



**UNIVERSIDAD ESTATAL
PENÍNSULA DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES**

**CARRERA DE ELECTRÓNICA Y
TELECOMUNICACIONES**

TRABAJO DE TITULACIÓN

Propuesta Tecnológica, previo a la obtención del Título de:

**INGENIERO EN ELECTRÓNICA Y
TELECOMUNICACIONES**

Prototipo de sistema de locomoción de un robot móvil 6x6
con módulo de suspensión rocker-bogie para aplicación
agrícola.

AUTOR

BACILIO CATUTO HÉCTOR MANUEL

PROFESOR TUTOR

ING. SENDEY VERA GONZÁLEZ

LA LIBERTAD – ECUADOR

2016

AGRADECIMIENTO

Agradezco a mi Dios, quien es mi guía, mi fortaleza frente a las adversidades que se presentan, bendiciendo cada paso en mi carrera profesional. A mis padres Héctor Hernández Bacilio Santos y Betty Maritza Catuto Pozo, por su amor y apoyo incondicional, quienes me inculcaron valores y principios, forjando mi carácter y perseverancia para poder alcanzar mis objetivos. A mis hermanos por estar siempre conmigo, a mi abuela Julia Santos, que aunque ya no estés conmigo fuiste un pilar fundamental para no desmayar. A mis amigos quienes me brindaron las manos en momentos difíciles. A mi tutor Ing. Sendey Vera por su ayuda incondicional en esta etapa de mi vida. A mis profesores por impartir sus conocimientos con paciencia. Gracias a ellos me he convertido en la persona que soy ahora.

Héctor Manuel Bacilio Catuto

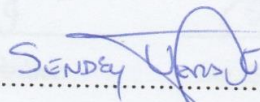
APROBACIÓN DEL TUTOR

TRIBUNAL DE GRADO

En mi calidad de Tutor del trabajo de titulación denominado: "Prototipo de sistema de locomoción de un robot móvil 6x6 con módulo de suspensión rocker-bogie para aplicación agrícola", elaborado por el estudiante Bacilio Catuto Héctor Manuel, de la carrera de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes y autorizo al estudiante para que inicie los trámites legales correspondientes.

DIRECTOR DE CARRERA

La Libertad, Septiembre del 2016.



Ing. Sendey Vera González

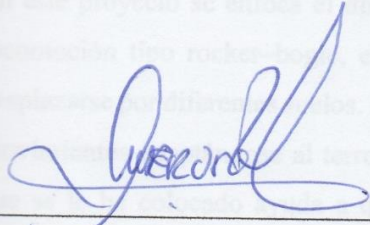
TUTOR



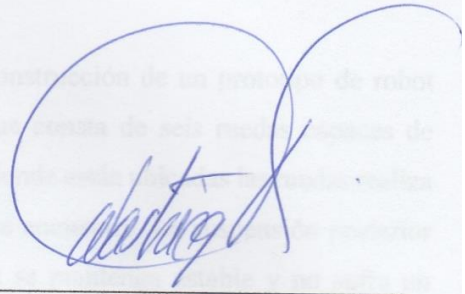
SECRETARIA GENERAL

TRIBUNAL DE GRADO

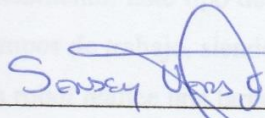
RESUMEN



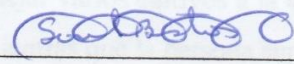
Ing. Walter Orozco Iguasnia, MSc.
DECANO DE FACULTAD



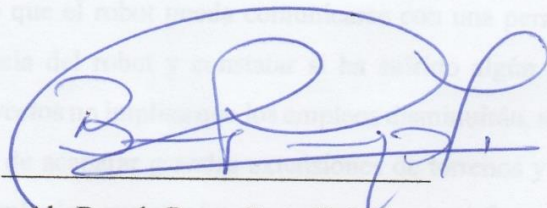
Ing. Washington Torres Guin, MSc.
DIRECTOR DE CARRERA



Ing. Sendey Vera González
PROFESOR TUTOR



Ing. Samuel Bustos Gaibor, MSc.
PROFESOR DE ÁREA



Ab. Brenda Reyes Tomalá, MSc.
SECRETARIA GENERAL

RESUMEN

En este proyecto se enfoca el diseño y la construcción de un prototipo de robot locomoción tipo rocker–bogie, el mismo que consta de seis ruedas capaces de desplazarse por diferentes suelos. El soporte donde están ubicadas las ruedas realiza movimientos adaptándose al terreno donde se encuentre. La suspensión posterior que se le ha colocado ayuda a que el robot se mantenga estable y no sufra un volcamiento. El sensor que se tiene en la parte frontal permite rastrear algún obstáculo que impida su continuidad o sufra un daño por algún choque. Su funcionalidad principal consiste en llevar a cabo su tarea sin contratiempo en su desplazamiento en los campos de cultivos de nuestro entorno. Como se sabe los campos no se caracterizan precisamente por ser terrenos llanos, por lo tanto se está construyendo robots con mayor razonamiento para afrontar situaciones adversas en desplazamiento. Este tipo de proyectos está ayudando a que las personas amplíen sus campos de trabajo, siendo capaces de manejar varios cultivos a la vez dejando que un robot realice una tarea en específica mientras que la persona supervisa otro campo. Uno de los factores negativos que se puede ver en los campos agrícolas es la utilización de químicos, por ende el uso de este robot es muy importante ya que las personas evitan el contacto directo con estos componentes que pueden ser perjudiciales para la salud. Otro aspecto importante es la comunicación inalámbrica, haciendo que el robot pueda comunicarse con una persona que está ubicada a gran distancia del robot y constatar si ha sufrido algún percance. La incursión de estos proyectos no implica que los empleos disminuirán, sino que serán de apoyo al momento de acaparar grandes extensiones de terrenos y aumentar su productividad. Los resultados que se obtuvieron son muy satisfactorios sabiendo que podrán ser mejorados en un futuro y con esto se está dando paso a la intervención de la tecnología robótica en el campo agrícola.

ABSTRACT

In this project, we talk about the design and construction of a robot's prototype rocker-bogie, the same that has six wheels, it can move through different soils. The support where are located the wheels makes movements adapting to the land where you are. The back suspension helps the robot remains stable and does not suffer a rollover. The sensor that is on the front allows you avoid obstacles to its continuity and don't suffer injury for a collision. Its main function is to carry out its task without mishap in their displacement in the fields in our environment. As is known, the fields are not characterized by being flat surfaces, therefore it is being built robots more reason to face adverse situations in displacement. This type of project is helping people expand their fields of work, being able to handle various crops while letting a robot perform a specific task while the person supervises another field. One of the negative factors that can be seen in the fields is the use of chemicals, therefore the use of this robot is very important because people avoid direct contact with these components that can be harmful to health. Another important aspect is the wireless communication, making the robot can communicate with a person who is located at a great distance from the robot and determine whether they have suffered any mishap. The incursion of these projects does not mean that jobs will decrease, but will support when large tracts of land hoarding and increase productivity. The results obtained are very satisfactory knowing that can be improved in the future and this is leading to the intervention of robotic technology in the agricultural field.

APROBACION DEL TUTOR	II
TRIBUNAL DE GRADO	III
RESUMEN	IV
ABSTRACT	V
DECLARACION	VI

DECLARACIÓN

El contenido del presente Trabajo de Graduación es de mi responsabilidad: el patrimonio intelectual del mismo pertenece a la Universidad Estatal Península de Santa Elena.

TABLAS	XII
INDICE DE ECUACIONES	XIII
LISTA DE ANEXOS	XIV
INTRODUCCION	1
CAPITULO I	2
1.1 ANTECEDENTES	2
1.2 DESCRIPCION DEL PROYECTO	3
1.3 OBJETIVOS	4
1.3.1 Objetivo general	4
1.3.2 Objetivos especificos	4
1.4 JUSTIFICACION	5
1.5 ALCANCE DEL PROYECTO	5
1.6 METODOLOGIA	6
CAPITULO II	7
2.1 MARCO CONTEXTUAL	7

.....
Héctor Manuel Bacilio Catuto.

TABLA DE CONTENIDOS

ITEM	PÁGINA
AGRADECIMIENTO	I
APROBACIÓN DEL TUTOR	¡Error! Marcador no definido.
TRIBUNAL DE GRADO	¡Error! Marcador no definido.
RESUMEN	IV
ABSTRACT	V
DECLARACIÓN	¡Error! Marcador no definido.
TABLA DE CONTENIDOS	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABLAS	XII
ÍNDICE DE ECUACIONES	XIII
LISTA DE ANEXOS	XIV
INTRODUCCIÓN	1
CAPÍTULO I	2
1.1 ANTECEDENTES	2
1.2 DESCRIPCIÓN DEL PROYECTO	3
1.3 OBJETIVOS	4
1.3.1 Objetivo general	4
1.3.2 Objetivos específicos	4
1.4 JUSTIFICACIÓN	5
1.5 ALCANCE DEL PROYECTO	5
1.6 METODOLOGÍA	6
CAPÍTULO II	7
2.1 MARCO CONTEXTUAL	7
2.2 MARCO CONCEPTUAL	9
2.3 MARCO TEÓRICO	14
2.4 COMPONENTES DE LA PROPUESTA.	16
2.5 DISEÑO DE LA PROPUESTA.	20
DISEÑO MECÁNICO DEL ROBOT DE LOCOMOCIÓN.	21
DISEÑO ELECTRÓNICO DEL ROBOT DE LOCOMOCIÓN	30
DISEÑO LÓGICO DEL ROBOT DE LOCOMOCIÓN.	41
2.6 ESTUDIO DE FACTIBILIDAD.	54

2.7 RESULTADOS.	56
CONCLUSIONES	60
RECOMENDACIONES	62
BIBLIOGRAFÍA	63
ANEXOS	65

ÍNDICE DE FIGURAS

ITEM	DESCRIPCIÓN	PÁGINA
Figura 1	Campos de cultivos: 2000agro.	7
Figura 2	Diferentes terrenos para cada tipo de robot.	7
Figura 3	Mars Sojourner Rover: NASA.	8
Figura 4	Robot detector de calidad de las uvas: agriculturers.com.	8
Figura 5	Robot rocker bogie rover para desplazamiento: picaxe.com	8
Figura 6	Radio de cobertura para comunicación.	10
Figura 7	Diseño rocker-bogie: wikimedia.org.	10
Figura 8	Movimiento del diseño rocker-bogie: wikimedia.org.	11
Figura 9	Método Triangulo semejante: Tales de Mileto.	11
Figura 10	Ejemplo de triangulo Semejante: Tales de Mileto.	11
Figura 11	Semejanza de triángulos: Tales de Mileto.	12
Figura 12	Semejanza de triángulos primos: Tales de Mileto.	12
Figura 13	Funciones Trigonométricas: Tales de Mileto.	12
Figura 14	Torque de una rueda: orientalmotor.com.	13
Figura 15	Vine robot: agriculturers.com	15
Figura 16	IDE Arduino	18
Figura 17	Inicio programa Visual studio 2015	18
Figura 18	Área de codificación.	18
Figura 19	Área de diseño	18
Figura 20	Programa de visualización cámara	19
Figura 21	Aplicativo para teléfono.	19
Figura 22	Programa X-CTU	19
Figura 23	Visualización del prototipo de parte frontal.	20
Figura 24	Visualización del prototipo de parte lateral.	20
Figura 25	Distancia entre ruedas.	21
Figura 26	Datos dados por la propuesta.	21
Figura 27	Datos de puntos A, B, D.	22
Figura 28	Datos de B, A, F.	22
Figura 29	Datos de A, B, D.	22
Figura 30	Datos de C, E, G.	22
Figura 31	Medidas para el diseño de robot.	23
Figura 32	Medidas del rocker.	24
Figura 33	Medidas del Bogie.	24
Figura 34	Tipo de madera: Masisa	25
Figura 35	Medidas base parte inferior y superior.	25
Figura 36	Medidas base frontal y lateral.	26
Figura 37	Medidas mini chumacera.	26
Figura 38	Base soporte posterior.	26
Figura 39	Punto de equilibrio.	28
Figura 40	Torque de rueda.	29
Figura 41	Movilidad de servomotores izquierda y derecha.	29

Figura 42	Diseño soporte de rueda.	29
Figura 43	Regulador de voltaje.	30
Figura 44	Circuito regulador de voltaje.	31
Figura 45	Circuito encendido de luces con conexión a la tarjeta Arduino.	31
Figura 46	Conexión de batería y dispositivos	32
Figura 47	Circuito de conexión de batería y dispositivos en Circuit Wizard.	32
Figura 48	Conexión del motoreductor y puente h	33
Figura 49	Conexión motoreductor, l298n y T. Arduino	33
Figura 50	Activación de transistores para direccionamiento.	33
Figura 51	Electrónica interna del dispositivo: Integrado l298n	34
Figura 52	Conexión de puente h con los motores.	34
Figura 53	Tarjeta Arduino Mega como controlador	35
Figura 54	Conexión de puentes h con la tarjeta Arduino.	35
Figura 55	Conexión del módulo xbee y sensor ultrasónico.	36
Figura 56	Recorrido del pulso en el sensor	36
Figura 57	Conexión Xbee-Arduino	37
Figura 58	Conexión Pc- Xbee	37
Figura 59	Programa X-CTU para configuración	38
Figura 60	Datos Programados en los XBee	38
Figura 61	Configuración XBee.	39
Figura 62	Terminales del Motoreductor	40
Figura 63	Conexión a tarjeta Arduino	40
Figura 64	Conexión de servo motor	40
Figura 65	Posicionamiento del servomotor	40
Figura 66	Diagrama de flujo	41
Figura 67	Proceso manual automático	44
Figura 68	Llamada de funciones-manual	45
Figura 69	Llamado de funciones-automático	49
Figura 70	Diagrama del prototipo.	54
Figura 71	Recorrido en campo de cultivo.	57
Figura 72	Pruebas en terrenos	57
Figura 73	Unión de las partes de fumigación y suspensión	59
Figura 74	Motoreductor 50:1 (Metal Gearmotor)-37Dx54L: Pololu	32
Figura 75	Servo MG995 Tower Pro.	32
Figura 76	Conexión para la alimentación.	33
Figura 77	Conexión 12v en puente H.	33
Figura 78	Módulo de sensor HC-SR04.	34
Figura 79	Microprocesador ATmega2560	35
Figura 80	Módulo xbee s2.	35
Figura 81	Adaptador Universal XBee USB.	69
Figura 82	Batería seca 12v a 5 Amperios.	69
Figura 83	Data sheet Puente H L298:	70
Figura 84	Data sheet Arduino mega	71
Figura 85	Regulador de Voltaje LM78XX	72
Figura 86	Data sheet Transistor NPN 2N3904	73

Figura 87	Corte de barras de aluminio rocker y bogie.	74
Figura 88	Colocación de pernos de ajustes.	74
Figura 89	Armado del soporte con los pernos.	74
Figura 90	Colocación de servos.	74
Figura 91	Cambio de las agarraderas de motores.	75
Figura 92	Vista inicial del soporte con ruedas.	75
Figura 93	Ensamblaje del Chasis.	75
Figura 94	Partes de la mini chumacera.	75
Figura 95	Armado del soporte interno.	75
Figura 96	Base de las ruedas con tubos PVC.	76
Figura 97	Recubrimiento de las bases de las ruedas con caucho de bicicleta.	76
Figura 98	Vista de las llantas con recubrimiento.	76
Figura 99	Unión de la base con los soportes.	76
Figura 100	Construcción de soporte para circuitos.	77
Figura 101	Colocación de circuitos eléctricos.	77
Figura 102	Realización de pruebas.	77
Figura 103	Diseño de nuestro robot móvil terminado.	77
Figura 104	Panel de Control.	78
Figura 105	Carga del código en el programa Arduino.	79
Figura 106	Compilamos el código en Arduino.	79
Figura 107	Asignación de placa Arduino.	79
Figura 108	Asignación de puerto serial.	79
Figura 109	Desconexión del pin de transmisión en T. Arduino.	80
Figura 110	Compilación del programa Arduino.	80
Figura 111	Ejecución del programa visual studio.	80
Figura 112	Ítem de acceso a nuestro código.	80
Figura 113	Ventana de desarrollo de aplicación.	81
Figura 114	Conexión de puerto serial en Arduino.	81
Figura 115	Ubicación del botón inicio.	81
Figura 116	Ejecución del programa.	81
Figura 117	Acceso rápido de nuestra aplicación.	82
Figura 118	Conexión de carga de batería.	82

ÍNDICE DE TABLAS

ITEM	DESCRIPCIÓN	PÁGINA
Tabla 1	Detalle de voltaje y corriente	17
Tabla 2	Comparación de diseños	20
Tabla 3	Materiales para armar la base.	27
Tabla 4	Pesos de componentes.	28
Tabla 5	Asignación de datos a los XBee	38
Tabla 6	Resumen de pines de tarjeta Arduino	42
Tabla 7	Proceso manual1	46
Tabla 8	Proceso manual2	48
Tabla 9	Movimientos Manuales	49
Tabla 10	Activación modo automático	50
Tabla 11	Función recorrido automático	51
Tabla 12	Movimientos automáticos	52
Tabla 13	Función posición servo	53
Tabla 14	Funciones Servos	53
Tabla 15	Detalle de costo de software	54
Tabla 16	Detalle de costo de pc.	54
Tabla 17	Valor de cada componente usado en la elaboración del proyecto.	56
Tabla 18	Potencia de los motores zona libre.	56
Tabla 19	Potencia de los motores zona con pendiente Ángulo 30.	56
Tabla 20	Prueba Manual.	58
Tabla 21	Prueba Automático.	58
Tabla 22	Consumo de corriente.	59
Tabla 23	Descripción de direccionamiento.	82

ÍNDICE DE ECUACIONES

ITEM	DESCRIPCIÓN	PÁGINA
Ecuación 1	Funciones Trigonómicas	12
Ecuación 3	Distancia entre las ruedas.	21
Ecuación 4	Ecuación de medidas generales	22
Ecuación 5	Cálculo entre A, B, F.	22
Ecuación 6	Cálculo entre C, E, G	23
Ecuación 7	Fórmula de Peso.	27
Ecuación 8	Conversión a kg	28
Ecuación 9	Peso de la base del prototipo	28
Ecuación 10	Cálculo de carga para cada llanta.	28
Ecuación 11	Fórmula de Torque	29
Ecuación 12	Cálculo de torque de nuestra rueda	29
Ecuación 13	Velocidad del sonido	36
Ecuación 14	Fórmula de la distancia para el sensor	36

LISTA DE ANEXOS

N.-	DESCRIPCIÓN
1	Descripción de componentes físicos.
2	Data sheet Puente H L298
3	Configuración de Pines Arduino Mega.
4	Regulador de Voltaje LM78XX
5	Data sheet Transistor NPN 2N3904.
6	Construcción del robot de locomoción.
7	Manual de usuario.
8	Codificación en visual Basic studio 2015

INTRODUCCIÓN

En los últimos años la robótica se ha establecido como tema principal en todos los ámbitos de nuestra vida cotidiana. Tal es el caso que en cada rincón de nuestra casa o nuestro trabajo podemos toparnos con una novedad tecnológica, haciendo que el modo de vivir de una persona sea más cómoda.

La incursión de robots que ayuden en nuestro trabajo no solo ha acaparado la atención de quienes vivimos en las grandes ciudades, sino también aquellos lugares donde la tecnología parece estar décadas atrás. Los campos agrícolas han visto en la tecnología una oportunidad de superación, llevando a sus cultivos a producir a gran escala.

La elaboración de robots que puedan cumplir con tareas específicas, el movimiento autónomo de vehículos y la comunicación inalámbrica, son temas que al ser unificados, hacen que se creen opciones de interés para las personas dedicadas a los cultivos.

Una de estas opciones es la construcción de robots móviles capaces de desplazarse por varios campos, pudiendo ser controlados manualmente desde lugares apartados, valiéndose de las comunicaciones inalámbricas que existen en la actualidad, o bien dejándoles que realicen una tarea de forma autónoma.

Debido a las condiciones de los campos agrícolas, se opta por un diseño basado en la suspensión del rocker-bogie, la cual se adapta muy bien al terreno, debido a sus 6 ruedas y la suspensión posterior.

El diseño de la suspensión hace que la base se mantenga estable, mientras un lado de ruedas se alza, el otro se mantiene sujeta al suelo. También se caracteriza por tener un sensor que hace que se detenga al detectar un obstáculo que supera sus limitaciones de movimiento.

El proyecto se basa en explicar su construcción, los límites de sus movimientos, y qué alcances puede tener, e incluso dejar datos que sirvan para poder mejorarla. Como resultado se espera que el robot de locomoción pueda desplazarse sin dificultad en un campo agrícola analizando su entorno y la orden que se le dé.

CAPÍTULO I

FUNDAMENTACIÓN

1.1 ANTECEDENTES

Como principio fundamental del funcionamiento de los robots en el campo de la agricultura, se basa en el reconocimiento del terreno en los cultivos, mediante imágenes o videos que son procesadas en la computadora para luego proceder a dar órdenes y realizar alguna tarea en específica, como ejemplo podemos mencionar: la aplicación de fertilizante, plaguicidas, recolección de muestras y riego. Todo esto con el objetivo de conseguir una agricultura de precisión.

Aunque en la actualidad existen robots muy sencillos para tareas específicas como: sembrar, cosechar y hacer labores de preparación del suelo, también podemos encontrar robots más avanzados que pueden realizar tareas de mayor complejidad, como transmisión de información sobre el terreno, tarea de búsqueda y registro sobre el estado de los frutos.

La información del terreno es un punto muy importante a considerar, esto se debe a que si el robot no está capacitado para movilizarse adecuadamente, puede ocasionar daños en los sembríos.

Hay casos en que un robot no cumple su recorrido por el hecho de quedarse estancando con piedras, no saber qué hacer si se encuentra con ramas caídas que puedan interferir su movilidad, afectando su lógica de reacción, o simplemente volcarse por mala maniobra en su mecanismo al momento de desplazarse.

El robot debe estar preparado para asumir retos de obstáculos que puedan surgir en los cultivo, siendo capaz de buscar una solución de forma autónoma.

Hoy en día existen muchos tipos de mecanismos móviles de variadas formas, funcionamiento y robustez, que han venido evolucionando a la par de la tecnología y que están logrando que un robot móvil trabaje sin problema en terrenos difíciles, pudiendo cumplir a cabalidad su objetivo de desplazamiento, recolección de datos y destrezas.

1.2 DESCRIPCIÓN DEL PROYECTO

El robot será ensamblado con el fin de que pueda ser capaz de desplazarse por cualquier campo en la agricultura, sean estas con cierto grado de desnivel o en terrenos con piedras. Cada cultivo es diferente y por ende su dificultad de desplazamiento aumenta.

Su diseño se basará en el sistema de suspensión ROCKER-BOGIE con el fin de mejorar su movilidad, a la par llevará un soporte posterior que ayudará en su estabilidad. Consistirá en 6 ruedas acopladas al tipo de terreno que generalmente existe en los cultivos de cualquier entorno.

Cada rueda llevará en su interior un motor con la potencia necesaria para su movilidad en el terreno. Los motores a utilizar, serán de acuerdo al peso del robot, y su carga adicional sobre la plataforma (parte del robot que se encarga de esparcir líquido en los cultivos).

Utilizará sensores ultrasónicos que permitirán detectar cualquier tipo de obstáculo en su camino, haciendo que pueda evadirlos con maniobras pre definidas en el software al momento de movilizarse de forma automática, o simplemente detener la funcionalidad de los motores.

