular.

DISEÑO DEL MOLDE.

- El primer paso fue colocar placas de acero inoxidable de 0.4mm alrededor del armazón lo más cerca a la forma del armazón de tubo, con la finalidad de tener en donde asentar la fibra de vidrio, y poder modelarla como se puede observar en la fotografía a continuación.



PROCESO DE FABRICACION

- Lo mejor de trabajar con esta fibra, es su simplicidad, no se requiere mas que unas cuantas tiras de este material, un galon de resina aproximadamente, y una brocha para así ir moldeando la carrocería a la forma del chasis tubular.



- Una vez que se ha esperado el tiempo necesario se procede a masillar la fibra de vidrio, primero con una masilla especial para la misma y luego aplicando una segunda capa esta vez de masilla automotriz.



- 
- Seguido a esto se procede a darle fondo y para finalizar se pinta del color escogido en el caso del proyecto con los colores de la carrera de ingeniería automotriz negro, rojo, y blanco.
 - Una vez acabado se deja secar por varios días y se aplica pulimento automotriz.









CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- El movimiento del robot es fluido y con la práctica adecuada del operario se hace muy fácil su control.
- El robot posee la autonomía y potencia esperadas al inicio del proyecto siendo un robot competitivo y versátil.
- El robot de combate cumple con las normas establecidas en su categoría.
- Los materiales elegidos cumplieron satisfactoriamente con las pruebas realizadas.
- El diseño del software de los componentes electrónicos AVR's están al alcance de los interesados para otros proyectos similares a futuro o incluso para que análisis y posible mejoramiento del mismo.
- Las pruebas de campo experimentales revelaron el eficiente trabajo de todos los componentes eléctricos y electrónicos que se esperaba.

RECOMENDACIONES

- Antes de llevar al robot a competencias, realizarle el debido mantenimiento como es la revisión de aceite, cambio de filtros de aceite y gasolina, chequeo de presión en los neumáticos, estado de los niveles de electrolito en la batería, estado de la bujía, cantidad de combustible, etc.
- Se podría, si se quiere; elegir otro tipo de material para la estructura como una aleación de aluminio para reducir el peso significativamente.
- Optar por módulos XbeePro[®] que poseen mayor alcance de comunicación inalámbrica.
- Que el operario haya practicado lo suficiente para acudir a cualquier torneo en representación de la ESPE.



FIN DE LA PRESENTACION

GRACIAS