La cámara instalada en la parte superior, nos servirá como reconocimiento visual del área del cultivo, mostrándonos de forma remota hasta un computador lo que se visualiza en su camino.

Tendrán servomotores que facilitarán sus movimientos hacia la izquierda o derecha, independientemente si se dirigen hacia adelante o hacia atrás. Utilizará componentes como luces o buzzer que servirán como alerta para los usuarios al momento de detenerse por obstáculos.

En la parte del software se utilizará el monitor serial de Arduino junto a una interfaz gráfica, la misma que será de fácil manejo y entendimiento con cualquier persona que quiera operar este tipo de robots.

Podrá ser operado de manera inalámbrica mediante controles ubicados en el teclado de nuestro pc.

1.3 OBJETIVOS

1.3.1 Objetivo general

Implementar un prototipo de un robot con sistema de locomoción 6x6 con diseño de suspensión ROCKER- BOGIE para movilizarse en terrenos agrícolas, mediante tecnología Arduino y uso de cámara y sensores.

1.3.2 Objetivos específicos

- Estudio del modelo ROCKER-BOGIE para la elaboración del soporte de nuestro prototipo móvil.
- Ensamblar un mecanismo de soporte basado en el sistema de suspensión Rocker-Bogie, con el fin de que pueda evadir obstáculos presentados en los terrenos de cultivos.
- Análisis del sistema de suspensión Rocker-Bogie para que el prototipo de locomoción pueda mantenerse estable durante su recorrido en los terrenos de sembríos.
- Diseñar el sistema electrónico, la cual nos va a permitir controlar la trayectoria, tanto en el modo manual como en el modo automático por medio del hardware y del software.
- Desarrollo de la codificación en la plataforma de Arduino para controlar el robot y pueda hacer los movimientos necesarios al momento de evadir obstáculos.
- Construcción y evaluación de resultados obtenidos de los datos experimentados, mediante el prototipo de robot con sistema de locomoción 6x6 con diseño de suspensión Rocker-Bogie para movilizarse en terrenos agrícolas usando tecnología Arduino, cámaras y sensores.

1.4 JUSTIFICACIÓN

Mediante ésta propuesta se busca fortalecer la utilización de éstos prototipos de robots automatizados de forma confiable y accesible para las personas que se dedican especialmente a la agricultura.

Al referirnos a terrenos de sembríos, si el mecanismo de movilización falla, por ende la función a la que está dirigido el robot, ejemplo fumigación, no podrá ser ejecutada con éxito y los sembríos se verían afectados al no recibir la atención respectiva.

La construcción de este prototipo de robot se basa generalmente porque en nuestro entorno el nivel del suelo es muy variado, a veces con mucho desnivel, por lo cual se busca que el robot pueda mantenerse estable.

El tipo de suelo también implica un problema a la hora de desplazarse porque pueden existir piedras que impidan el paso libre del robot.

Relevancia social: Se beneficiarán todos quienes conforman la comunidad agrícola.

Implicaciones prácticas: Ayuda a desplazarse en terrenos difíciles.

Valor teórico: Con la investigación se podrá profundizar los temas concernientes al software que utilizaremos y el mecanismo de suspensión.

Utilidad metodológica: La investigación nos ayudará a ponernos al alcance con temas de actualidad como es la robótica.

1.5 ALCANCE DEL PROYECTO

El prototipo del robot de locomoción tiene como alcance obtener mayor información técnica de movilidad y visual que pueda mejorar la productividad en la comunidad agrícola.

- El prototipo será capaz de evadir obstáculos como piedras.
- Su sistemas de suspensión será capaz de mantenerlo estable ante los desniveles que se puedan generar por los obstáculos.
- Este prototipo podrá enviar información visual continua como forma de exploración del entorno mediante una cámara.

- Tendrá la capacidad de detectar algún objeto que impida su paso, mediante sensores.
- Será capaz de cumplir un recorrido de forma autónoma, dándole con anterioridad los datos del área del cultivo.
- Su recorrido normal dependerá de los datos recogidos por los sensores.

1.6 METODOLOGÍA

El estudio se emplea para dar a conocer la problemática de ciertos terrenos y poder movilizarse sobre ellos. Para su construcción se basó en el sistema de suspensión ROCKER-BOGIE, la cual nos ayudará a mantener estable nuestro prototipo ante la diversidad del terreno y sus dificultades.

Estudio de visita de campo

Terrenos a desnivel y de difícil movilización que se presentan en los diferentes cultivos en cualquier parte del país.

Estudio descriptivo

Podrá movilizarse sin ningún problema de estabilidad, aún si se encuentra con piedras en su camino. El análisis de esta teoría será importante para su construcción.

Métodos e instrumentos de investigación

Método de modelación: Este método opera de forma práctica o teórica con un objeto de forma indirecta, utilizando un sistema intermedio llamado modelo, mediante el cual se crean abstracciones para investigar la realidad, donde se revela la unidad de lo objetivo y lo subjetivo.

Método empírico: Este método se basa en aprender o hacer algo a base de experiencias propias o de otros autores referentes a la problemática, que pueden ser tanto positivos como negativos. Entre los métodos empíricos tenemos la técnica de observación, donde tomamos información y la registrarnos para posteriormente analizarla, la medición, donde determinamos la magnitud del objeto a investigar, y el experimento donde se estudia el comportamiento de acuerdo a nuestra hipótesis.

Método sistémico: Este método permite potenciar la integración de varios conceptos que proceden de varios campos, formando un todo y que tiene un fin.

CAPÍTULO II LA PROPUESTA

2.1 MARCO CONTEXTUAL

Ubicación del problema:



Figura 1: Campos de cultivos: 2000agro.

El problema se suscita en los cultivos donde existe diferentes tipos de terrenos, en la cual un móvil normal no puede desempeñarse con normalidad.

Las siguientes imágenes recrean un terreno ideal para cada tipo de robot.

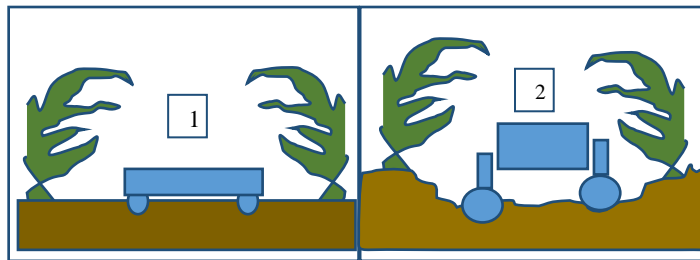


Figura 2: Diferentes terrenos para cada tipo de robot.

En la primera figura podemos ver que el robot puede movilizarse libremente en un terreno llano, mientras que en la segunda podemos ver que el robot puede adaptarse al terreno desnivelado debido a su diseño. Si hacemos el cambio de terrenos tendremos problemas con el primer robot porque sus ruedas no están adaptadas a ese tipo de desnivel, mientras que el segundo no tendría ningún problema en movilizarse en un terreno llano.

No obstante este proyecto no es el primer robot que se realiza, existen muchos proyectos que se anticipan a este prototipo, con el fin de darles el uso adecuado en muchas áreas no solo en la agricultura.

Anteriormente han optado por construir robots con distintos diseños de suspensión entre las cuales podemos mencionar:

Mars Sojourner Rover.



Figura 3: Mars Sojourner Rover: NASA.

Este tipo de robot es utilizado en exploraciones espaciales, capaces de movilizarse en terrenos con alto grado de complejidad. Su función es recolectar información del tipo de suelo en la que se desplaza. Tipo de suspensión rocker bogie

Robot detector de calidad de las uvas



Figura 4: Robot detector de calidad de las uvas: agriculturers.com.

Este robot es utilizado para recolectar y transmitir información al viñador sobre el estado de las uvas mediante cartografías inteligentes. El método utilizado es de un móvil normal de cuatro ruedas para plantaciones de gran tamaño.

Robot rocker bogie rover para desplazamiento



Figura 5: Robot rocker bogie rover para desplazamiento: picaxe.com

Este tipo de robot es utilizado para desplazarse en terrenos difíciles, con la desventaja de no soportar grandes pesos encima. El método que se utiliza es el tipo rocker-bogie.

2.2 MARCO CONCEPTUAL

La utilización de estos robots implica temas muy relevantes, ya sea por su función en el terreno, mayor cuidado de la salud, así como ayudar en la agilidad de producción.

Al referirnos sobre la función que desempeña, nos brinda facilidad de recorrido debido a sus soportes de equilibrio que lo mantienen en ruta sin que se detenga por problemas de obstáculos.

Estos robots evitarán que las personas entren en contacto de forma física con los químicos que se utilizan para eliminar plagas, las mismas que pueden ser perjudiciales si su uso es constante en las personas. A cambio las personas pueden operar estos vehículos de forma remota debido a su sistema de comunicación y automatización.

La revolución que tienen estos robots en la agricultura hace que la producción aumente, no solo por la mayor extensión de tierra que puede abarcar, si no por su agilidad de movimientos haciendo que los trabajos sean más rápidos.

Funcionamiento general

Este vehículo deberá cumplir un solo propósito, que conlleva a realizar un recorrido de manera eficaz y eficiente con su trabajo.

El prototipo será diseñado para realizar el recorrido de dos maneras. La primera se basa haciendo los movimientos de forma manual, la misma que se activará en el panel posterior del vehículo, o sino utilizando los botones de direccionamiento que se diseñarán en el panel de control de nuestro software.

Sus movimientos específicamente serán adelante, atrás, izquierda, derecha y paro. Adicionalmente el prototipo tendrá movimientos de retroceso que nos ayudarán en cualquier imprevisto, y queremos desistir en su avance, estas acciones serán izquierda_atrás y derecha_atrás.

La segunda forma para realizar el recorrido será el automático, su funcionamiento dependerá de la activación del Switch que se ubicará en la parte posterior del vehículo, una vez accionado realizará el recorrido que se programe en el IDE Arduino en conjunto con la interfaz de visual. También se podrá accionar el modo automático mediante un botón ubicado en el panel de control de nuestro software.

Si por algún motivo se produce un fallo en la conexión con los motores al momento de realizar movimientos, el programa contará con un botón de abortar, que aísla la corriente que se conecta entre la batería y los motores. Si queremos que la corriente vuelva a circular se adicionará un botón de reconexión. Estos botones funcionan con ambos modos (automático y manual).

Al estar en modo automático no podremos re-direccionar el vehículo si existe algún obstáculo en el campo de sembríos. Por esta razón se le incorporará un sensor de proximidad, haciendo que nuestro prototipo se detenga si hay algo que obstruya su camino a una distancia menor a 60 cm. Si esto ocurriese, tendrá una bocina que alertará a las personas sobre el obstáculo.

Su funcionamiento se situará en un radio de 120 metros que es la distancia máxima que puede transmitir nuestra antena desde nuestra ubicación.

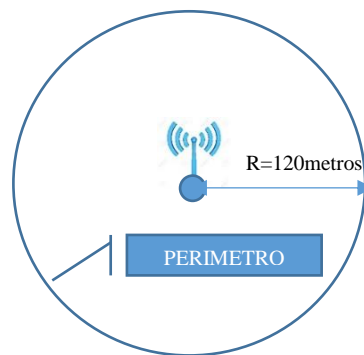


Figura 6: Radio de cobertura para comunicación.

Sistema de suspensión rocker bogie.

En nuestro proyecto es muy importante que la movilidad se adapte al tipo de terreno. Por eso se escogió la suspensión Rocker-Bogie la cual tiene las siguientes características físicas.

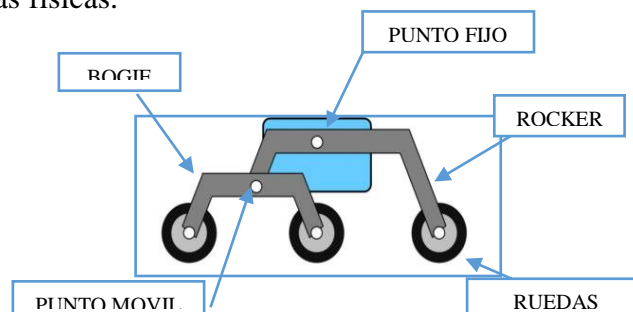


Figura 7: Diseño rocker-bogie: [wikimedia.org](https://commons.wikimedia.org/wiki/File:Rocker_bogie.svg).

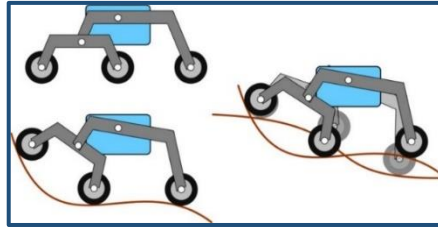


Figura 8: Movimiento del diseño rocker-bogie: wikimedia.org.

En la actualidad se han diseñados varios robots de estos tipos especialmente en ámbito espacial como por ejemplo el rover curiosity. “El Rover Curiosity es un vehículo poderoso y fuerte, con un peso de casi mil kilogramos que posee seis ruedas las que, en extensión, abarcan una superficie mayor a la de la plataforma del vehículo, a diferencia de sus antecesores, los vehículos Spirit y Opportunity.” [18] Estos proyectos nos motivan para su utilización no solo en el espacio como explorador, sino también en la agricultura para los cultivos.

De la mano a este diseño (rocker-bogie) se lo relaciona el método de Tales de Mileto y las funciones trigonométricas de ángulos, junto a estos conceptos se podrán analizar las ecuaciones para el posterior ensamblaje de nuestro robot.

Tales de Mileto, (Método de triángulos semejantes).

“Triángulos semejantes: Los segmentos determinados por rectas paralelas sobre dos rectas concurrentes son proporcionales.”[3]



Figura 9: Método Triangulo semejante: Tales de Mileto.

“Por tanto toda paralela a un lado de un triángulo determina con los otros dos lados un nuevo triangulo semejante al primero:”. [3]

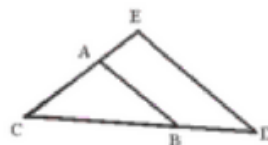


Figura 10: Ejemplo de triangulo Semejante: Tales de Mileto.

Funciones Trigonómicas.

“Introduciremos las funciones trigonométricas en la forma en que se originaron históricamente: como razones entre los lados de un triángulo rectángulo. Un triángulo es un **triángulo rectángulo** si uno de sus ángulos es recto. Si θ es un ángulo agudo, podemos considerar un triángulo rectángulo que tiene a θ como uno de sus ángulos, como en la figura 1, donde el símbolo especifica el ángulo de 90° . Se pueden obtener seis razones usando las longitudes a , b y c de los lados del triángulo:” [1]

$$\frac{b}{c}, \frac{a}{c}, \frac{b}{a}, \frac{a}{b}, \frac{c}{a}, \frac{c}{b}$$

Figura 11: Semejanza de triángulos: Tales de Mileto.

“Podemos demostrar que estas razones dependen sólo de θ y no del tamaño del triángulo, como se indica en la figura 2. Como los dos triángulos tienen ángulos iguales, son semejantes y por tanto las razones entre lados correspondientes son proporcionales. Por ejemplo”. [1]

$$\frac{b}{c} = \frac{b'}{c'}, \frac{a}{c} = \frac{a'}{c'}, \frac{b}{a} = \frac{b'}{a'}$$

Figura 12: Semejanza de triángulos primos: Tales de Mileto.

Definición de funciones trigonométricas.

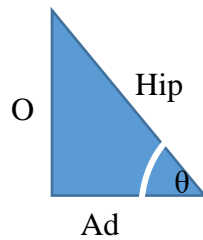


Figura 13: Funciones Trigonómicas: Tales de Mileto.

$$\begin{aligned} \operatorname{sen}\theta &= \frac{Op}{Hip} & \operatorname{csc}\theta &= \frac{Hip}{Op} \\ \operatorname{cos}\theta &= \frac{Ady}{Hip} & \operatorname{sec}\theta &= \frac{Hip}{Ady} \\ \operatorname{tan}\theta &= \frac{Op}{Ady} & \operatorname{cot}\theta &= \frac{Ady}{Op} \end{aligned}$$

Ecuación 1: Funciones Trigonómicas

Segunda ley de newton.

Esta ley será necesario para realizar el equilibrio de nuestra base a través de datos de fuerza. Utilizaremos la siguiente formula: $F = (m)(a)$.

“En la cual \mathbf{F} es la Fuerza que está actuando sobre el cuerpo; \mathbf{m} es la masa del cuerpo y \mathbf{a} representa las aceleraciones que recibe.” [4]

De esto podremos analizar el peso de un cuerpo con la siguiente formula: $P = (m)(g)$

“El peso de un cuerpo se refiere a la fuerza gravitacional con la que es atraída la masa de dicho cuerpo. Es decir, si un cuerpo tiene mayor masa, entonces tendrá un mayor peso. Asimismo, como el peso es directamente proporcional a la fuerza de gravedad con la que la masa es atraída, un hombre, por ejemplo, tendrá un mayor peso en la Tierra que en la Luna ya que la fuerza de gravedad en la Luna es menor.” [4]

Momento de una fuerza o torque.

Para saber qué tipos de motores usaremos para mover nuestro vehículo es necesario tener en cuenta los siguientes conceptos. “El momento de una fuerza o torque, es igual al producto de la fuerza, por la longitud del brazo del momento sobre el que actúa.”[2]. “La longitud del brazo de momento se mide perpendicularmente a la fuerza desde el pivote a la línea de acción de la fuerza.”[2]

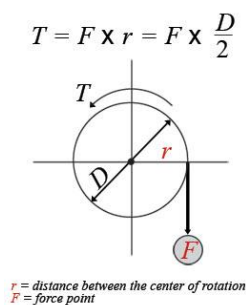


Figura 14: Torque de una rueda: orientalmotor.com.

Concepto de suspensión vehicular.

“Los vehículos automóviles existen para permitir el desplazamiento de personas o cosas de un lugar a otro. Esta afirmación tan rotunda viene a expresar implícitamente que para conseguir este fin, se debe minimizar cuanto sea posible cualquier perturbación o trastorno sobre lo que o quien está siendo transportado.” [5]

“Por otra parte, la necesidad de utilizar un sistema de suspensión en un automóvil, no está motivada únicamente por la susceptibilidad o tolerancia humana a la pérdida de confort que producen las vibraciones transmitidas al habitáculo, aunque el aislamiento de las irregularidades transmitidas desde la calzada constituya una función altamente deseable. Un objetivo, que justifica el uso de un sistema de suspensión en el vehículo, es la necesidad de mantener el contacto entre la rueda y la carretera, ya que el control y la estabilidad del mismo depende del ello.” [5]

2.3 MARCO TEÓRICO

Históricamente estos tipos de robots móviles han sobresalido en varios ámbitos prácticos, pero han incursionado notablemente en los últimos años en temas de agriculturas, la mayoría de las veces el tema principal es la recolección de los frutos. “La introducción de robótica avanzada en la agricultura permitirá automatizar trabajos para los que normalmente se requiere la destreza de personas. Actualmente se está investigando cómo recoger algunas frutas y hortalizas masivamente, preservándolas de sufrir golpes u otros daños.” [7]

No solo estos robots quedan como proyectos e investigaciones, sino que muchas empresas optan por fabricar de forma masiva e incluso abaratando costos.

Estos temas son de gran relevancia si hablamos de recoger fruto de forma automática, sin embargo también es necesario realizar la automatización antes y durante el proceso del cultivo, refiriéndonos al proceso de cuidado y manutención de la planta o sembrío. Para ello, ya se están realizando proyectos que si bien recién están empezando, muy pronto podremos ver la automatización general, desde que se siembra hasta que se cosecha.

“El principio fundamental de funcionamiento de los robots en la agricultura se basa en el reconocimiento del cultivo mediante imágenes que son procesadas y a su vez mandan órdenes precisas en cuanto a las tareas a realizar con alguna planta del cultivo como seria: aplicar fertilizante, riego, plaguicidas, realizar alguna recolección o simplemente almacenar sus datos.” [9]

Estructuras previas.

Anteriormente a los diseños que podemos ver en la actualidad, enfocados básicamente a robots locomotores de 6 ruedas, los diseños eran construidos de formas básicas con cuatro ruedas y de tamaño muy grandes.

Robotización de la maquinaria agrícola.

“Un primer paso en la robotización del sector agrícola consiste en utilizar la maquinaria que se emplea típicamente en las labores agrarias y convertirlas en vehículos autónomos. Existen algunos proyectos en los que ya se han conseguido desarrollar este tipo de sistemas, destacando las siguientes aplicaciones:” [6]

- Vehículo para tratamiento fitosanitario.
- Vehículo para plantación sobre tractores.
- Vehículo para tareas de labranza.

Las expectativas que se tenían en años anteriores reflejaban un buen futuro para los agricultores, más aún cuando la tecnología empezó a tener relevancia en este aspecto.

“Las investigaciones en robótica prometen que en los próximos años habrá una revolución en la agricultura. Integrando el GPS, sensores de movimiento y hardware y software avanzado, los nuevos robots que se están probando en el campo pueden, por sí solos, hacer el trabajo de 20 trabajadores en un día.” [11]

También podemos ver robots vehiculares básicos muy avanzados.

Un ejemplo de como la tecnología ha puesto gran interés en el campo de la agricultura tenemos al vine robot, capaz de reunir información de las plantas y sus frutos.

Vine robot.

“VineRobot cuenta con sistemas para dar información sobre el desarrollo y la composición de la uva, su desarrollo y su estado hídrico, entre otros datos” [13]

“Un consorcio de investigación europeo, formado por universidades y empresas de España, Francia, Italia y Alemania, está trabajando en el desarrollo de un pequeño robot no tripulado, equipado con sensores no invasivos avanzados y sistemas de inteligencia artificial, que ayudará a la gestión de los viñedos. El robot permitirá obtener de forma rápida información relevante como el desarrollo vegetativo, el estado hídrico, la producción y la composición de la uva, entre otros datos.” [13]



Figura 15: Vine robot: agriculturers.com

2.4 COMPONENTES DE LA PROPUESTA.

Para el desarrollo de nuestro robot móvil es necesario conocer los componentes a utilizar y como nos ayudará en nuestro ensamblaje. A continuación detallaremos cada uno de los componentes.

Componentes físicos.

Motoreductor:

Nuestro proyecto se compondrá de algunos materiales que serán de mucho peso, por ende necesitaremos motores que tengan un buen torque y de fácil manejo. Para saber el tipo de motor a utilizar, es preciso conocer el peso aproximado que van a mover los motores. Conociendo los datos se llega a la conclusión de adquirir motores con las características que podemos observar en el (Anexo 1).

Servo motor

Estos serán de importancia para el direccionamiento del robot. Al igual que los motores estos están encargado de mover mucho peso, caso contrario no cumpliría su objetivo. En este caso utilizaremos servo motores con las descripciones que se muestran en el (Anexo 1).

Tarjeta Arduino Mega.

Este sin duda es el componente de mayor importancia, ya que se encarga de la lógica y la comunicación entre el usuario y el robot. Por la cantidad de entrada y salida de información que se maneja se optó por usar el Arduino Mega, debido a la gran cantidad de pines que tiene. El detalle de este componente lo podemos encontrar en el (Anexo 1).

Puente H

Sin este componente nuestro robot no tendría movilidad, debido a que se encarga de aumentar la potencia, haciendo que pueda trabajar con mayor voltaje y corriente. Factores que la tarjeta Arduino mega no provee. Este componente ayuda en la interacción entre la tarjeta mega y la batería de 12 voltios y trabaja con amperaje máximo de 2 A, lo suficiente para hacer mover los motores. Las características lo podemos encontrar en el (Anexo 1).

Xbee S2

Su función principal es hacer la comunicación entre el usuario y el robot, los mismos que pueden encontrarse separados a distancias muy grandes. El área que cubre es suficientemente grande para el uso que se le quiere dar en los cultivos, la misma que promedia los 120m en áreas libres. Los detalles lo podemos encontrar en el (Anexo 1).

Adaptador universal para módulo Xbee.

Este componente facilita la comunicación entre al Arduino mega y el Xbee S2, debido a sus pines de conexión. Características (Anexo 1).

Tarjeta de potencia

Su tarea es distribuir de forma constante los diferentes voltajes que requiere cada componente para su funcionamiento. Ayuda en la interacción entre los componentes y la batería de 12voltios.

Sensor ultrasónico

Su función es detectar objetos a una distancia corta menores a 60cm. Sus características principales se encuentran en el (Anexo 1).

Batería seca 12 voltios.

Este componente se encargará de dar alimentación a todo el prototipo, haciendo funcionar cada uno de sus componentes. (Anexo 1). El voltaje y la cantidad de corriente dependerán del consumo al momento de desplazarse.

Nº	Detalle	valor
1	Voltaje máximo correspondiente a los motores	12V
2	Corriente de consumo abriendo el circuito	2.30A. min-3A. Máx.
3	Datos de la Batería a utilizar	12v-5A

Tabla 1: Detalle de voltaje y corriente

Cables.

Estos se encargan de llevar la corriente a los distintos componentes del robot. Es necesario conocer qué tipo de cables usar, y que puedan soportar la corriente que pasa en cada sección.

Cooler o ventilador

Este componente ayudará en el enfriamiento de los componentes de la tarjeta de potencia, ya que por este módulo pasa mucha carga y se calienta demasiado.

Cámara ip

Esta cámara es necesaria en el campo de aplicación, debido a que podemos cerciorarnos visualmente de que el vehículo no tenga ningún imprevisto.

Componentes Lógicos.

Arduino 1.6.5

Es la versión actualizada del software Arduino, aquí podremos realizar la codificación que nos ayudará en la manipulación del prototipo.

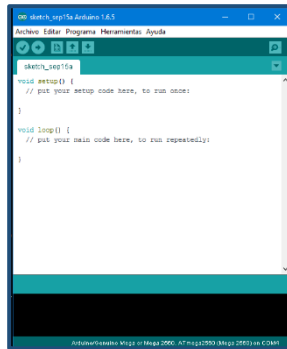


Figura 16: IDE Arduino

Visual Studio.

En este programa podremos crear la interfaz gráfica que se requiere para que el usuario pueda entablar comunicación con el robot móvil.

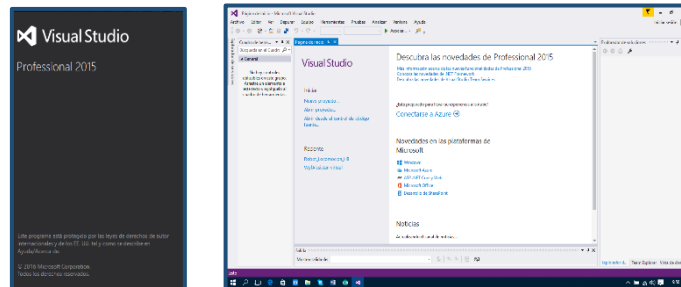


Figura 17: Inicio programa Visual studio 2015

Este programa nos ofrece dos áreas de trabajo: Form1.vb [diseño] para diseñar la interfaz, y Form1.vb para realizar la codificación.

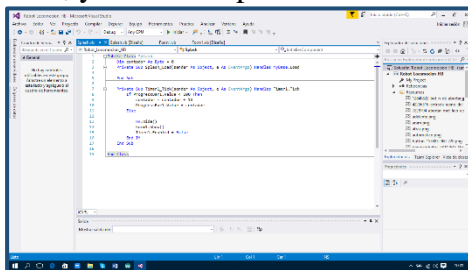


Figura 18: Área de codificación.

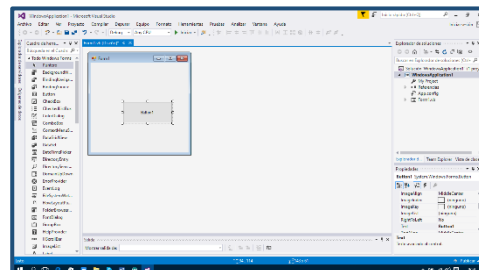


Figura 19: Área de diseño

MVCMS_Lite

Su función es visualizar la cámara en el monitor

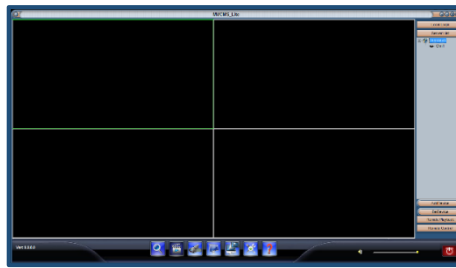


Figura 20: Programa de visualización cámara

V380

Aplicativo que puede ser usado para manipular la cámara a través del teléfono.



Figura 21: Aplicativo para teléfono.

X-CTU SOFTWARE

Esta versión permite configurar el dispositivo Xbee S2, la cual nos permiten la transmisión y recepción de datos.

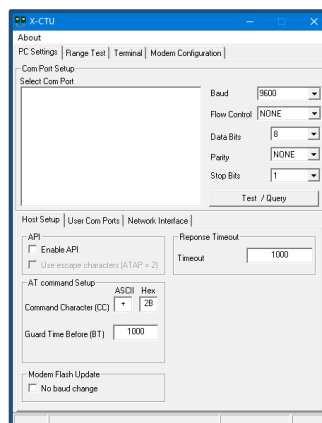


Figura 22: Programa X-CTU

2.5 DISEÑO DE LA PROPUESTA.

A continuación veremos las diferencias de 3 diseños, las mismas que nos ayudará a escoger la mejor opción en nuestro proceso de elaboración de nuestro móvil. Debemos destacar su fijación al suelo y el peso que puede soportar.



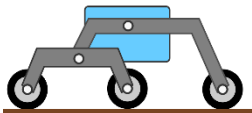
DISEÑO	RUEDAS	BASE	Descripción.
	4 Ruedas	Si utiliza. Pequeña. Una base.	Tipo normal. Rígido, movimiento uniforme. Adaptable a suelo liso
	6 Ruedas	Si utiliza. Grande. Doble base.	Movimientos doble. Difícil mantener base estable. Adaptable a suelos no liso.
	6 Ruedas	Si utiliza. Pequeña/grande. Una base.	Equilibrio de base. Adaptable a suelos no liso.

Tabla 2: Comparación de diseños

Visualización del proyecto que se va a realizar.

En términos generales hemos vistos los tipos de vehículo que intervienen en los distintos campos. Esto nos da apertura al tipo y tamaño de robot móvil que queremos diseñar. A continuación veremos un bosquejo del proyecto a realizar.

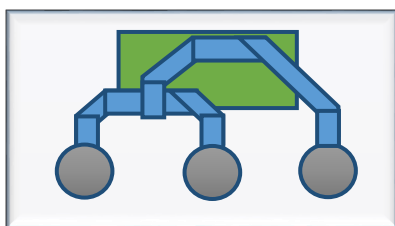


Figura 24: Visualización del prototipo de parte lateral.

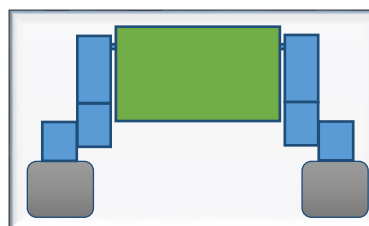


Figura 23: Visualización del prototipo de parte frontal.

En la figura de la izquierda podemos ver el robot móvil vista en forma lateral, compuesta de base, soporte y ruedas. En la figura de la derecha nos muestra el robot vista de manera frontal

DISEÑO MECÁNICO DEL ROBOT DE LOCOMOCIÓN.

El diseño mecánico nos permite definir las dimensiones del prototipo de robot de locomoción.

- El chasis donde se encuentran la parte de circuitería y la parte de alimentación.
- Las estructuras que sirven de soporte entre las ruedas y el chasis.
- La suspensión para la estabilidad del robot móvil.
- Las ruedas para el desplazamiento.

Para empezar es necesario hacer los cálculos correspondientes de acuerdo al tamaño que se especifique, nos haremos referencia al prototipo del “robot fumigador” [20], las cuales sus medidas son: 40cm de anchura y 50 cm de largo.

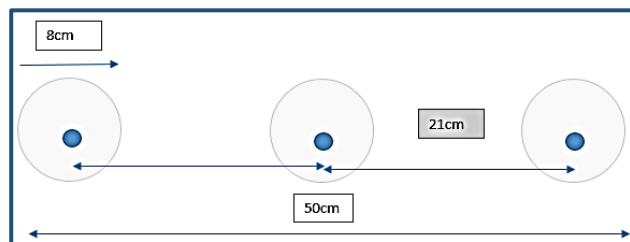
Distancia entre ruedas.

Dada la longitud máxima del robot móvil ($L_{TOTAL}=50cm$) y conociendo el diámetro ($D_{RUEDA}=8cm$) de nuestras ruedas procedemos a utilizar la siguiente fórmula:

$$L_r = \frac{L_{TOTAL} - D_{RUEDA}}{2}$$

$$L_r = \frac{50cm - 8cm}{2}$$

$$L_r = 21cm$$



Ecuación 2: Distancia entre las ruedas.

Figura 25: Distancia entre ruedas.

Cálculo de medidas de soporte rocker-bogie.

Calculamos usando el método de Tales de Mileto y funciones trigonométricas.

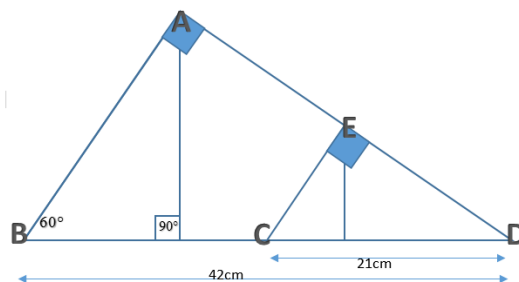


Figura 26: Datos dados por la propuesta.

Usamos funciones trigonométricas:

$\text{sen } \beta = \frac{O}{H}$	$\text{cos } \beta = \frac{A}{H}$	$\text{tan} \beta = \frac{O}{A}$
$\text{sen } 60^\circ = \frac{O}{42\text{cm}}$	$H(\text{cos } 60^\circ) = A$	$\beta = \frac{\text{tan}^{-1}(O)}{A}$
$O = 42(\text{sen}60^\circ)$	$42(\text{cos } 60^\circ) = A$	$\beta = \frac{\text{tan}^{-1}(36,37)}{21}$
$O = 36,373\text{cm}$	$21\text{cm} = A$	$\beta = 60^\circ$

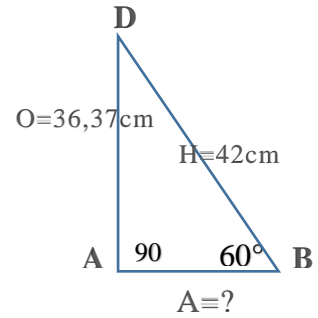


Figura 27: Datos de puntos A, B, D.

Ecuación 3: Ecuación de medidas generales

Cálculo de medidas internas:

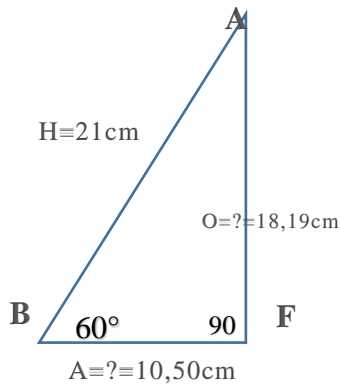


Figura 28: Datos de B, A, F.

$\text{sen } \theta = \frac{O}{H}$	$\text{tan} \theta = \frac{O}{A}$
$H(\text{sen } 60^\circ) = O$	$A = \frac{O}{\text{tan} \theta}$
$O = 21(\text{sen}60^\circ)$	$A = \frac{18,19}{\text{tan}60^\circ}$
$O = 18,186\text{cm}$	$A = 10,50\text{cm}$

Ecuación 4: Cálculo entre A, B, F.

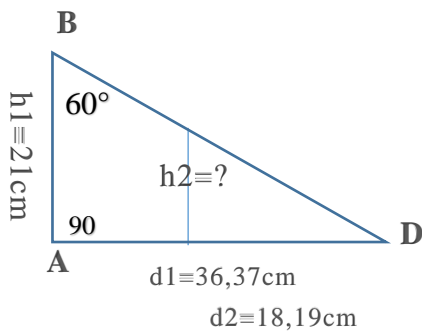


Figura 29: Datos de A, B, D.

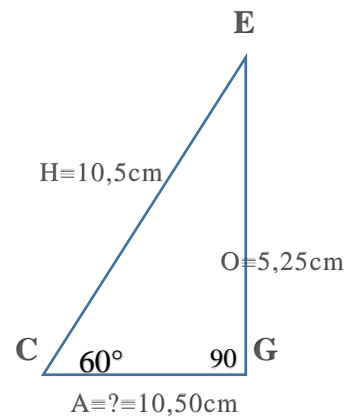


Figura 30: Datos de C, E, G.

$\frac{h1}{h2} = \frac{d1}{d2}$ $\frac{21cm}{h2} = \frac{36,37cm}{18,19cm}$ $h2(36,37) = 21(18,19)cm$ $h2 = \frac{381,99}{36,37}$ $h2 = 10,50cm$	$\cos 60^\circ = \frac{h2}{H}$ $h2 = 10,50cm$ $h2 = 10,50(\cos 60^\circ)cm$ $h2 = 5,25cm$ $A = \frac{0}{\tan 60^\circ} = \frac{5,25cm}{\tan 60^\circ} A = 3,03cm$
--	---

Ecuación 5: Cálculo entre C, E, G

Al final obtenemos las siguientes medidas para elaborar el soporte rocker-bogie.

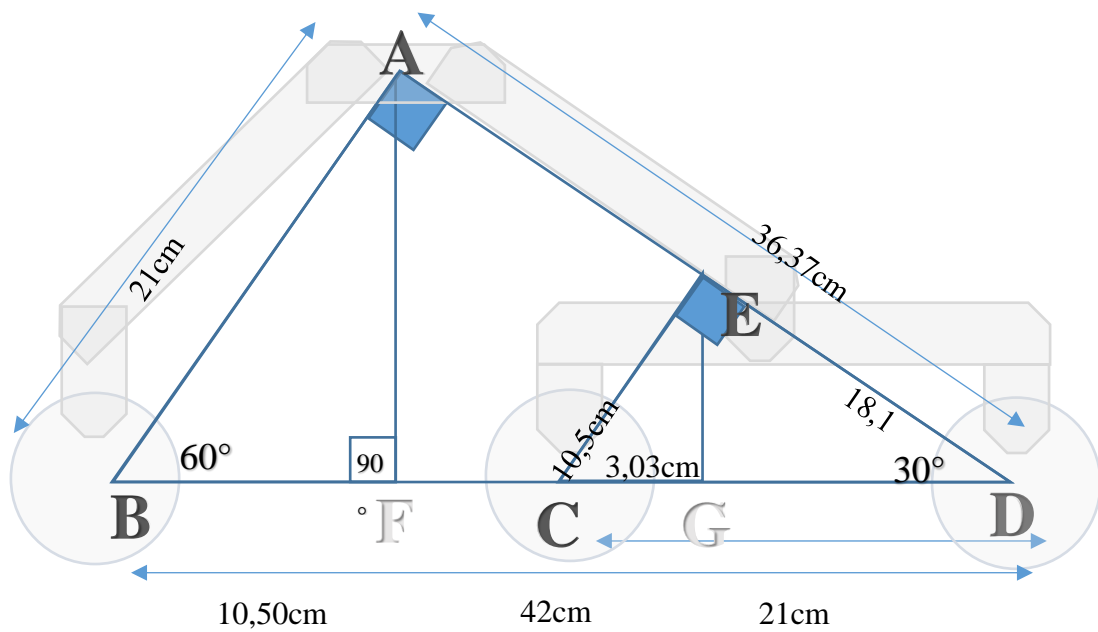


Figura 31: Medidas para el diseño de robot.

Dimensiones del soporte rocker-bogie.

Una vez conocido los resultados, se procede a sacar la medidas de los soportes, esto dependerá del tamaño del aluminio. En este caso usaremos Aluminio cuadrado de 1 pulg (2,5cm).

ROCKER

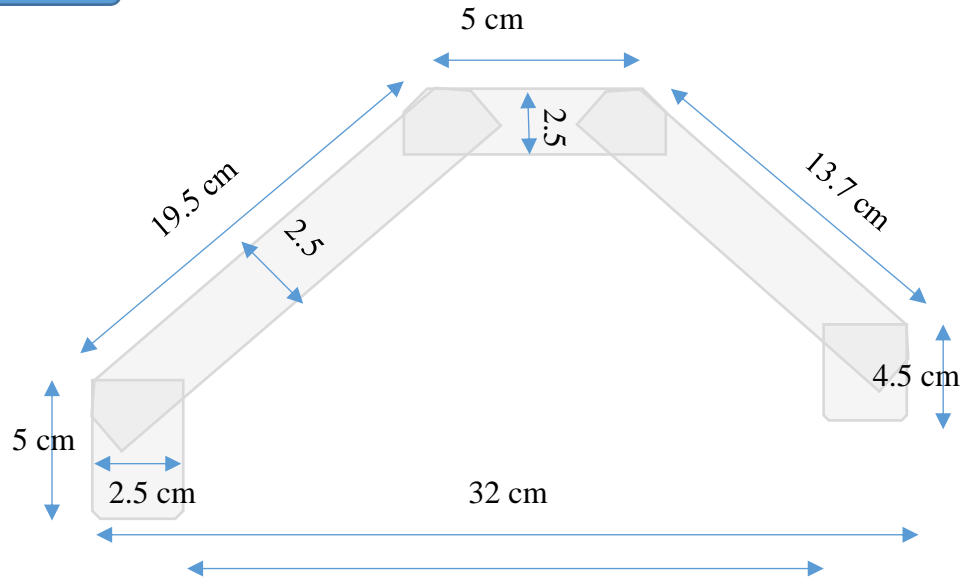


Figura 32: Medidas del rocker.

BOGIE

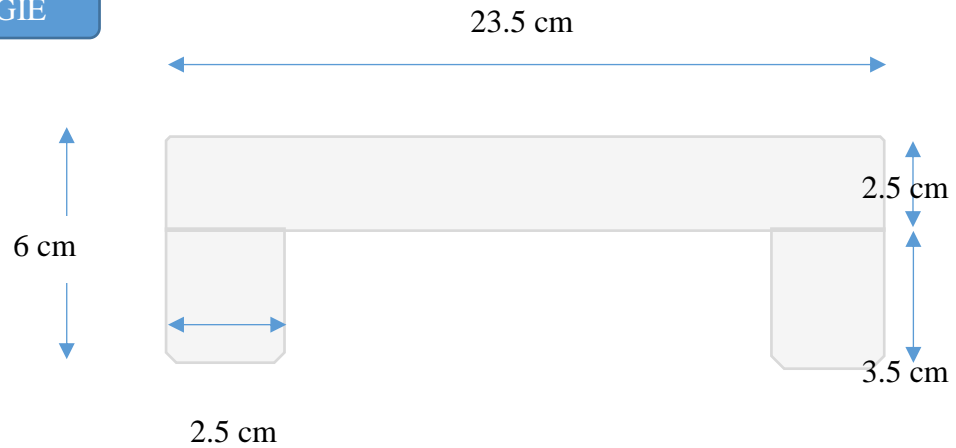


Figura 33: Medidas del Bogie.

Chasis

Es necesario saber el tipo de material que debemos usar para la elaboración de nuestro robot. Para su construcción se deberá escoger entre varias alternativas según sus características.

- Aluminio natural
- Acrílico:
- Madera:

La opción a considerar para la construcción de nuestra base es la madera, la misma que se destaca por ser económica y fácil de diseñar.



Figura 34: Tipo de madera: Masisa

A continuación procedemos a dar medidas para la elaboración de la base.

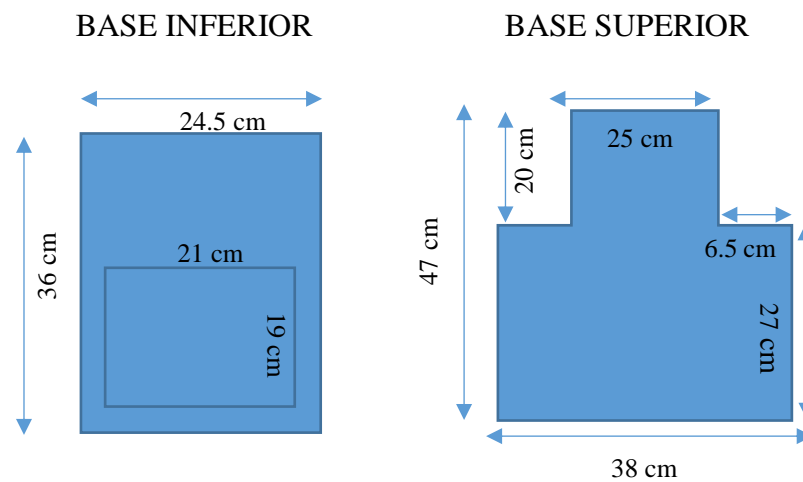


Figura 35: Medidas base parte inferior y superior.

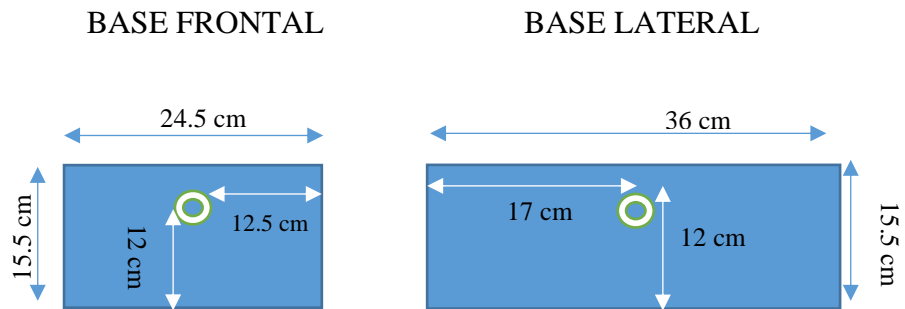


Figura 36: Medidas base frontal y lateral.

MEDIDAS MINICHUMACERAS

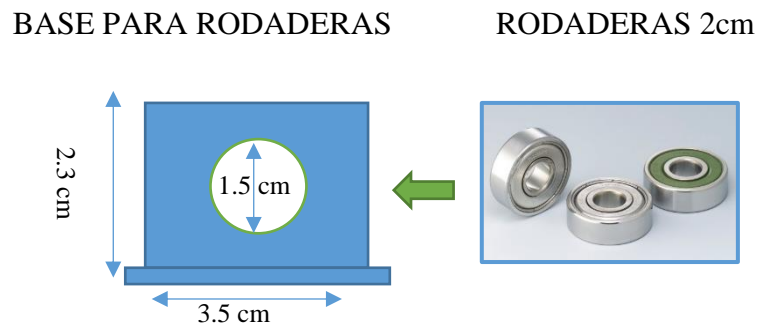


Figura 37: Medidas mini chumacera.

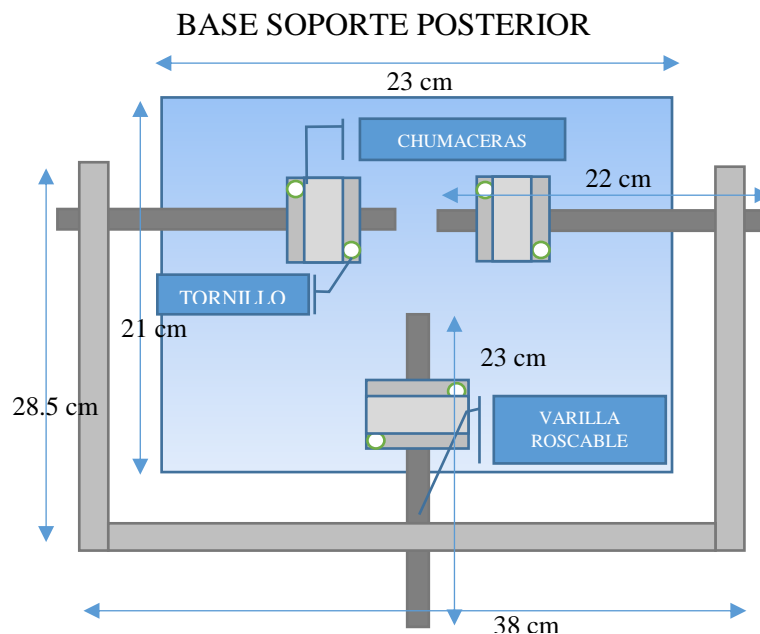


Figura 38: Base soporte posterior.

Luego de tener todas las medidas, procedemos con su elaboración, por tanto necesitamos conseguir los materiales.

DETALLES	PROPORCION
Madera con laminado blanco	1m x 2m (ancho-largo)
Barras de aluminio tipo L 1pulg.	3m
Barras de aluminio cuadrado 1pulg.	2m
Pernos 7cm	2 unidades
Arandelas	4 unidades
Tuercas	11 unidades
Pernos 2cm	38 unidades
Pernos 1cm	20 unidades
Tuercas pequeñas 0.5 cm	60 unidades
Rodaderas 2cm	3 unidades
Remaches 1.5 cm	20 unidades
Barra de aluminio tipo L 1x1cm	2m

Tabla 3: Materiales para armar la base.

Una vez listo todos los materiales y hechas las medidas necesarias, se procederá con el armado de su base, esto lo podemos ver paso a paso en el (Anexo 6).

Equilibrio punto medio

El equilibrio de toda estructura depende en la distribución equitativa del peso que se coloque, tanto en la parte interna como externa de la base. Para mayor estabilidad se proporciona un soporte posterior conectado a la base evitando de esta manera movimientos exagerados.

Punto de equilibrio de la base.

Después de armar la base podemos conocer su masa, la misma que nos ayudará en el equilibrio al momento de distribuir el peso total en un punto medio.

Es preciso conocer la fórmula a utilizar, para realizar el cálculo.

$$P = m \cdot a$$

Ecuación 6: Fórmula de Peso.

Conociendo los siguientes datos podemos equilibrar la base general del robot.

Base exterior, cubierta (chasis).	7.70 libras	34.30 N
Base interior	0.50 libras	2.23 N
Base circuito	0.20 libras	2.23 N
Batería	3.00 libras	13.35 N
Tarjeta Arduino mega.	0.08 libras	0.36 N
Tarjeta de potencia	0.09 libras	0.40 N
Puente h/ cables/otros.	0.50 libras	2.22 N
cámara	0.92 libras	4.09 N
Total	13.30 libras	59.18 N

Tabla 4: Pesos de componentes.

$$1 \text{ lbf} = 4.4482216152605 \text{ N}$$

$$12 \text{ lb} = 5.44 \text{ kg}$$

Ecuación 7: Conversión a kg

$$P = m \cdot a$$

$$P = (5.44 \text{ kg}) \left(\frac{9.8 \text{ m}}{\text{s}} \right)$$

$$P = 53.31 \text{ N}$$

Ecuación 8: Peso de la base del prototipo

Ahora podemos dividir la fuerza en dos, donde el punto central es de 29.59N.

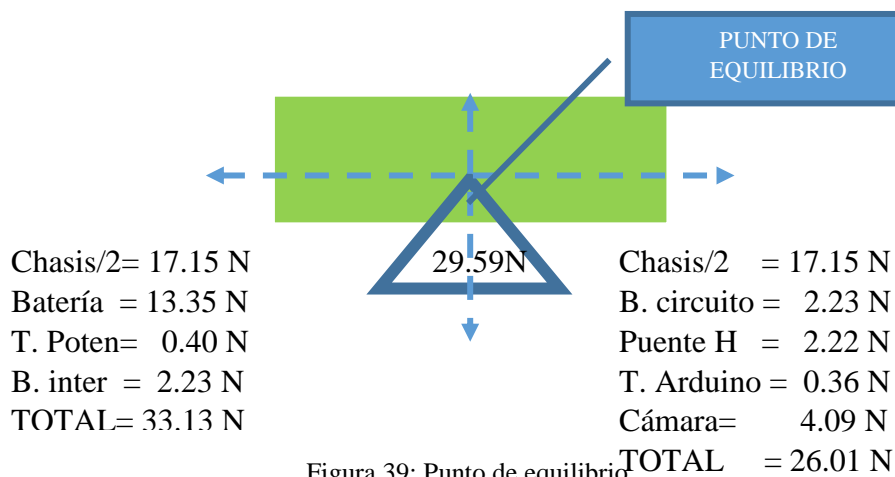


Figura 39: Punto de equilibrio.

Carga por rueda

Datos:

$$C_{total} = 18 \text{ libras} + 2 \text{ libras}$$

$$\#ruedas = 6 \text{ ruedas}$$

$$C_{rueda} = \frac{C_{total}}{\#ruedas}$$

$$C_{rueda} = \frac{20 \text{ libras}}{6}$$

$$C_{rueda} = 3,3 \text{ libras}$$

Ecuación 9: Cálculo de carga para cada llanta.

Cálculo para escoger las características del motor.

Fuerza Rotacional = Torque = τ

$$\vec{\tau}_0 = \vec{r} \times \vec{F}$$

$$\tau = r \cdot F \cdot \sin(90)$$

$$\tau = r \cdot F$$

$$F = m \cdot a$$

$$\tau = [N \cdot m]$$

Ecuación 10: Fórmula de Torque

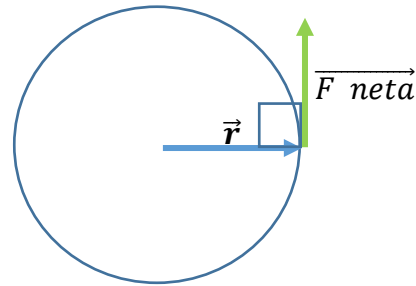


Figura 40: Torque de rueda.

Esta fórmula nos ayudará a obtener el torque individual necesario para mover todo el prototipo (carga por rueda 1.5kg=14,7N).

$$\tau = r \cdot F$$

$$\tau = 4cm \cdot (14,71N)$$

$$\tau = 58,84N \cdot cm$$

$$\tau = 0,5884N \cdot m$$

$$\tau = 6kg - cm$$

Ecuación 11: Cálculo de torque de nuestra rueda

Diseño soporte de rueda y angulo de desviacion del movil.

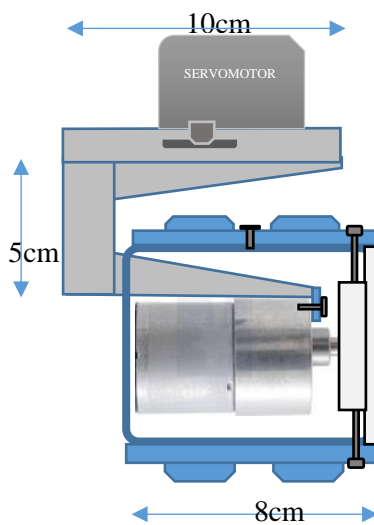


Figura 42: Diseño soporte de rueda.

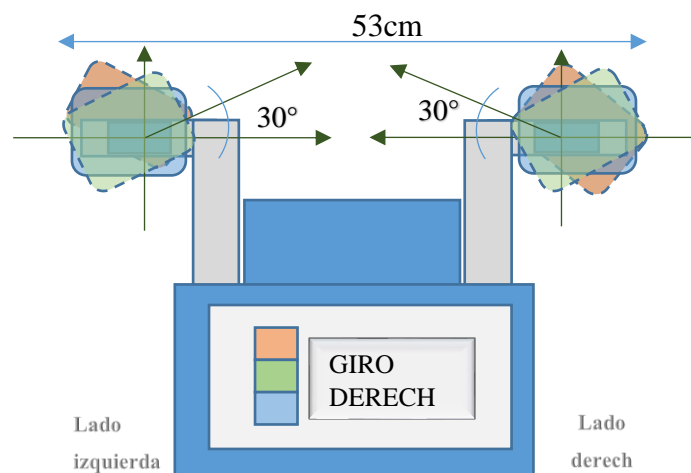


Figura 41: Movilidad de servomotores izquierda y derecha.

DISEÑO ELECTRÓNICO DEL ROBOT DE LOCOMOCIÓN

Un robot es un conjunto de sistemas que están organizados de tal forma que responden con una acción a cualquier orden dada externamente. A continuación se detallarán algunos de estos sistemas.

Sistema de alimentación

Este sistema se encarga de distribuir la tensión de manera medible para el funcionamiento de los dispositivos. Como fuente de alimentación principal tenemos la batería de 12 voltios a 5 amperios, y para repartir la tensión que necesita cada componente, se tuvo que regular a distintos voltaje de salida con el siguiente circuito.

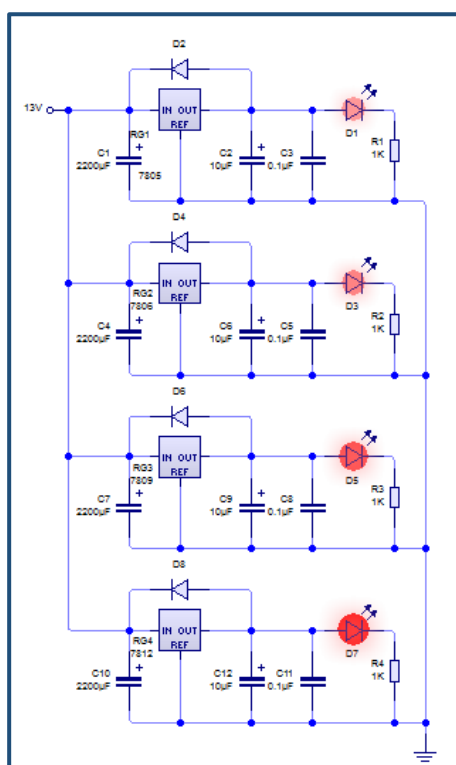


Figura 43: Regulador de voltaje.

En primera instancia podemos observar que utiliza un regulador LM7805 (Anexo 4), que nos arroja un voltaje de 5v dc. En paralelo se conectó un diodo rectificador de silicio 1n4007, que nos sirve de protección para nuestro componente regulador, utiliza un capacitor electrolítico de 2200uf, uno de 10uf y un capacitor cerámico de 0.1uf para equilibrar la corriente que recibe. De igual manera tenemos reguladores de 6,9y12v con la misma configuración para sus salidas. En la siguiente figura podemos ver nuestro regulador con las diferentes salidas de voltaje. Tiene una conexión de entrada para los 12v de la batería.

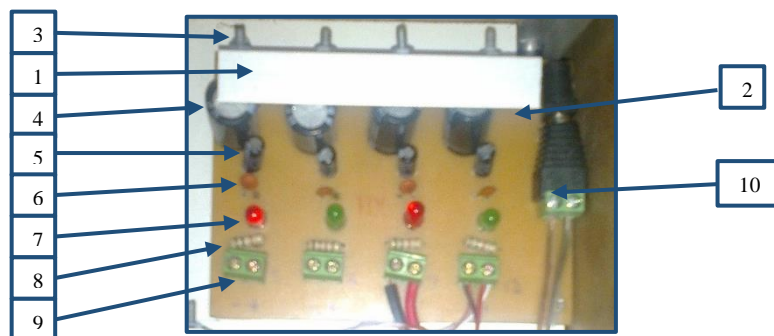


Figura 44: Circuito regulador de voltaje.

- | | |
|------------------------------------|---------------------------------------|
| 1.-Disipador | 7.-Diodos LEDs (rojo, verde) |
| 2.-Diodos rectificadores 1n4007 | 8.-señal de salida |
| 3.-Regulador de tensión (5-6-9-12) | 9.-Resistencias 1k |
| 4.-Capacitor electrolítico 2200uf | 10.-Salidas 5-6-9-12 voltios 1amp |
| 5.-Capacitor electrolítico 10uf | 11.-Entrada 12v de la batería a 5 amp |
| 6.-Capacitor cerámico 0.1uf | |

Circuito encendido de luces 12v frontales conectados al Arduino

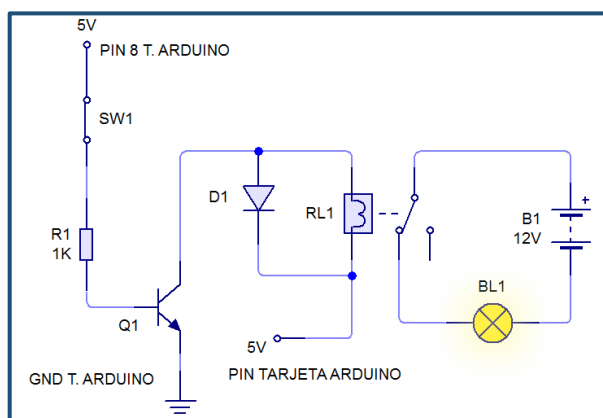


Figura 45: Circuito encendido de luces con conexión a la tarjeta Arduino.

Esta figura nos muestra el funcionamiento de las luces frontales en el simulador Circuit Wizard. Debido a que las luces LEDs son de 12voltios y nuestra tarjeta Arduino solo nos arroja 5v, no es posible hacer la conexión directamente, por esta razón se tuvo que utilizar relés de 12v para que haga la interacción entre las dos partes. Utiliza un transistor 2n3904 (Anexo 5), donde su base se conecta al pin 8 de

la tarjeta Arduino, donde su dato de salida nos sirve como gatillado para activar el relé que está conectado a 12voltios.

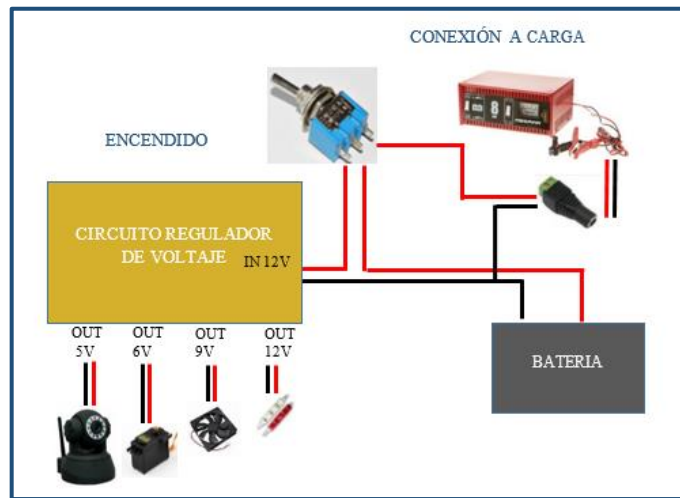


Figura 46: Conexión de batería y dispositivos

En esta figura podemos observar la conexión de los diferentes dispositivos (cámara, servo motor, ventilador, luces) con el circuito regulador de voltaje. Cuenta con un Switch de tres estados que aísla el funcionamiento normal con el proceso de carga, se lo hace con el fin de proteger el circuito interno.

A continuación se muestra en *circuit Wizard*, el circuito de conexión entre la batería y los dispositivos aislados con el cargador mediante el Switch.

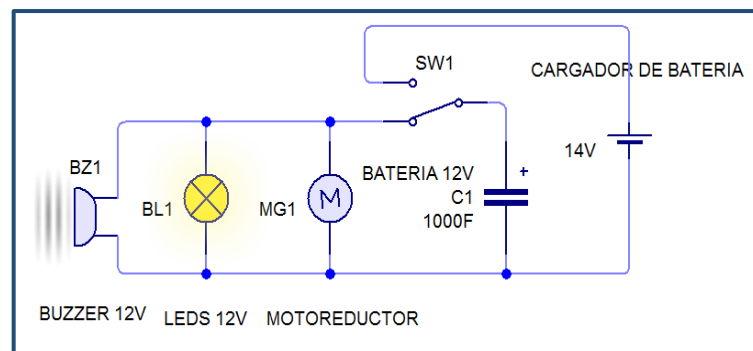


Figura 47: Circuito de conexión de batería y dispositivos en Circuit Wizard.

Cabe indicar que los componentes no están conectados directamente, sino que deben pasar por el regulador de voltaje respectivo.

Sistema de potencia.

Nuestro proyecto consta de actuadores como el motoreductor que necesita mucha potencia para su funcionamiento. Para esto será necesario la utilización del integrado L298N, que la podemos encontrar en el puente H de Arduino. Este dispositivo deberá estar conectado entre la tarjeta Arduino y los motores. A continuación mostraremos como mover un motor a través del puente h.

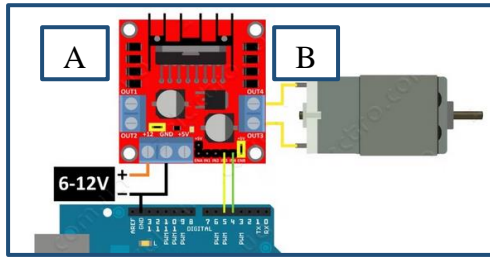


Figura 49: Conexión motoreductor, l298n y T. Arduino

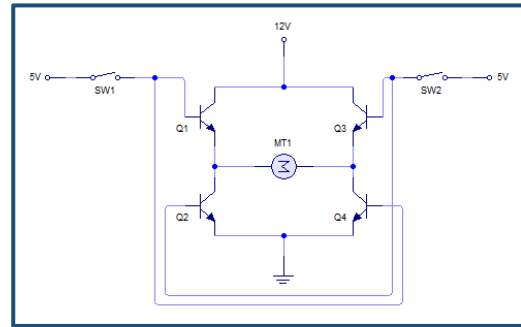


Figura 48: Conexión del motoreductor y puente h

Con el fin de que el motor B pueda moverse, el puente h deberá recibir información en los pines (IN3, IN4) que son enviadas por la tarjeta Arduino a través de los pines 13 y 14. Para obtener una dirección (ej. ADELANTE), el pin IN3=1 y el pin IN4=0 esto activa un par de transistores como podemos apreciar en las figuras.

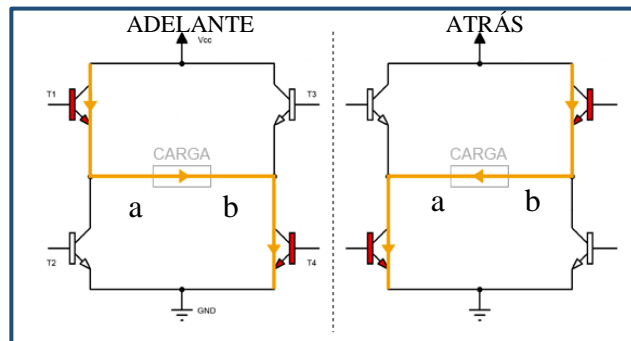


Figura 50: Activación de transistores para direccionamiento.

El pin IN3 activa los transistores (T1 -T4) y el pin IN4 por estar en estado LOW mantiene a los transistores restantes desactivados, esto hace que la corriente fluya de a =>b. Si queremos que se mueva hacia atrás, el pin IN3 deberá estar en "0" mientras que el pin IN4 deberá estar en "1", haciendo que la corriente fluya de b=a. Esta configuración se lo emplea en los 6 motores de nuestro proyecto.

Esta placa es muy práctica ya que incorpora una electrónica que simplifica las conexiones, aislando las salidas de los motores A y B evitando algún cortocircuito entre ellos.

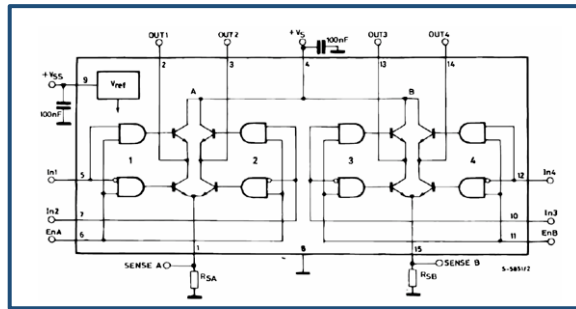


Figura 51: Electrónica interna del dispositivo: Integrado 1298n

A continuación se puede apreciar la conexión entre el puente h y el motoreductor.

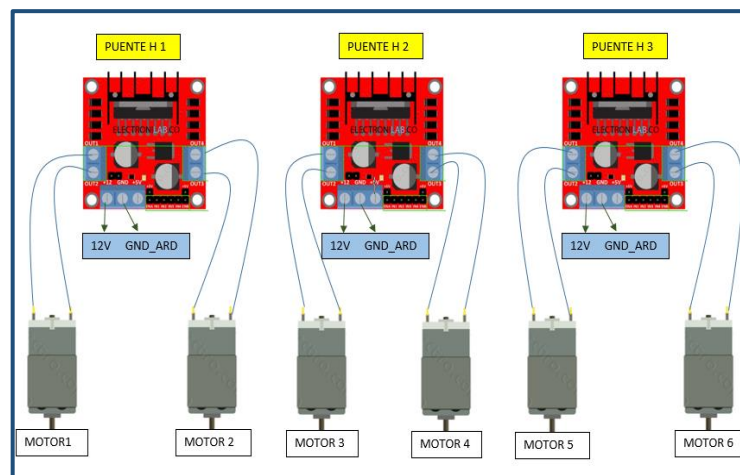


Figura 52: Conexión de puente h con los motores.

Para el funcionamiento general de los motores se tuvo que dividir en tres grupos. Cada grupo contiene un puente H (Anexo 2), que se encarga de controlar 2 motores. Esto nos sirve para evitar sobrecargas en cada uno de los puentes, debido a que sus motores pueden consumir mucha corriente. La división en grupos hace que la corriente se reparta equitativamente. Cada puente H tiene una entrada de 12 voltios que se conectan a la batería (12V) y un tope máximo de 2 Amp como salidas (OUT1, OUT2, OUT3, OUT4), necesarios para el movimiento de los dos motores. Para el cierre del circuito se hace una conexión (GND_ARD) a un pin tierra (GND) de la tarjeta Arduino.

Sistema de control.

Básicamente es el cerebro de nuestro proyecto y se convierte en el responsable de determinar todas las acciones. En esta ocasión utilizaremos la tarjeta Arduino mega como controlador.

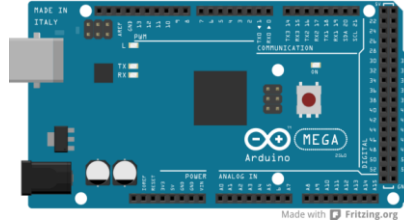


Figura 53: Tarjeta Arduino Mega como controlador

Esta tarjeta se encargara de la activación de los pines entradas y salida configuradas en su código. Nuestra tarjeta se conectará con el sistema de potencia de la siguiente manera.

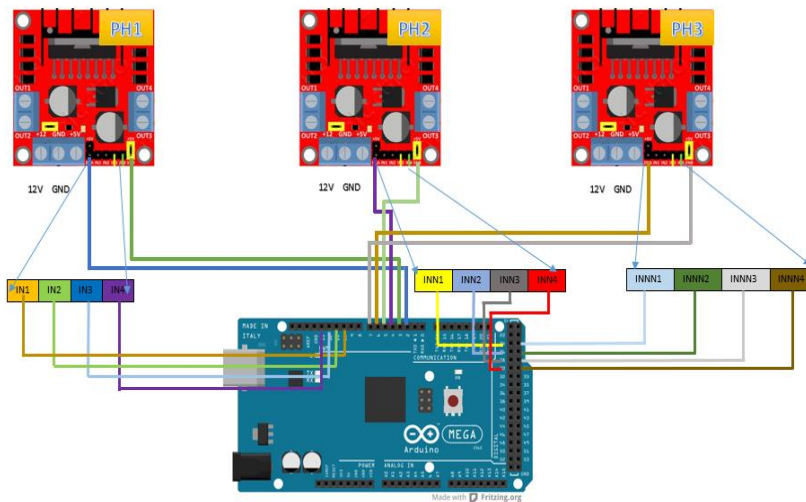


Figura 54: Conexión de puentes h con la tarjeta Arduino.

Aquí podemos observar los pines (IN1, IN2, IN3, IN4) del puente H1 que se conectan a los pines de la tarjeta Arduino (10, 11, 12, 13) respectivamente. Cada par (IN1, IN2...ej.) de estos pines se encarga del movimiento de un motor.

Para activar el movimiento de cada par, deben estar habilitadas los pines (ENA, ENB) del puente H, las mismas que están conectadas a los pines PWM (2, 3) de la tarjeta Arduino. De igual forma se hacen las conexiones con los puentes H (2 y 3) utilizando los pines PWM (4,5 y 6,7) respectivamente. También utilizamos los pines (24, 26, 28, 30) para PH2 y los pines (25, 27, 29, 31) para PH3. Cada uno de estos pines PWM (2, 3, 4, 5, 6, 7) nos ayudará a regular la velocidad de los motores.

Sistema sensorial

Esta parte se encarga de recoger información acerca de su entorno

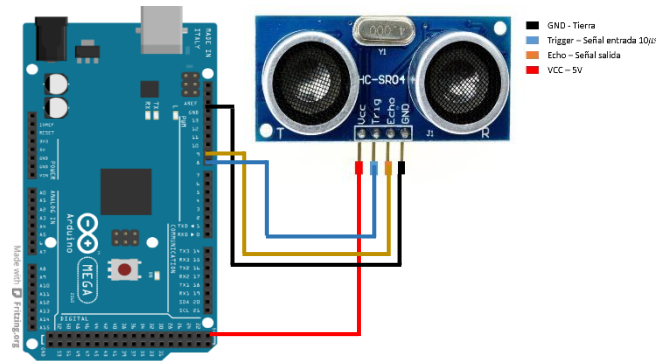


Figura 55: Conexión del módulo xbee y sensor ultrasónico.

En primera instancia podemos observar la conexión del sensor ultrasónico. Los pines (VCC, ECHO, TRIG, GND) del sensor ultrasónico van conectados a los pines (+5v, 9, 8, GND) de la tarjeta Arduino (Anexo 3) respectivamente. También se conectan los pines (Tx0, Rx0) del Arduino a los pines (Tx, Rx) del módulo Xbee, al igual que los pines de 5v y Gnd.

Funcionamiento

El sensor mide el tiempo entre el envío y la recepción de un pulso sonoro. Para calcular la distancia debemos saber la velocidad del sonido que a temperatura de 20° C es 343m/s. transformando tenemos.

$$343 \frac{m}{s} \cdot 100 \frac{cm}{m} \cdot \frac{1s}{1000000us} = \frac{1}{29.2} \frac{cm}{us}$$

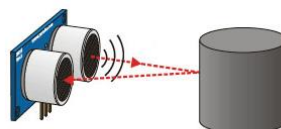
Ecuación 12: Velocidad del sonido

Es decir, que para recorrer 1cm el sonido tarda 29,2 microsegundos, por ende la fórmula de la distancia sería.

$$Distancia(cm) = \frac{tiempo(us)}{29.2(2)}$$

Ecuación 13: Fórmula de la distancia para el sensor

Debido a que hemos calculado el tiempo en enviar y recibir un pulso, entonces tenemos que dividir esta para 2



$$\begin{aligned} \text{Tiempo} &= 2 * (\text{Distancia} / \text{Velocidad}) \\ \text{Distancia} &= \text{Tiempo} \cdot \text{Velocidad} / 2 \end{aligned}$$

Figura 56: Recorrido del pulso en el sensor

Sistema de comunicación.

Nuestro proyecto necesita de un sistema de comunicación que le permita al usuario manejarlo a cierta distancia. Para esto utilizaremos los dispositivos inalámbricos XBee2. Este dispositivo no trabaja solo, deberá ir conectado a un adaptador, en nuestro caso utilizaremos el “Adaptador universal para módulo Xbee2”.

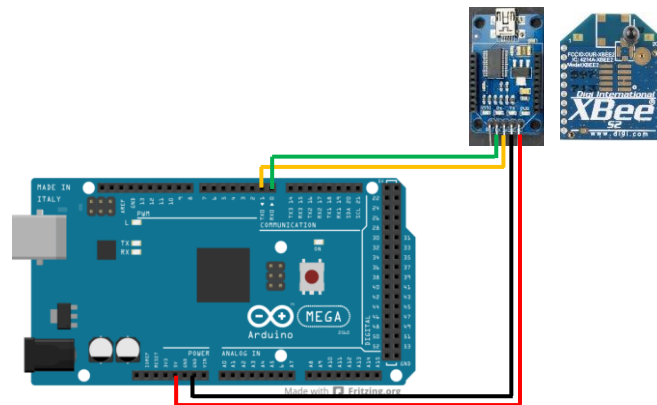


Figura 57: Conexión Xbee-Arduino

El módulo xbee 2 tiene los Pines de 5v, GND, Tx, y Rx, las cuales van conectados a los pines de 5v, GND, Tx, y Rx de la tarjeta Arduino MEGA.

Es muy importante mantener bien fija esta conexión, debido a que nuestro móvil experimenta muchos movimientos y podría desconectarse.



Figura 58: Conexión Pc- Xbee

También haremos la conexión entre Pc y el módulo xbee mediante un cable USB mini. Las características de nuestro adaptador USB universal hace que las conexiones sean sencillas, ya que consta de pines para conectar al xbee, así como el conector USB para adaptarlo a la pc

La configuración para hacer la paridad entre los 2 módulos Xbee se lo hará mediante el software X-CTU

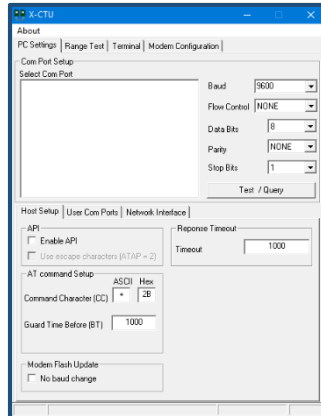


Figura 59: Programa X-CTU para configuración

Test de una comunicación PC => Xbee

Lo que se hará es establecer un dialogo entre dos Xbee, una conectada remotamente a una tarjeta Arduino Mega (Receptor) y la otra conectada a la pc por medio del puerto USB (Emisor), su comunicación será dúplex, es decir bidireccional.

Primero programaremos los módulos de acuerdo a la siguiente tabla:

	Xbee Emisor	Xbee Receptor	Descripción
PAN ID	9032	9032	Red que se pretende montar.
DH	0	0	Dirección de destino parte alta (emisor)
DL	1	0	Dirección de destino parte baja (receptor)
MY	0	1	Dirección con la que se comunica el modulo

Tabla 5: Asignación de datos a los XBee

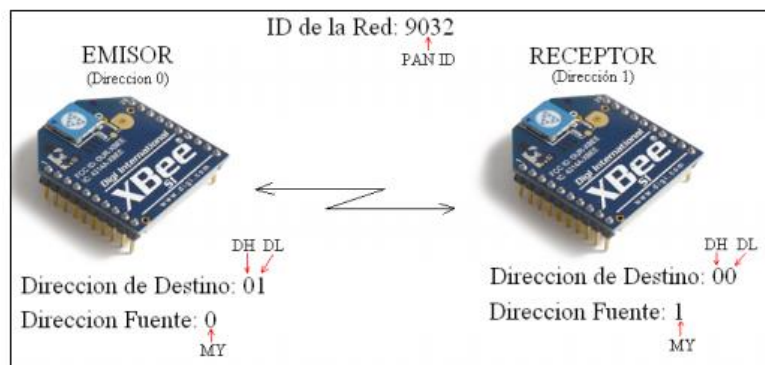


Figura 60: Datos Programados en los XBee

Configuración del módulo Xbee.

Abrimos la aplicación y en la opción “Modem Configuration” presionamos el botón “Read” para leer el tipo de dispositivo conectado. Luego le damos los siguientes parámetros:

Emisor:

PAN ID	escribimos 9032
DL Destination Address High	escribimos 0
DH Destination Address Low	escribimos 1
MY 16-bit Source Address	escribimos 0
BD Interface Rate Date	escribimos 9200 (baudios)

De igual forma se procede con el receptor

Receptor:

PAN ID	escribimos 9032
DL Destination Address High	escribimos 0
DH Destination Address Low	escribimos 0
MY 16-bit Source Address	escribimos 1
BD Interface Rate Date	escribimos 9200 (baudios)

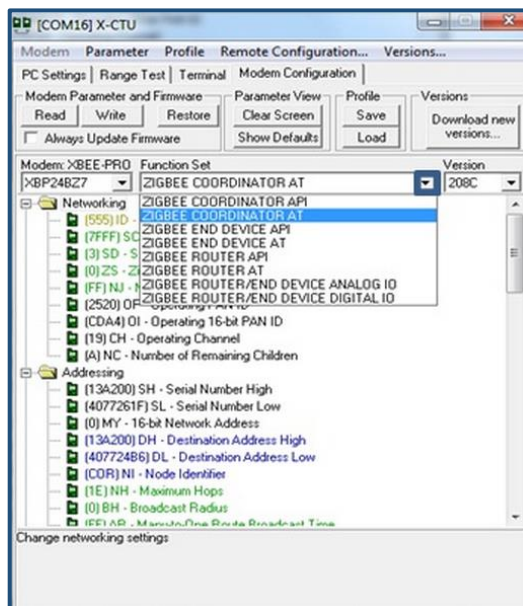


Figura 61: Configuración XBee.

Sistema de accionamiento

En este sistema podemos encontrar elementos actuadores como el motoreductor y el servo motor que son de tipo eléctrico. Estos se encargan de dar fuerza sobre las partes del robot haciendo que se muevan, convirtiendo la energía eléctrica en movimiento rotacional. Sus conexiones son las siguientes.

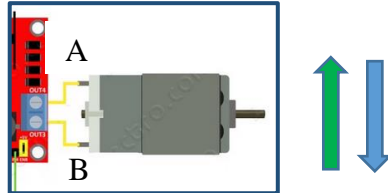


Figura 62: Terminales del Motoreductor

El motoreductor tiene dos terminales (A, B) que se conectan a las salidas (OUT4, OUT3) del L298N, y son controladas mediante el Arduino a través de sus conexiones con los pines 2,3,4,5,6 de la tarjeta mega.

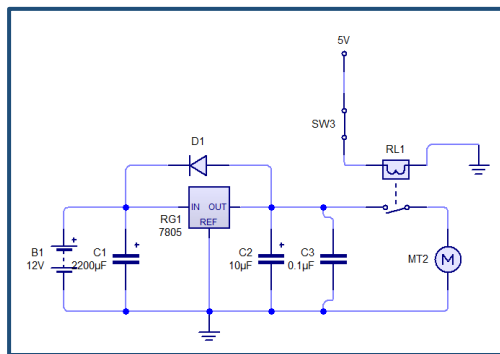


Figura 64: Conexión de servo motor

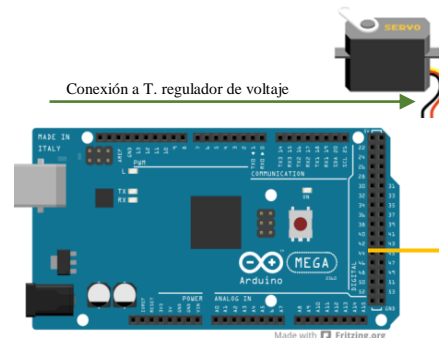


Figura 63: Conexión a tarjeta Arduino

También podemos ver las conexiones de nuestro servo MG995 que se alimenta con 6V, este voltaje lo obtenemos de nuestro circuito regulador 7806. Si le conectamos a mayor voltaje el circuito de nuestro servo sufrirá daños. Para obtener los datos del Angulo lo hacemos a través del pin 44 del Arduino.

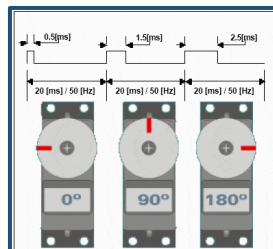


Figura 65: Posicionamiento del servomotor

Este servo nos ofrece un ángulo máximo de 180°, nuestro proyecto utiliza un ángulo de 30° para su desplazamiento hacia la izquierda y derecha.

DISEÑO LÓGICO DEL ROBOT DE LOCOMOCIÓN.

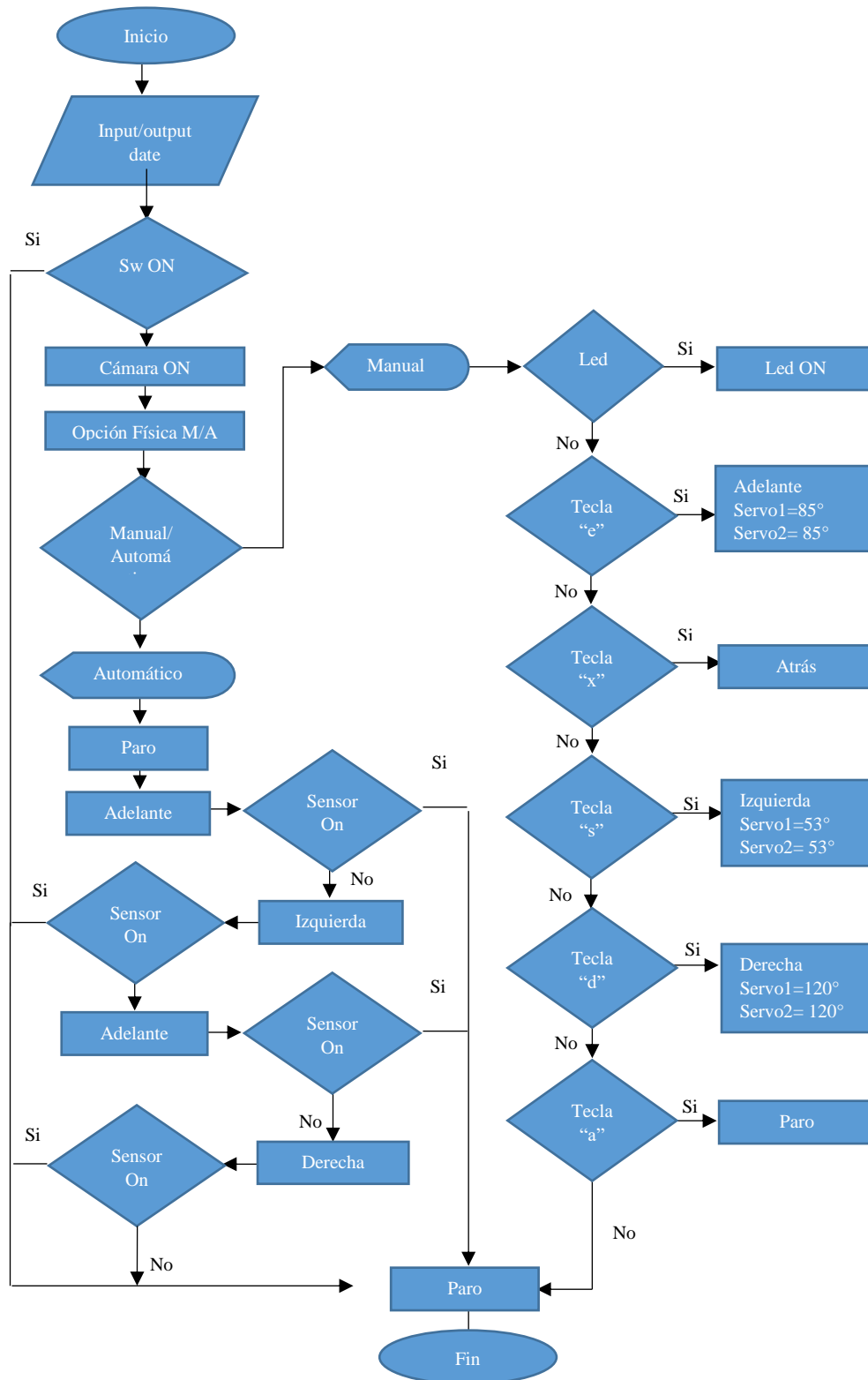


Figura 66: Diagrama de flujo

Resumen de pines de la tarjeta Arduino.

Nº	Descripción	Variable	Entrada/ salida	Pin Ard.
1	Envío de datos para módulo XBee	Tx	OUTPUT	0
2	Recepción de datos para módulo XBee	Rx	INPUT	1
3	Activa los pines IN1,IN2 del puente h1	ENA	OUTPUT	2
4	Activa los pines IN3,IN4 del puente h1	ENB	OUTPUT	3
5	Activa los pines IN1,IN2 del puente h2	ENC	OUTPUT	4
6	Activa los pines IN3,IN4 del puente h2	END	OUTPUT	5
7	Activa los pines IN1,IN2 del puente h3	ENE	OUTPUT	6
8	Activa los pines IN3,IN4 del puente h3	ENF	OUTPUT	7
9	Pulso entrante al sensor ultrasónico	echo	INPUT	8
10	Pulso saliente del sensor ultrasónico	trigger	OUTPUT	9
11	Envía dato al pin IN1 del puente h1	IN1	OUTPUT	10
12	Envía dato al pin IN2 del puente h1	IN2	OUTPUT	11
13	Envía dato al pin IN3 del puente h1	IN3	OUTPUT	12
14	Envía dato al pin IN4 del puente h1	IN4	OUTPUT	13
15	Activa modo manual o automático	pulsador_ma	INPUT	22
16	Envía dato al pin INN1 del puente h2	INN1	OUTPUT	24
17	Envía dato al pin INNN1 del puente h3	INNN1	OUTPUT	25
18	Envía dato al pin INN2 del puente h2	INN2	OUTPUT	26
19	Envía dato al pin INNN2 del puente h3	INNN2	OUTPUT	27
20	Envía dato al pin INN3 del puente h2	INN3	OUTPUT	28
21	Envía dato al pin INNN3 del puente h3	INNN3	OUTPUT	29
22	Envía dato al pin INN4 del puente h2	INN4	OUTPUT	30
23	Envía dato al pin INNN4 del puente h3	INNN4	OUTPUT	31
24	Indica modo manual activado	led_manual	OUTPUT	32
25	Indica modo automático activado	led_automático	OUTPUT	33
26	Indica obstáculo o paro	led_rojo	OUTPUT	40
27	Indica vehículo en movimiento	led_verde	OUTPUT	41
28	Permite cortar la corriente a motores	abortar	OUTPUT	42
29	Activa conexión con servomotor1	miServo1	OUTPUT	44
30	Activa conexión con servomotor2	miServo2	OUTPUT	45
31	Se activa cuando detecta obstáculo	buzzer	OUTPUT	51

Tabla 6: Resumen de pines de tarjeta Arduino

A continuación se procede a detallar la lógica de programación de nuestro proyecto utilizando el código Arduino.

Tipos de variables que lee el código Arduino en nuestro proyecto.

Tipo char:

Char val ;

Lee el valor de tipo caracter del puerto serial asignado a la variable “val”.

Tipo int:

int led_manual =32;

Lee valores enteros, asignando al pin 32 el valor de “1” como modo manual activado, y “0” como modo manual desactivado.

Tipo long:

long distancia ;

Al igual que int, también lee valores enteros pero largos, asignado a la variable distancia el valor que existe entre el vehículo y el obstáculo.

Asignación de variables a los pines del Arduino con puente h

PWM PUENTE H 1

int ENA = 2;

int ENB = 3;

int IN1 = 10;

int IN2 = 11;

int IN3 = 12;

int IN4 = 13;

La variable ENA asignado al pin 2 de la tarjeta Arduino, permite habilitar los pines IN1, IN2 del puente h1 para que pueda leer los datos enviados por los pines 10 y 11 del Arduino. De igual manera es para la variable ENB con los pines restantes.

Esta configuración es asignada para el puente h2 y puente h3. Ver (Anexo 9).

Configuración de puertos como entrada y salida. Estos datos son asignados en el “void setup()”.

```
pinMode(led_automtico,OUTPUT);  
pinMode(pulsador_ma,INPUT);
```

Para la configuración de entrada o salida se procede con la sintaxis “pinMode”, dentro del paréntesis se coloca la variable del pin Arduino seguido del estado que le queremos asignar.

Como ejemplo tenemos la variable “led_automtico”, donde le asignamos como pin de salida de datos, la misma que arrojará un voltaje haciendo que el led amarillo de encienda.

Luego tenemos la variable “pulsador_ma”, la cual le asignamos como entrada de datos, donde un valor externo ejecutado por un Switch, será leído por la tarjeta Arduino. Este proceso se aplica para todos los pines que utilizaremos en nuestra tarjeta mega. Ver (tabla de resumen de pines).

A continuación se realiza el proceso de manual o automático, su código se ejecuta en “void loop()”.

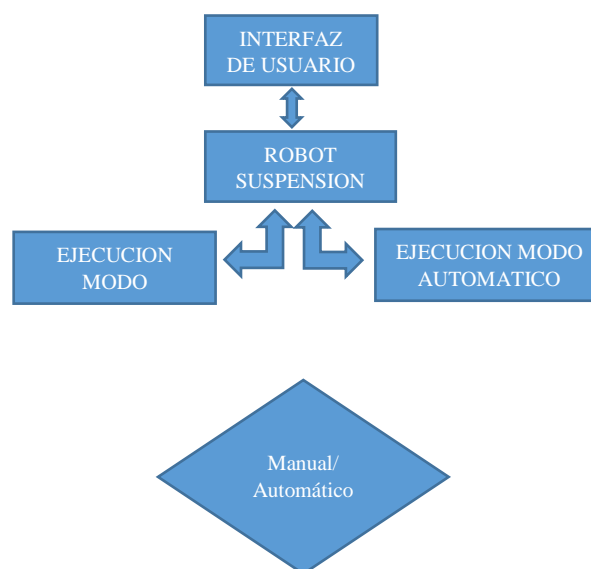


Figura 67: Proceso manual automático

Estructura “Llamada de función” en proceso manual.

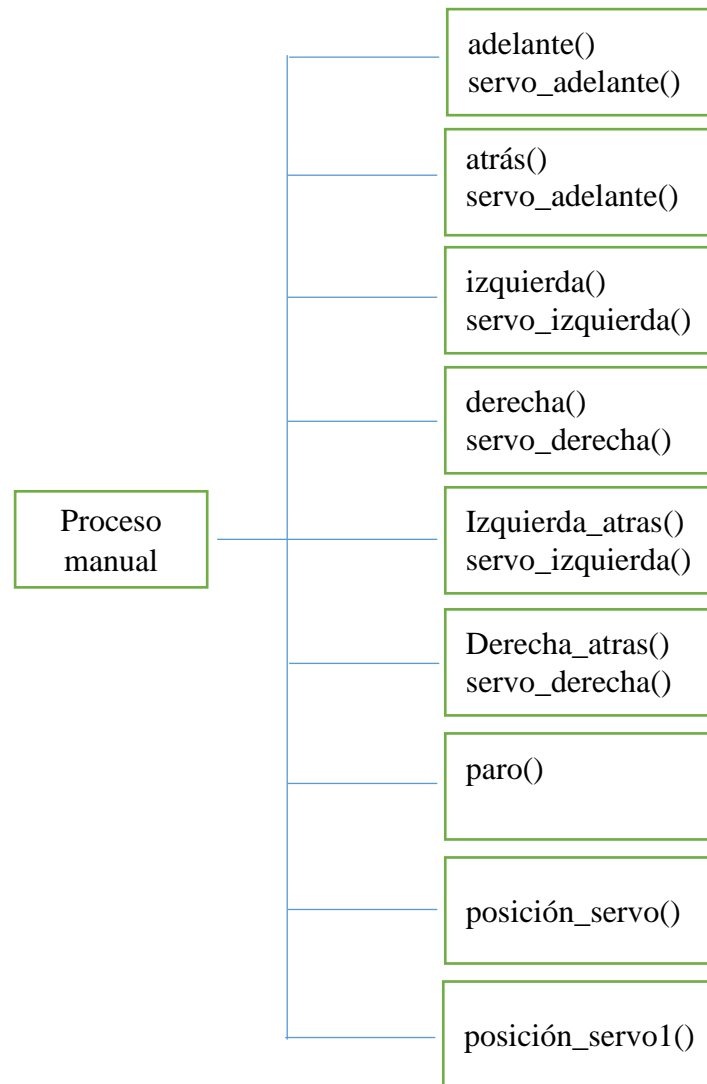


Figura 68: Llamada de funciones-manual

El proceso manual se encarga de llamar a cada una de las funciones para realizar los diferentes movimientos. Cada acción (adelante, atrás, izquierda, derecha), está acompañada de los movimientos de los servomotores (servo_adelante, servo_izquierda, servo_derecha), las cuales les proporcionará una dirección diferente. También consta de funciones para posicionar los servos a su estado inicial. Debido a que este proceso lo manejamos de forma manual, no es necesario tener activado el sensor, por esta razón no se manda a llamar a la función “ultrasónico”.

Proceso manual.

En esta sección se asignará una letra a cada acción. Esta letra es leída por el puerto serial para su posterior conexión con el programa visual studio.

Código	Descripción
<pre>if (digitalRead(pulsador_ma)==LOW) { digitalWrite(led_manual,HIGH); digitalWrite(led_automatgico,LOW); }</pre>	Si el pulsador está presionado, se activa el modo manual y se enciende el led verde
<pre>if (Serial.available(>0) { val=Serial.read(); }</pre>	Si el valor del puerto serial es mayor a cero, habilita la lectura del puerto serial asignándola a la variable "val".
<pre>if(val=='e') { servo_adelante(); adelante(); }</pre>	Si la tecla presionada en el teclado es la letra "e", el motor se mueve hacia adelante y el servo se ajusta a 85°.
<pre>if(val=='x') { servo_adelante(); atras(); }</pre>	Si la tecla presionada en el teclado es la letra "x", el motor se mueve hacia atrás y el servo se ajusta a 85°.
<pre>if(val=='s') { servo_izquierda(); izquierda(); }</pre>	Si la tecla presionada en el teclado es la letra "s", el motor se mueve hacia adelante y el servo se ajusta a 53°.
<pre>if(val=='d') { servo_derecha(); derecha(); }</pre>	Si la tecla presionada en el teclado es la letra "d", el motor se mueve hacia adelante y el servo se ajusta a 120°.
<pre>if(val=='z') { servo_izquierda(); izquierda_atras(); }</pre>	Si la tecla presionada en el teclado es la letra "z", el motor se mueve hacia atrás y el servo se ajusta a 53°.
<pre>if(val=='c') { servo_derecha(); derecha_atras(); }</pre>	Si la tecla presionada en el teclado es la letra "c", el motor se mueve hacia atrás y el servo se ajusta a 120°.
<pre>if(val=='a') { paro(); }</pre>	Si la tecla presionada en el teclado es la letra "a", el motor se para.
<pre>if(val=='o') { posicion_servo(); }</pre>	Si la tecla presionada en el teclado es la letra "o", el servo1 se posiciona a 89° y el servo2 se posiciona a 85°

Tabla 7: Proceso manual1

Código	Descripción
<pre>if(val=='p') { posicion_servo1(); }</pre>	Si la tecla presionada en el teclado es la letra “p”, el servo1 se posiciona a 83° y el servo2 se posiciona a 80°
<pre>if(val=='k') { digitalWrite(abortar,LOW); }</pre>	Si la tecla presionada en el teclado es la letra “k”, se desactiva el botón de abortar.
<pre>if(val=='l') { digitalWrite(abortar,HIGH); }</pre>	Si la tecla presionada en el teclado es la letra “l”, se activa el botón de abortar.
<pre>if(val=='g') { recorrido_automtico(); }</pre>	Si la tecla presionada en el teclado es la letra “g”, se activa el botón de recorrido automático.
<pre>if(val=='q') { reset(); }</pre>	Si la tecla presionada en el teclado es la letra “q”, se colocan en cero los valores de vel1, vel2, vel3
<pre>if(val=='y') { if (vel1 < 130) {vel1 = vel1+5; Serial.print(" ");Serial.println(vel1); } }</pre>	Si la tecla presionada en el teclado es la letra “y”, la velocidad del motor de enfrente aumenta
<pre>if(val=='u') { if (vel1 > -125) {vel1 = vel1-5; Serial.print(" ");Serial.println(vel1); } }</pre>	Si la tecla presionada en el teclado es la letra “u”, la velocidad del motor de enfrente disminuye
<pre>if(val=='h') { if (vel2 < 0) {vel2 = vel2+5; Serial.print(" ");Serial.println(vel2); } }</pre>	Si la tecla presionada en el teclado es la letra “h”, la velocidad del motor medio aumenta
<pre>if(val=='j') { if (vel2 > -255) {vel2 = vel2-5; Serial.print(" ");Serial.println(vel2); } }</pre>	Si la tecla presionada en el teclado es la letra “j”, la velocidad del motor medio disminuye.
<pre>if(val=='n') { if (vel3 < 130) {vel3 = vel3+5; Serial.print(" ");Serial.println(vel3); } }</pre>	Si la tecla presionada en el teclado es la letra “n”, la velocidad del motor de atrás aumenta.
<pre>if(val=='m') { if (vel3 > -125) {vel3 = vel3-5; Serial.print(" ");Serial.println(vel3); } }</pre>	Si la tecla presionada en el teclado es la letra “m”, la velocidad del motor de atrás disminuye.

Código	Descripción
<pre>if(val=='v') { digitalWrite(led_verde,LOW); }</pre>	Si la tecla presionada en el teclado es la letra “v”, se apaga el led frontal verde.
<pre>if(val=='b') { digitalWrite(led_verde,HIGH); }</pre>	Si la tecla presionada en el teclado es la letra “b”, se enciende el led frontal verde.

Tabla 8: Proceso manual2

Movimientos manuales

A continuación tenemos las acciones que se usarán para su desplazamiento manual.

Código	Descripción
<pre>void adelante() { //MOTOR 1-2-3-4-5-6 digitalWrite(IN1,HIGH); digitalWrite(IN2,LOW); digitalWrite(IN3,HIGH); digitalWrite(IN4,LOW); analogWrite(ENA,255+vel1); analogWrite(ENB,255+vel1); analogWrite(ENC,125+vel2); analogWrite(END,125+vel2); analogWrite(ENE,125+vel3); analogWrite(ENF,125+vel3); }</pre>	<p>Asignación de voltaje 1 a IN1-IN3, y voltaje 0 a IN2-IN4, para mover los motores hacia adelante. Esta configuración aplica para todos los motores.</p> <p>Los valores de ENABLE tienen asignados el valor pwm+vel que nos permite subir o bajar su valor.</p>
<pre>void atras() { //MOTOR 1-2-3-4-5-6 digitalWrite(IN1,LOW); digitalWrite(IN2,HIGH); digitalWrite(IN3,LOW); digitalWrite(IN4,HIGH); analogWrite(ENA,255+vel1); analogWrite(ENB,255+vel1); analogWrite(ENC,125+vel2); analogWrite(END,125+vel2); analogWrite(ENE,125+vel3); analogWrite(ENF,125+vel3); }</pre>	<p>Asignación de voltaje 0 a IN1-IN3, y voltaje 1 a IN2-IN4, para mover los motores hacia atrás. Esta configuración aplica para todos los motores.</p> <p>Los valores de ENABLE tienen asignados el valor pwm+vel que nos permite subir o bajar su valor.</p>
<pre>void izquierda() { //MOTOR 1-2-3-4 digitalWrite(IN1,HIGH); digitalWrite(IN2,LOW); digitalWrite(IN3,HIGH); digitalWrite(IN4,LOW); analogWrite(ENA,255+vel1); analogWrite(ENB,255+vel1); analogWrite(ENC, 50); analogWrite(END,125+vel2); analogWrite(ENE, 50); }</pre>	<p>Asignación de voltaje 1 a IN1-IN3, y voltaje 0 a IN2-IN4, para mover los motores hacia adelante. Esta configuración aplica para todos los motores.</p> <p>Los valores de ENABLE tienen asignados el valor pwm+vel que nos permite subir o bajar su valor.</p> <p>El direccionamiento lo dará el servo.</p>

<code>analogWrite(ENF,125+vel3); }</code>	
<pre>void derecha() { //MOTOR 1-2-3-4-5-6 digitalWrite(IN1,HIGH); digitalWrite(IN2,LOW); digitalWrite(IN3,HIGH); digitalWrite(IN4,LOW); analogWrite(ENA,255+vel1); analogWrite(ENB,255+vel1); analogWrite(ENC,250+vel2); analogWrite(END, 50); analogWrite(ENE,125+vel3); analogWrite(ENF, 50); }</pre>	<p>Asignación de voltaje 1 a IN1-IN3, y voltaje 0 a IN2-IN4, para mover los motores hacia adelante. Esta configuración aplica para todos los motores.</p> <p>Los valores de ENABLE tienen asignados el valor pwm+vel que nos permite subir o bajar su valor.</p> <p>El direccionamiento lo dará el servo.</p>
<pre>void paro() { analogWrite(ENA,00); analogWrite(ENB,00); analogWrite(ENC,00); analogWrite(END,00); analogWrite(ENE,00); analogWrite(ENF,00); }</pre>	<p>Para la opción “paro”, solo basta con deshabilitar los valores de ENABLE.</p>

Tabla 9: Movimientos Manuales

Estructura “Llamada de función” en proceso automático.

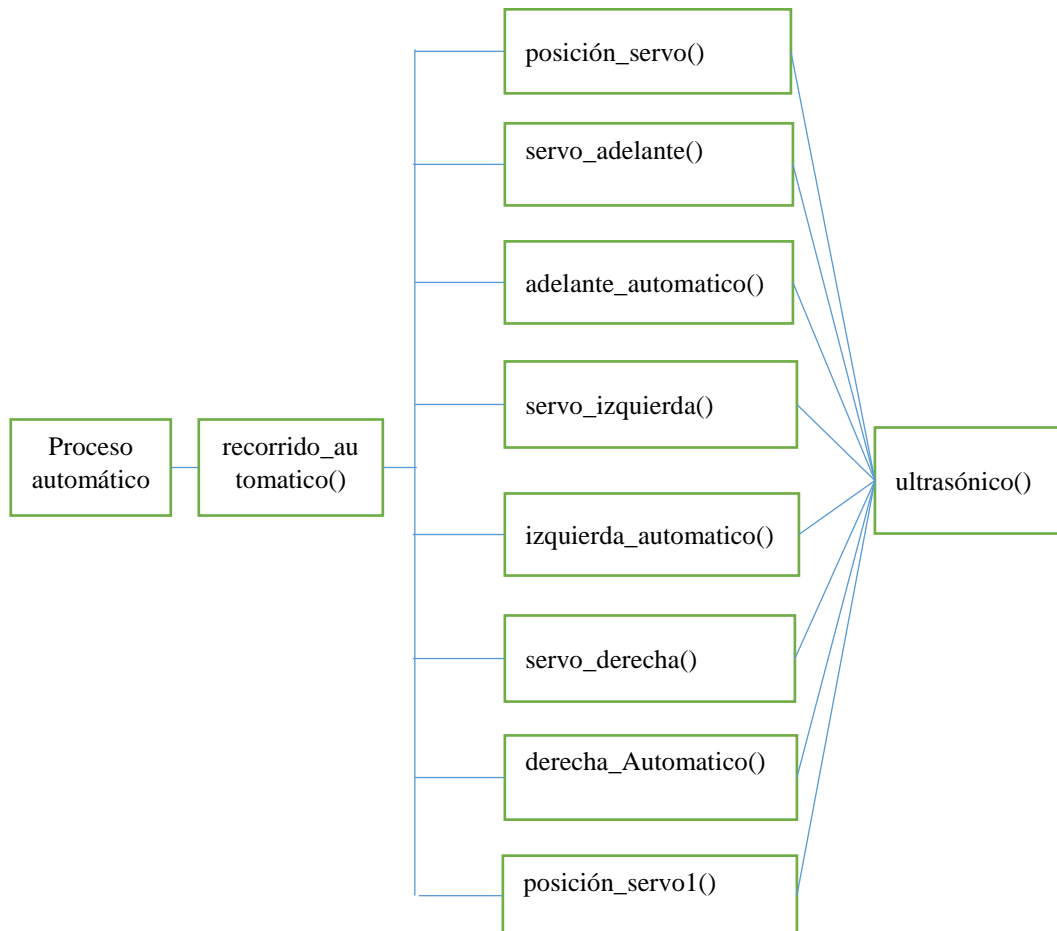


Figura 69: Llamado de funciones-automático

Para activar el proceso automático es necesario llamar a la función “recorrido_automático”, esta a su vez se encarga de llamar a las funciones (posición_servo, servo_adelante, adelante_automático, servo_izquierda, izquierda_automático, servo_derecha, derecha_automático, posición_servo1).

A diferencia del modo manual, esta tendrá que utilizar la función “ultrasónico” debido a que el prototipo debe movilizarse automáticamente y tiene que sensar algún obstáculo en frente.

Proceso automático

Al activar el modo automático, éste empezará a realizar los movimientos que se le programó.

Código	Descripción
<pre>if (digitalRead(pulsador_ma)==HIGH) { digitalWrite(led_manual,LOW); digitalWrite(led_automático,HIGH); }</pre>	Si el pulsador no está presionado, se activa el modo automático y se enciende el led amarillo
<pre>if ((distancia>60)or(distancia<1)) { recorrido_automático(); }</pre>	Si el sensor indica un valor mayor a 60 y menor a 1, entonces se activa el recorrido automático.

Tabla 10: Activación modo automático

Función “recorrido_automático()”

Código	Descripción
<pre>val=Serial.read();</pre>	Se asigna la variable “val” a la lectura de puerto serial.
<pre>for (int i=0; i<10; i++) { posicion_servo(); servo_adelante(); adelante_automático(); val=Serial.read(); if(val=='1') {digitalWrite(abortar,HIGH); } delay(100); }</pre>	El código se repite 10 veces con un lapso de tiempo de 0.1 seg. Haciendo que los motores se muevan hacia adelante, siempre que no exista la activación de la tecla “1” para cortar la energía a los motores.
<pre>for (int i=0; i<15; i++) { servo_izquierda();////////giro de servo izquierda_automático();//giro de motor val=Serial.read(); if(val=='1') {digitalWrite(abortar,HIGH); } }</pre>	El código se repite 15 veces con un lapso de tiempo de 0.1 seg. Haciendo que los motores se muevan hacia la izquierda, siempre que no exista la activación de la tecla “1” para cortar la energía a los motores.

<pre> delay(100); } </pre>	
<pre> for (int i=0; i<10; i++) { posicion_servo(); servo_adelante(); adelante_automtico(); val=Serial.read(); if(val=='l') {digitalWrite(abortar,HIGH); } delay(100); } </pre>	<p>El código se repite 10 veces con un lapso de tiempo de 0.1 seg. Haciendo que los motores se muevan hacia adelante, siempre que no exista la activación de la tecla “l” para cortar la energía a los motores.</p>

Tabla 11: Función recorrido automático

El recorrido puede variarse en tiempo y movimiento, dependerá de la configuración que se le asigne, si queremos que un movimiento dure más o menos, solo debemos cambiar la cantidad de veces que se repite el “for”, por otro lado podemos incrementar los movimiento escribiendo en el código, más funciones.

Movimientos automáticos

A continuación tenemos las acciones que se usaran para su desplazamiento automático.

Código	Descripción
<pre> void adelante_automtico() { Serial.begin(9600); ultrasonico(); //MOTOR 1-2-3-4-5-6 digitalWrite(IN1,HIGH); digitalWrite(IN2,LOW); digitalWrite(IN3,HIGH); digitalWrite(IN4,LOW); if((distancia>60)or(distancia<1)) { digitalWrite(led_verde,HIGH); delay(200); digitalWrite(led_verde,LOW); delay(200); analogWrite(ENA-B-C-D-E-F,255+vel); if(val=='k') {digitalWrite(abortar,LOW); } } else {digitalWrite(buzzer,HIGH); digitalWrite(led_rojo,HIGH); delay(50); digitalWrite(buzzer,LOW); digitalWrite(led_rojo,LOW); } </pre>	<p>Asignación de voltaje 1 a IN1-IN3, y voltaje 0 a IN2-IN4, para mover los motores hacia adelante. Esta configuración aplica para todos los motores.</p> <p>Los valores de ENABLE tienen asignados el valor pwm+vel que nos permite subir o bajar su valor.</p> <p>El sensor de distancia actua constantemente.</p> <p>Si la distancia registrada por el sensor es mayor a 60 cm, entonces se desplaza normalmente y se enciende los led verdes frontales.</p> <p>Caso contrario se detendrá, se prenderá el buzzer y el led rojo frontal.</p> <p>El direccionamiento lo dará el servo.</p>

<pre>delay(50); analogWrite(ENA-B-C-D-E-F,000); Serial.println('a'); }}</pre>	
<pre>void izquierda_automatico() { analogWrite(ENA,255+vel1); analogWrite(ENB,255+vel1); analogWrite(ENC, 50); analogWrite(END,125+vel2); analogWrite(ENE, 50); analogWrite(ENF,125+vel3); }</pre>	<p>Su configuración es igual al de la función adelante_automatico(), con excepción del analogWrite(). Esto hace que ayude a girar hacia la izquierda.</p>
<pre>void derecha_automatico() { analogWrite(ENA,255+vel1); analogWrite(ENB,255+vel1); analogWrite(ENC,125+vel2); analogWrite(END, 50); analogWrite(ENE,125+vel3); analogWrite(ENF, 50); }</pre>	<p>Su configuración es idéntica al de la función adelante_automatico(), con excepción del analogWrite(). Esto hace que ayude a girar hacia la derecha.</p>
<pre>void paro_automatico() { analogWrite(ENA-B-C-D-E-F,00); Serial.println('a'); }</pre>	<p>Queda desahabilitado si el ENABLE es igual a cero</p>
<pre>void atras_automatico() { Serial.begin(9600); ultrasonico(); //MOTOR 1-2-3-4-5-6 digitalWrite(IN1,LOW); digitalWrite(IN2,HIGH); digitalWrite(IN3,LOW); digitalWrite(IN4,HIGH); analogWrite(ENA-B-C-D-E-F,255+vel); }</pre>	<p>Asignación de voltaje 0 a IN1-IN3, y voltaje 1 a IN2-IN4, para mover los motores hacia adelante. Esta configuración aplica para todos los motores. Los valores de ENABLE tienen asignados el valor pwm+vel que nos permite subir o bajar su valor.</p>
<pre>void ultrasonico() { digitalWrite(9,LOW); delayMicroseconds(2); digitalWrite(9, HIGH); delayMicroseconds(10); digitalWrite(9,LOW); tiempo=pulseIn(8, HIGH); distancia= int(0.017*tiempo); //Serial.println(distancia); //delay(50); }</pre>	<p>Se manda un dato desde el pin 9 hasta el pin 8, para calcular la distancia.</p>

Tabla 12: Movimientos automáticos

Los posicionamientos de los servos se los realiza cuando se hace un giro tanto izquierdo como derecho, y al regresar no llega a su punto de inicio. Por tanto esto nos ayuda a re-direccionarlo nuevamente.

Código	Descripción
<pre>void posicion_servo() { ultrasonico(); miServo1.write(89); miServo2.write(85); }</pre>	Se establece el ángulo de los servos1 y 2, mientras trabaja el sensor ultrasónico. Su función es calibrar el ángulo después de girar a la derecha.
<pre>void posicion_servo1() { ultrasonico(); miServo1.write(83); miServo2.write(80); }</pre>	Se establece el ángulo de los servos1 y 2, mientras trabaja el sensor ultrasónico. Su función es calibrar el ángulo después de girar a la izquierda.

Tabla 13: Función posición servo

En este código observamos los ángulos que toman los servos al momento de dar un giro, su posicionamiento hacia el frente es de 85°, hacia la derecha es de 53°, y hacia la izquierda es de 120°.

Código	Descripción.
<pre>void servo_adelante() { ultrasonico(); miServo1.write(85); //izq miServo2.write(85); //der }</pre>	Los servos se posicionan hacia adelante, mientras trabaja el sensor ultrasónico.
<pre>void servo_izquierda() { ultrasonico(); miServo1.write(53); miServo2.write(53); delay(80); }</pre>	Los servos se posicionan hacia adelante, mientras trabaja el sensor ultrasónico.
<pre>void servo_derecha() { ultrasonico(); miServo1.write(120); miServo2.write(120); delay(80); }</pre>	Los servos se posicionan hacia adelante, mientras trabaja el sensor ultrasónico.

Tabla 14: Funciones Servos

Diagrama del proceso del prototipo.

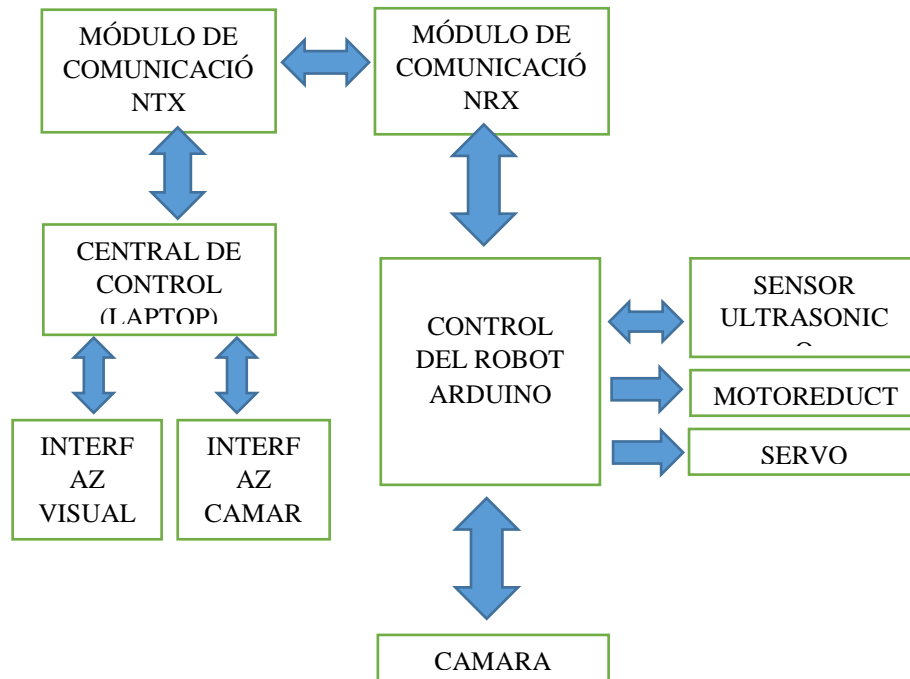


Figura 70: Diagrama del prototipo.

2.6 ESTUDIO DE FACTIBILIDAD. PRESUPUESTO

Para realizar nuestro proyecto era indispensable el uso de algunos recursos adicionales mostrados en las siguientes tablas.

DETALLES DE SOFTWARE	CANT.	PRECIO	TOTAL
Sistema Operativo Windows 8	1	\$100.00	\$100.00
Microsoft office 2013 Professional	1	\$50.00	\$50.00
Software Arduino	1	\$0.00	\$0.00
Software Visual Studio	1	\$0.00	\$0.00
Software Configuración X.CTU	1	\$0.00	\$0.00
Software Camera (pc-móvil)	1	\$0.00	\$0.00
Total en software			\$150.00

Tabla 15: Detalle de costo de software

DETALLES HARDWARE	CANT.	PRECIO	TOTAL
Pc Portátil	1	\$500.00	\$500.00
Total en software			\$500.00

Tabla 16: Detalle de costo de pc.

A continuación se detalla el valor de cada componente usado en la elaboración del proyecto. Algunos componentes vienen incluido el valor de envío a provincias (costo adicional por envío).

DETALLES	CANT.	PRECIO	TOTAL
TARJETA REGULADORA DE VOLTAJE			
Baquelita sin perforar 10x20 cm	1	1.70	1.70
Lm7805 metálico	1	0.54	0.54
Lm7806 metálico	1	0.54	0.54
Lm7809 metálico	1	0.54	0.54
Lm7812 metálico	1	0.54	0.54
Diodo rectificador	4	0.09	0.36
Capacitor electrolítico 25v 2200uf	4	0.45	1.80
Capacitor electrolítico 25v 10uf	4	0.14	0.56
Capacitor cerámico	4	0.18	0.72
Diodo led	4	0.18	0.72
Resistencia	4	0.10	0.40
Borneras	4	0.10	0.40
Plug dc con borneras	1	0.98	0.98
Disipador	1	1.00	1.00
TARJETA DE CONTROL			
Switch 3 posiciones	1	0.63	0.63
Plug dc con borneras hembra	1	0.98	0.98
Switch	1	0.63	0.63
Arduino mega	1	36.00	36.00
Diodo led	2	0.18	0.36
pulsador	1	0.50	0.50
cables	3	0.50	1.50
TARJETAS DE POTENCIA Y MOTORES			
Puente H	3	13.00	13.00
Moto reductores Pololu 12v 100:1	4		118.40
Moto reductores Pololu 12v 50:1	2		65.20
Servo motores	4	15.00	60.00
CHASIS Y SOPORTES			
Varilla roscable 5/16"x 1 m	1	1.79	1.79
Anillo plano galv sencillo 5/16"	6	0.04	0.24
Tuercas 5/16"	9	0.03	0.27
Tablero prensado blanco 1x2m	1	5.00	5.00
Barra de aluminio 1x1/2	1	4.00	4.00
Barra de aluminio 1x1	1	2.00	2.00
Barra de aluminio cuadrado 2cm	1	7.00	7.00
Pernos 5cm	2	0.80	0.80
Remaches x docena 0.3 mm	8	0,17	1.36
Tubos PVC	6	1.00	6.00
Tapón tubo PVC	6	0.50	3.00
Llanta de caucho bicicleta	1	8.00	8.00
OTROS			

Chicharra 3v-24vmini-buzzer	1	0.89	0.89
Ventilador 12v pequeño	1	2.86	2.86
Metro de solda	1	0.54	0.54
Metro de cable utp cat 5e	1	0.54	0.54
CARGA-ALIMENTACION			
Batería seca 12 v 5 amp	1	23.00	23.00
Terminales de conexión	2	0.10	0.20
Cámara inalámbrica	1	70.00	70.00
TARJETAS INALÁMBRICAS			
Xbee	2	40.5	81
Adaptador xbee	2	15.00	30.00
Cables USB para antena	1	5.00	5.00
Cables Arduino	37	0.16	5.92
			565.80

Tabla 17: Valor de cada componente usado en la elaboración del proyecto.

El costo total del proyecto es de \$1215.80

2.7 RESULTADOS.

El prototipo estuvo expuesto a pruebas rigurosas en terrenos de difícil manejo, con el objetivo de dar a conocer sus límites más allá de los terrenos de cultivos. Con esto se pudo calcular los siguientes datos con respecto a ciertos componentes.

Potencia de los motores en terreno liso (desplazamiento sin obstáculo).

$$I = \frac{P}{V}$$

		CORRIENTE	VOLTAJE	POTENCIA
MOTORES 100:1	FREE RUN	130 mA	12 V	1.56 W
	CON PESO	270 mA	12 V	3.24 W
MOTORES 50:1	FREE RUN	150 mA	12 V	1.80 W
	CON PESO	180 mA	12V	2.16 W

Tabla 18: Potencia de los motores zona libre.

Potencia de los motores en pendiente con ángulos de 30° (°).

		CORRIENTE	VOLTAJE	POTENCIA
MOTORES 100:1				
	SIN PESO	370 mA	12 V	4.44 W
MOTORES 50:1				
	SIN PESO	590 mA	12V	7.08 W

Tabla 19: Potencia de los motores zona con pendiente Ángulo 30.

Peso adicional corresponde a 8 libras del fumigador (partes necesarias)

Recorrido en un campo de cultivos.

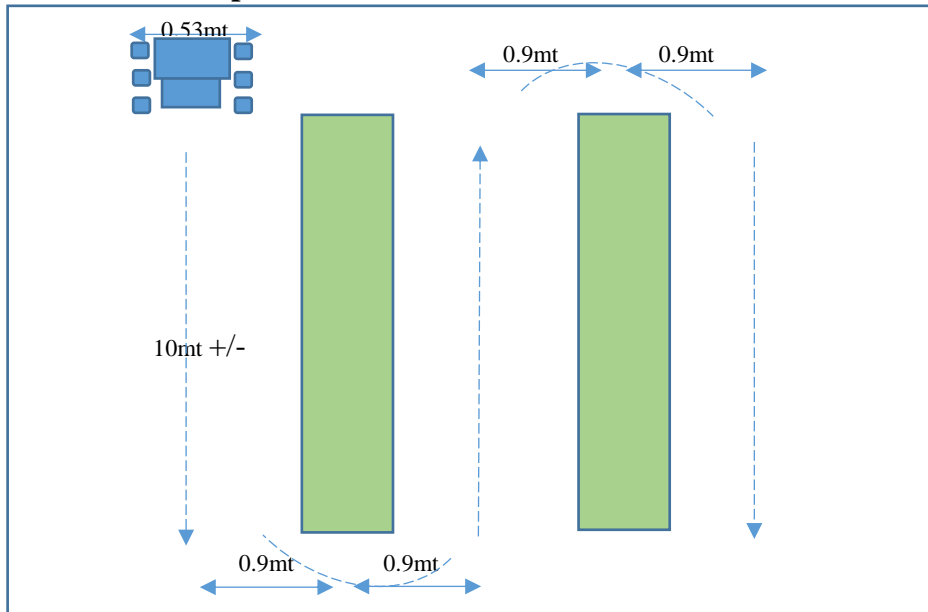


Figura 71: Recorrido en campo de cultivo.

En el modo manual su trayectoria se cumple sobrepasando los obstáculos realizando los giros programados con un ángulo de 30° en los servos, haciendo un desplazamiento semicircular con diámetro de 180 cm.

En el modo automático existe la particularidad, que al sobrepasar un obstáculo, éste puede desviar su trayectoria.

Estabilidad

La estabilidad del prototipo trabaja en un 95%, esto quiere decir que no se balanceaba salvo por el terreno desnivelado.

Se realizó pruebas en pendientes difíciles de 40° , logrando bajar con éxito, mientras que en la subida se dificultaba en terrenos con mucha tierra.



Figura 72: Pruebas en terrenos

PRUEBAS MANUAL

N°	Detalle de la prueba	Tipo de suelo	cantidad	Exitoso	
1	Desplazamiento Libre	Cemento	1	Si	100%
			2	Si	100%
			3	Si	100%
			4	Si	100%
			5	Si	100%
2	Desplazamiento Libre	Tierra	1	Si	100%
			2	Si	100%
			3	Si	100%
			4	Si	100%
			5	No	50%
3	Subida pendiente	Tierra	1	Si	100%
			2	Si	100%
			3	Si	100%
			4	No	30%
			5	No	30%
4	Bajada pendiente	Tierra	1	Si	100%
			2	Si	100%
			3	Si	100%
			4	Si	100%
			5	Si	100%
	porcentaje				86%

Tabla 20: Prueba Manual.

PRUEBAS AUTOMÁTICO

N°	Detalle de la prueba	Tipo de suelo	cantidad	Exitoso	
1	Desplazamiento	Cemento	1	Si	100%
			2	Si	100%
			3	Si	100%
			4	Si	100%
			5	Si	100%
2	Desplazamiento	Tierra	1	Si	100%
			2	Si	100%
			3	Si	100%
			4	No	50%
			5	No	50%
	porcentaje				90%

Tabla 21: Prueba Automático.

Porcentaje general = 88%

Tomando en cuenta que se hicieron pruebas fuera de los límites de un cultivo podemos decir que su funcionamiento es accesible para los campos.

Las pruebas realizadas en los terrenos fueron analizadas sin el peso correspondiente a los componentes del robot fumigador (8 libras = 3.6 kg), este adicional aumentaría el consumo de energía, haciendo que la batería se consuma más rápido y sus movimientos sean más lentos.



Figura 73: Unión de las partes de fumigación y suspensión

Las partes que se consideran para adaptar al robot de suspensión rocker bogie sería el de proceso de fumigación evitando algunas partes como el de movilidad o cámara.

Consumo de corriente.

Se hicieron pruebas donde se le adicionó una carga equivalente 4.5 libras

		CORRIENTE	VOLTAJE	POTENCIA
Robot móvil	Sin carga	2.30A	12 V	27.60 W
Robot móvil	Con carga	2.84A	12 V	34.08 W

Tabla 22: Consumo de corriente.

CONCLUSIONES

El análisis de los 3 tipos de suspensión, mostraron al modelo rocker-bogie como el más opcional, debido a que soporta mayor peso sin afectar su desplazamiento, permite el movimiento individual de cada soporte, tiene mejor acoplamiento al suelo, y logra que la base se equilibre sin importar el tamaño.

El mecanismo de suspensión tuvo dos etapas de alineamiento, la primera presentaba un leve desvío de 5° hacia la izquierda, en la segunda etapa, al agregar soporte a los lados se pudo corregir en un 95%, haciendo más estable su direccionamiento. La suspensión puede soportar una carga adicional de 6 libras (2.7kg), pasado de eso los soportes de las ruedas comienzan a desnivelarse. Esto se debe a que el soporte central es delgado y solo permite una carga aproximada de 19 libras (8.6 kg).

El sistema subamortiguado, que representa la base y el soporte respecto a su punto de equilibrio, muestra un 90% de estabilidad, muy bueno en referencia a su empleo en terrenos desnivelados. Sin embargo sus ruedas se encuentran vulnerables, al no amortiguar bajadas mayores a 10cm, esto puede causar desvíos en la posición de los servos.

La potencia de los motores permite desplazarse en terrenos con pendientes de 30° en subida y 40° de bajada. El consumo de energía de estos motores (1,5 a 2 A) la mitad de su consumo general, es más eficaz que otros tipos de motores. La potencia de 2mW de la antena permite una comunicación constante entre pc y robot móvil, trabajando sin problema a una distancia de 45 a 50m con línea de vista. Este dato es corroborado por sus especificaciones técnicas.

El prototipo puede ejecutarse mediante el monitor serial del software Arduino, sin embargo, se diseñó una interfaz gráfica en el software Visual Studio, todo con el fin de que el usuario puede manejar con mayor facilidad el prototipo.

El móvil deja de avanzar al momento en que el ultrasónico detecta un obstáculo que se encuentra dentro del área de censando, a una distancia de 60cm como mínimo para evitar choque. Pueden pasar obstáculos menores a 4 cm de altura, pasado de esto hay dificultades en las ruedas centrales, debido al peso que se centra por la base.

Los servos permiten un desplazamiento semicircular de 0.90m de radio, al momento de realizar un giro, debido a que solo tiene 2 servos y las otras ruedas no permiten un giro menor. El torque de 15 kg de los servos, es suficiente para mover el robot que oscila los 10 kg y su consumo de energía es de 100mA.

RECOMENDACIONES

- Es preciso mencionar que los motores no están protegidos en su totalidad y pueden quedar expuesto a daños en sus conexiones.
- No hay que exponer al robot a muchos golpes debido a que los servomotores pueden sufrir daños y variar sus grados de movilidad, haciendo que tomen direcciones distintas a las configuradas.
- Las baterías no deben pasar sus límites de descarga, debido a que la potencia disminuiría considerablemente.
- La carga adicional (8 libras correspondientes al robot fumigador), debe ser colocada de manera que proporcione equilibrio y no afecte su movilidad.
- No exagerar el peso de la carga adicional porque el soporte comenzaría a doblarse.
- Debemos estar pendiente en el sistema de carga, y desconectar la batería porque puede disminuir su tiempo de vida.
- Para hacer cambios en la configuración del código, hay que desconectar la comunicación de la antena inalámbrica, porque sería imposible cargarla a la tarjeta.
- Verificar el estado del sensor, que funcione correctamente, porque podría incidir en su recorrido.
- Es importante que las antenas transmisoras y receptoras no sufran ningún golpe, esto haría una mala comunicación entre ambos terminales.

BIBLIOGRAFÍA

- [1] E. W. A. C. Swokowski, Álgebra y Trigonometría con geometría analítica, Décimo tercera edición, Mexico: Brooks/Cole, Cengage Learning, 2011.
- [2] V. M. G. Cabrera, FÍSICA FUNDAMENTAL, Mexico, D.F.: EDITORIAL PROGRESO, S.A. DE C.V., 1971.
- [3] Colegio de ingenieros técnicos de obras públicas, Elementos de ingeniería romana, Tarragona: Isaac Moreno Gallo, 2004.
- [4] C. d. L. Hernández, Física, Mexico, D.F.: EDITORIAL PROGRESO, S.A. DE C.V., 1994.
- [5] P. L. R.-. D. Á. Mantaras, Investigación de accidentes de tráfico, Oviedo: Universidad de Oviedo, Servicio de Publicaciones, 2003.
- [6] F. R. Díaz, CONTROL Y ROBÓTICA EN AGRICULTURA, Almería: Universidad de Almería. Servicio de Publicaciones, 2004.
- [7] P. G. Bejerano, «Blogthinkbig.com,» 29 Julio 2013. [En línea]. Available: <http://blogthinkbig.com/robots-agricolas-revolucion-agricultura/>.
- [8] A. Cruz, «Electronilab Ingeniería y Diseño Electrónico,» 07 Mayo 2014. [En línea]. Available: <http://electronilab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/>.
- [9] Y. Cruz, «La génesis de la tecnología,» 23 Marzo 2015. [En línea]. Available: <http://newcreacion.blogspot.com/2015/03/la-robotica-en-la-agricultura-y-la.html>.
- [10] DFROBOT, «DFROBOT DRIVE THE FUTURE,» [En línea]. Available: http://www.dfrobot.com/index.php?route=product/product&product_id=72#.V3wYptzufDc.
- [11] N. RUEDA, «ENTER.CO,» 26 Julio 2013. [En línea]. Available: <http://www.enter.co/cultura-digital/secretos-tecnologia/estos-robots-podrian-revolucionar-la-agricultura/>.
- [12] C. M. Rojas, «GeekyTheory,» 05 Abril 2013. [En línea]. Available: <https://geekytheory.com/tutorial-arduino-comenzando-con-xbee/>.
- [13] J. NICOLAU, «EL MUNDO,» 17 Enero 2015. [En línea]. Available: <http://www.elmundo.es/comunidad-valenciana/2015/01/17/54ba3d4a22601dc36a8b456c.html>.

- [14] POLOLU ROBOTIC & ELECTRONIC, «POLOLU ROBOTIC & ELECTRONIC,» [En línea]. Available: <https://www.pololu.com/product/1104>.
- [15] DILIUM, «PILAS Y BATERIAS,» 23 Junio 2014. [En línea]. Available: <https://pilasybaterias.wordpress.com/2014/06/23/introduccion/>.
- [16] Geek Factory, «Geek Factory,» [En línea]. Available: <http://www.geekfactory.mx/tienda/sensores/hc-sr04-sensor-de-distancia-ultrasonico/>.
- [17] Geek Factory, «Geek Factory,» [En línea]. Available: <http://www.geekfactory.mx/tienda/motores-y-controladores/mg995-tower-pro-servo-motor-metalico/>.
- [18] Diario ABC, «La NASA muestra en directo como construye el rover que enviará a Marte,» *Ciencia*, p. 19, 01 Agosto 2012.
- [19] U. F. M. R. / O. D. S. B, «MANUAL DE USUARIO DE LA TARGETA ARDUINO MEGA,» 22 Febrero 2011. [En línea]. Available: <http://tecnoarduinamegaurdistrical.blogspot.com/>.
- [20] J. Figueroa Sarcos, "Prototipo de robot semiautónomo especializado en fumigación agrícola bajo la tecnología Arduino.", *La Libertad; Universidad Estatal Península de Santa Elena, 2015.*, 2015. <http://repositorio.upse.edu.ec/handle/46000/2438>

ANEXOS

ANEXO 1

1 Descripción de componentes físicos.

MOTO REDUCTORES 50:1 (METAL GEARMOTOR)-37DX54L

“This gearmotor is a powerful brushed DC motor with 50:1 metal gearbox intended for operation at 12 V. These units have a 0.61"-long, 6 mm-diameter D-shaped output shaft. This gearmotor is also available with an integrated encoder.” [14]

“These motors are intended for use at 12 V, though in general, these kinds of motors



can run at voltages above and below the nominal voltage (they can begin rotating at voltages as low as 1 V). Lower voltages might not be practical, and higher voltages could start negatively affecting the life of the motor.” [14]

Figura 74: Motoreductor 50:1 (Metal Gearmotor)-37Dx54L: Pololu

SERVOMOTORES.

“Dentro de los servos económicos el servo MG995 Tower Pro destaca por su gran



torque, engranajes metálicos y gran robustez. Funciona con la mayoría de tarjetas electrónicas con microcontroladores y además con la mayoría de los sistemas de radio control comerciales. Es utilizado principalmente en proyectos de robótica y modelismo de mediano tamaño.” [17]

Figura 75: Servo MG995 Tower Pro.

Características:

- Interfaz: analógica
- Dimensiones: 40.6 longitud x 19.8 anchura x 42.9 alto mm.
- Peso: 55 g.
- Torque: 10 kg/cm
- Voltaje: 4 a 7 voltios.

PUENTE H

“Este módulo basado en el chip L298N te permite controlar dos motores de corriente continua o un motor paso a paso bipolar de hasta 2 amperios.” [8]

“El módulo cuenta con todos los componentes necesarios para funcionar sin necesidad de elementos adicionales, entre ellos diodos de protección y un regulador LM7805 que suministra 5V a la parte lógica del integrado L298N. Cuenta con jumpers de selección para habilitar cada una de las salidas del módulo (A y B). La salida A esta conformada por OUT1 y OUT2 y la salida B por OUT3 y OUT4. Los pines de habilitación son ENA y ENB respectivamente.” [8]

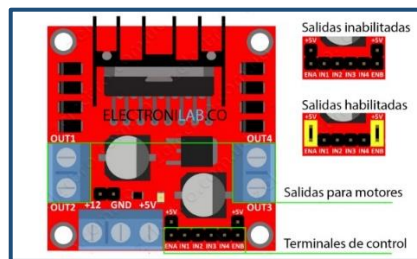


Figura 76: Conexión para la alimentación.

“Este módulo se puede alimentar de 2 maneras gracias al regulador integrado LM7805.” [8]

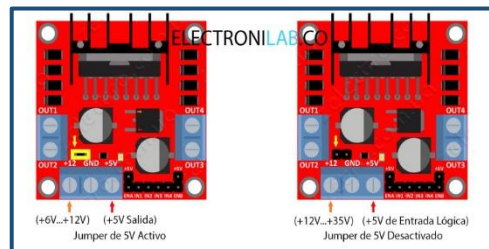


Figura 77: Conexión 12v en puente H.

“Cuando el jumper de selección de 5V se encuentra activo, el módulo permite una alimentación de entre 6V a 12V DC. Como el regulador se encuentra activo, el pin marcado como +5V tendrá un voltaje de 5V DC. Este voltaje se puede usar para alimentar la parte de control del módulo ya sea un microcontrolador o un Arduino, pero recomendamos que el consumo no sea mayor a 500 mA.” [8]

“Cuando el jumper de selección de 5V se encuentra inactivo, el módulo permite una alimentación de entre 12V a 35V DC. Como el regulador no esta funcionando, tendremos que conectar el pin de +5V a una tensión de 5V para alimentar la parte

lógica del L298N. Usualmente esta tensión es la misma de la parte de control, ya sea un microcontrolador o Arduino.” [8]

SENSOR HC-SR04

“El sensor HC-SR04 es una excelente opción como sensor de distancia ultrasónico. Su relación de costo/beneficio lo hace óptimo para un gran abanico de aplicaciones. El uso de este módulo es bastante sencillo debido a que se toda la electrónica de control, transmisión y recepción se encuentra contenida en PCB. El usuario solamente debe enviar un pulso de disparo y medir en tiempo alto del pulso de respuesta. Solamente se requieren 4 hilos para completar la interfaz con el módulo de sensor HC-SR04.” [16]

Características del Sensor:

- 5 voltios de alimentación.
- Interfaz: Vcc, Trigger, Echo y GND
- Rango de medición: 400 cm máximo.
- 15 mA en corriente de alimentación.
- Frecuencia: 40 Khz.
- Señal disparo: 10microSegundo.
- Tamaño: 45x20x15 (mm).

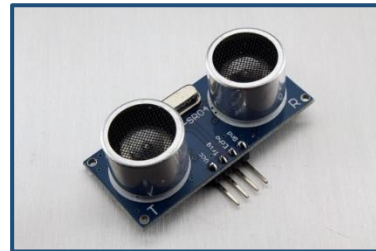


Figura 78: Módulo de sensor HC-SR04.

Microcontrolador ATmega2560

“La tarjeta Arduino Mega 2560 es una tarjeta basada en el microcontrolador ATmega2560. Cuenta con 54 pines digitales de entrada/salida (de los cuales 14 pueden ser utilizados como salidas PWM), 16 entradas analógicas, 4 UARTs (hardware serial ports), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador, simplemente se conecta a un ordenador con un cable USB, a un adaptador de CA a CC, o de la batería para iniciar. La tarjeta Arduino Mega es compatible con la mayoría de shields diseñados para Arduino Duemilanove o Diecimila.” [19]

Características principales:

- Voltaje: 7-12V.
- Estabilización de 5Vcc.
- 54 líneas de Entradas/Salidas Digitales
- 14 pines se pueden utilizar como salidas PWM
- Entradas Analógicas 16
- Máxima corriente: 40 mA.
- Voltaje de salida 3.3V con 50 mA.
- Memoria de 256Kb
- Memoria SRAM de 8Kb.
- Memoria EEPROM para datos.
- Velocidad del reloj:16MHz
- Tamaño de 100 x 50 mm

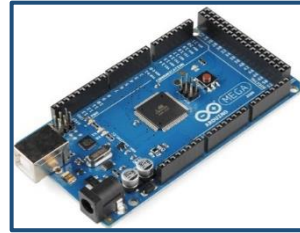


Figura 79: Microprocesador ATmega2560

Módulo xbee s2.

“Los Xbee son dispositivos inalámbricos fabricados por Digi International, tienen su propio protocolo de comunicación por radio frecuencia (en banda de radio libre), son robustos, de bajo costo, bajo consumo y tienen un alcance en sus distintos modelos entre 100 metros y los 10 kilómetros ! . Otra cosa que los hace interesantes es que los dispositivos de la serie 2 pueden hacer redes mesh que según el proyecto o el espacio físico puede ser de gran ayuda.” [12]

Características:

- Utiliza 3,3 V a 40 mA.
- Máximo bit rate 250 kbps.
- Potencia 2mW output.
- 120 m rango, el S2.
- Antena externa.
- 8 pines digitales IO.
- 128-bit de encriptación.
- Set de commando AT o API.



Figura 80: Módulo xbee s2.

Adaptador Universal XBee USB

“The upgraded DFRobot Xbee USB adapter is designed for the XBee module configuration parameters to facilitate using or to work as a FTDI writer. The new version of Xbee USB adapter is also welded with the female pin for FTDI so that users can directly use the adapter as a FTDI program writer, such as Arduino FIO/pro/mini, lilypad Arduino and so on. It's a must-have for interactive electronic installations and projects.” [10]

Especificaciones:

- Compatible con 3.3V y 5V.
- USB 2.0.
- Tiene conector mini USB.
- Soporta el software X-CTU.
- No necesita energía externa.
- USB 1.0 y USB 2.0
- 256/128 bytes de RX TX.
- Regulador de 3.3V a 250mA.
- Leds de comunicación TX y RX.
- Soporte para Windows7, 8 y 8.1.

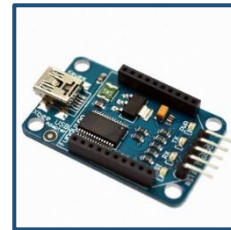


Figura 81: Adaptador Universal XBee USB.

BATERÍA

“Una pila o batería eléctrica es el formato industrializado y comercial de la celda galvánica o voltaica. Es un dispositivo que convierte energía química en energía eléctrica por un proceso químico transitorio, tras lo cual cesa su actividad y han de renovarse sus elementos constituyentes, puesto que sus características resultan alteradas durante el mismo. Se trata de un generador primario. Esta energía resulta accesible mediante dos terminales que tiene la pila, llamados polos, electrodos o bornes. Uno de ellos es el polo negativo o ánodo y el otro es el polo positivo o cátodo.” [15]



Figura 82: Batería seca 12v a 5 Amperios.

ANEXO 2

2Data sheet Puente H L298



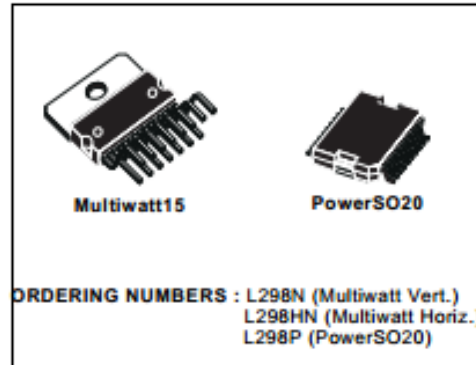
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM

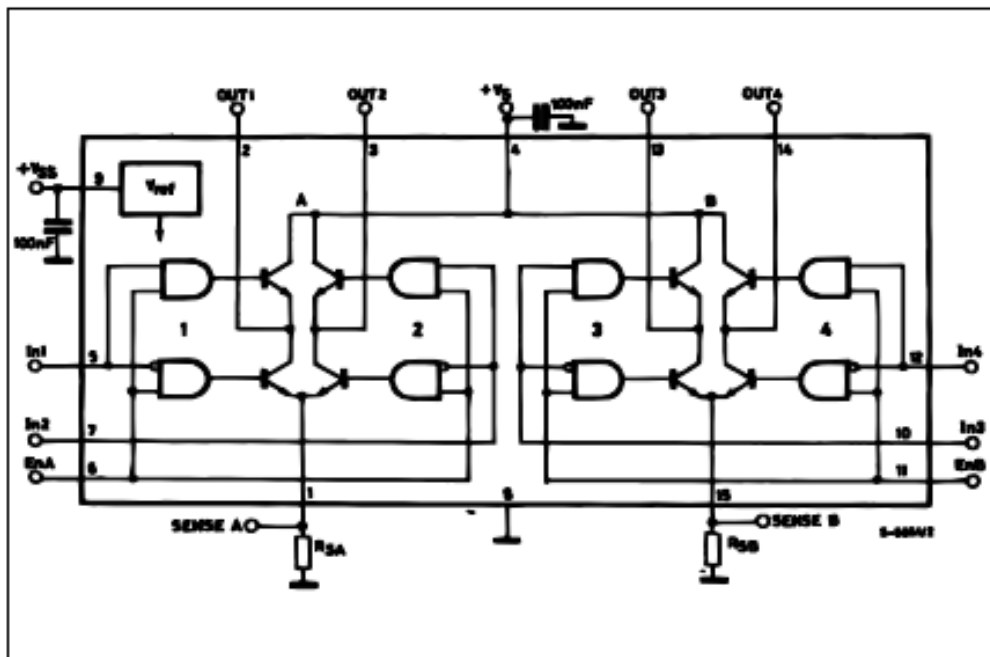
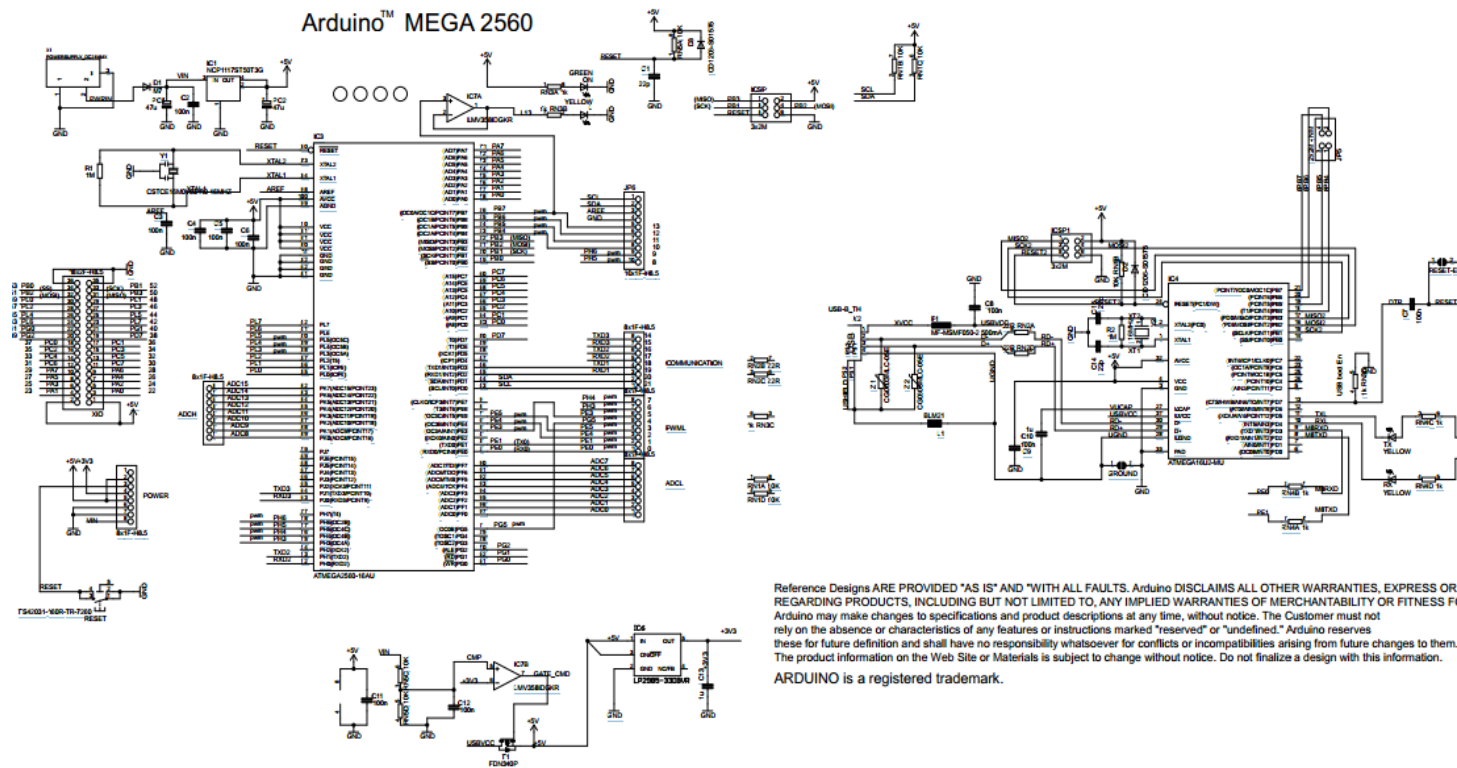


Figura 83: Data sheet Puente H L298:

ANEXO 3

3 Configuración de Pines Arduino Mega.



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information. ARDUINO is a registered trademark.

Figura 84: Data sheet Arduino mega

ANEXO 4

4Regulador de Voltaje LM78XX



LM78XX / LM78XXA 3-Terminal 1 A Positive Voltage Regulator

Features

- Output Current up to 1 A
- Output Voltages: 5, 6, 8, 9, 10, 12, 15, 18, 24 V
- Thermal Overload Protection
- Short-Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The LM78XX series of three-terminal positive regulators is available in the TO-220 package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut-down, and safe operating area protection. If adequate heat sinking is provided, they can deliver over 1 A output current. Although designed primarily as fixed-voltage regulators, these devices can be used with external components for adjustable voltages and currents.



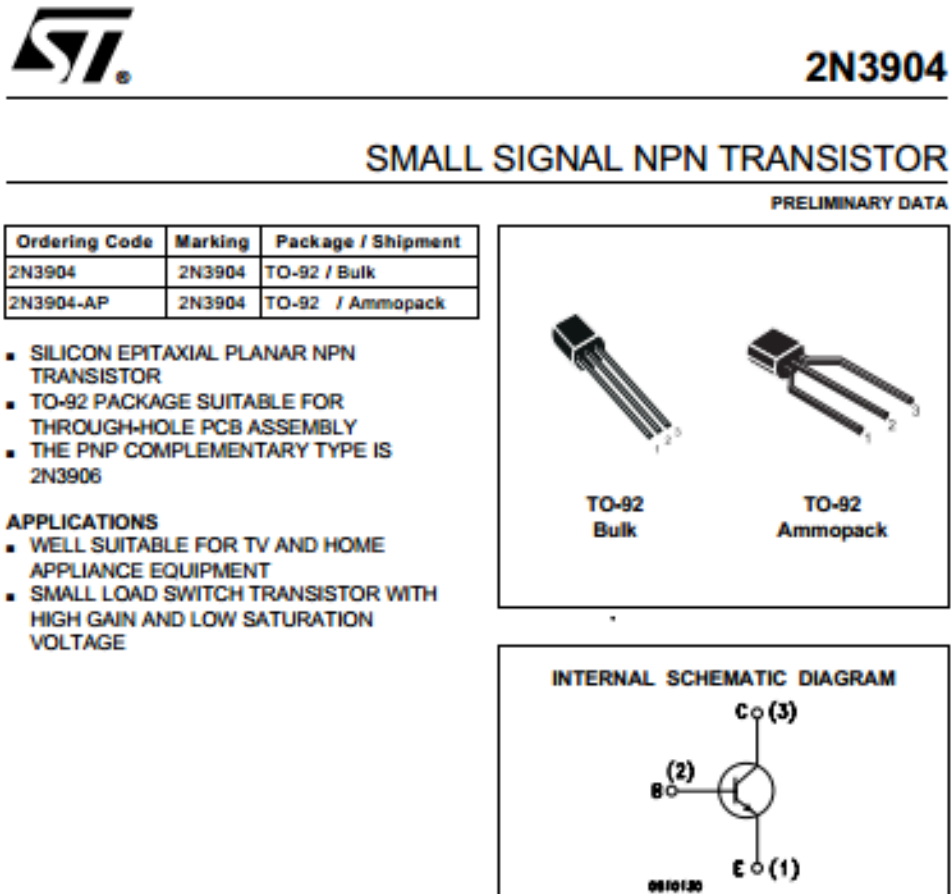
Ordering Information⁽¹⁾

Product Number	Output Voltage Tolerance	Package	Operating Temperature	Packing Method
LM7805CT	±4%	TO-220 (Single Gauge)	-40°C to +125°C	Rail
LM7806CT				
LM7808CT				
LM7809CT				
LM7810CT				
LM7812CT				
LM7815CT				
LM7818CT				
LM7824CT	±2%		0°C to +125°C	
LM7805ACT				
LM7809ACT				
LM7810ACT				
LM7812ACT				
LM7815ACT				

Figura 85: Regulador de Voltaje LM78XX

ANEXO 5

5Data sheet Transistor NPN 2N3904.



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CB0}	Collector-Base Voltage ($I_C = 0$)	60	V
V_{CE0}	Collector-Emitter Voltage ($I_B = 0$)	40	V
V_{EB0}	Emitter-Base Voltage ($I_C = 0$)	6	V
I_C	Collector Current	200	mA
P_{tot}	Total Dissipation at $T_C = 25\text{ }^\circ\text{C}$	625	mW
T_{stg}	Storage Temperature	-65 to 150	$^\circ\text{C}$
T_j	Max. Operating Junction Temperature	150	$^\circ\text{C}$

Figura 86: Data sheet Transistor NPN 2N3904

ANEXO 6

6 Construcción del robot de locomoción.

Primero se procedió realizar los soportes basado en el método del rocker-bogie.

Cortamos la barra de aluminio según las medidas de los planos y armamos sus piezas, primero el soporte superior.



Figura 87: Corte de barras de aluminio rocker y bogie.

Armamos las piezas de los soportes inferiores.



Figura 88: Colocación de pernos de ajustes.

Juntamos las partes superiores e inferiores de los soportes.



Figura 89: Armado del soporte con los pernos.

Colocamos los servos en los soportes inferiores frontales.



Figura 90: Colocación de servos.

Ahora se colocan los motores en todos los soportes inferiores.



Figura 91: Cambio de las agarraderas de motores.

En primera instancia quedaría de la siguiente manera.



Figura 92: Vista inicial del soporte con ruedas.

Luego procedemos con el ensamblaje del chasis.

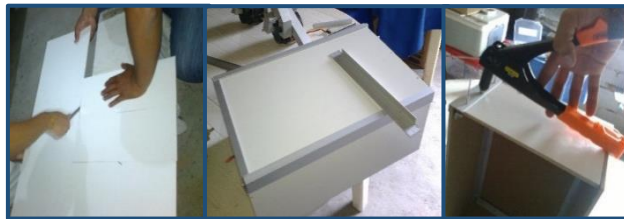


Figura 93: Ensamblaje del Chasis.

Estas piezas nos servirán para armar nuestras mini chumaceras



Figura 94: Partes de la mini chumacera.

Se procede a armar el soporte interno con las chumaceras y la varilla sin fin.

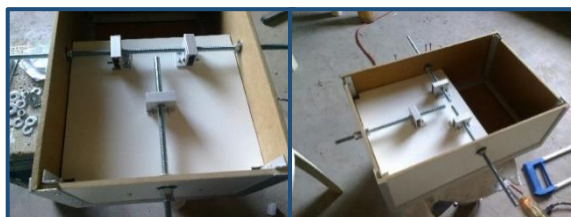


Figura 95: Armado del soporte interno.

Para la elaboración de las ruedas se necesitaron tubos pvc 8cm y Tapas para tubos pvc de 8cm

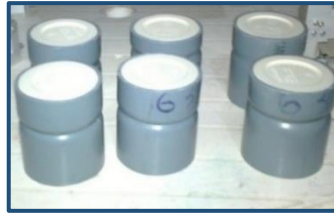


Figura 96: Base de las ruedas con tubos PVC.

El caucho de bicicleta nos sirve para el recubrimiento de las ruedas



Figura 97: Recubrimiento de las bases de las ruedas con caucho de bicicleta.

Colocamos pernos de 1 Pulgada par ajustar el caucho al tubo.



Figura 98: Vista de las llantas con recubrimiento.

Luego uniremos el soporte con el chasis.



Figura 99: Unión de la base con los soportes.

Anteriormente le añadimos un soporte para el sensor ultrasónico en la parte delantera de chasis. Luego pasamos hacer un soporte para los componentes electrónicos y lo colocamos dentro de la base del robot.

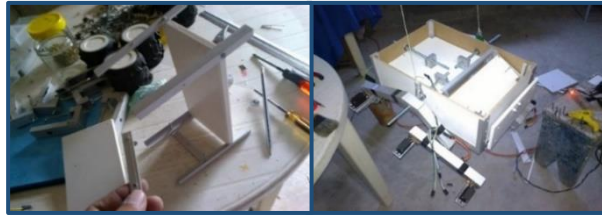


Figura 100: Construcción de soporte para circuitos.

Una vez listo comenzamos a colocar los componentes electrónicos.



Figura 101: Colocación de circuitos eléctricos.

Después de haber colocado los componentes procedemos con las pruebas



Figura 102: Realización de pruebas.

Peso aproximado de nuestro vehículo oscila los 19 a 20 libras (8,6 a 9 kg).

DISEÑO FINALIZADO



Figura 103: Diseño de nuestro robot móvil terminado.

ANEXO 7

7Manual de usuario.

Manual para el manejo del prototipo de locomoción.

Este manual nos ayudará en el proceso de preparación del robot y su posterior ejecución mediante los software asignados.

Software requerido para el funcionamiento y manejo del robot.

- Windows 7, 8, 8.1,10
- Arduino 1.6.5
- Visual studio 2015

Antes de todo debemos encender nuestro prototipo, por lo tanto debemos saber cómo funciona nuestro control físico del robot.

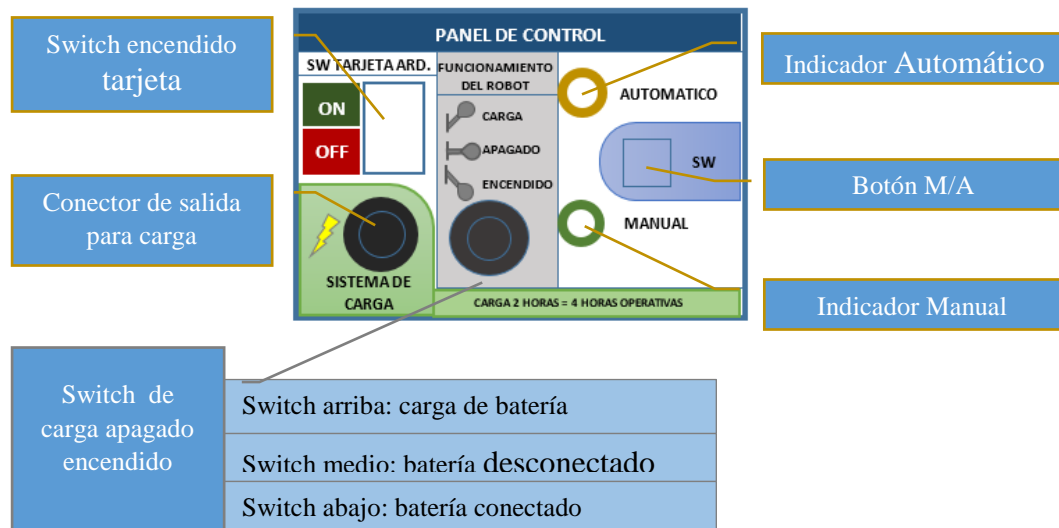


Figura 104: Panel de Control.

Al tener activado el Botón M/A en “manual” podemos hacer la interacción entre el robot y nuestro software en la pc. Caso contrario si está en modo automático ejecutará los movimientos que fueron grabados con anterioridad. Después de saber estos ajustes podemos seguir con nuestro software.

Primero debemos tener cargada la programación de Arduino en la tarjeta Mega.

Para esto realizamos los siguientes pasos:

Conecte el cable USB de la tarjeta Arduino a la pc.

Iniciamos el software Arduino.

Cargamos nuestro programa.

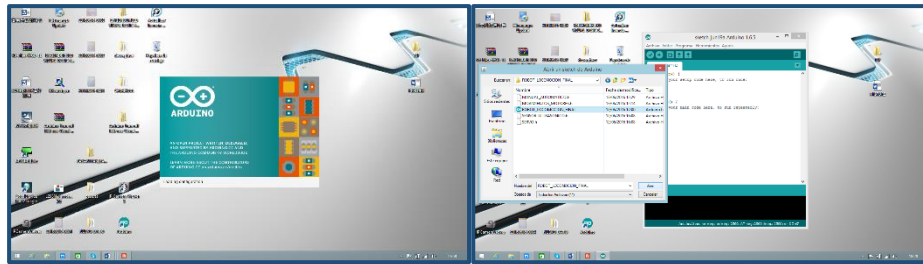


Figura 105: Carga del código en el programa Arduino.

Compilamos

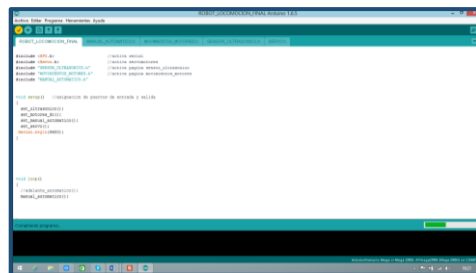


Figura 106: Compilamos el código en Arduino.

Luego asignaremos nuestra placa Arduino mega.

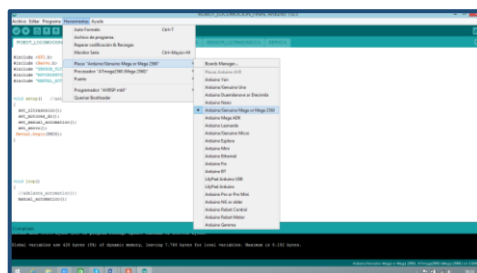


Figura 107: Asignación de placa Arduino.

Después asignamos el puerto serial correspondiente a nuestra placa Arduino. En nuestro caso es “COM 8”

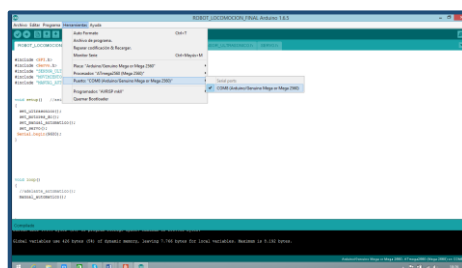


Figura 108: Asignación de puerto serial.

Antes de subir el programa necesitamos desconectar el pin de recepción inalámbrica. (Rx)

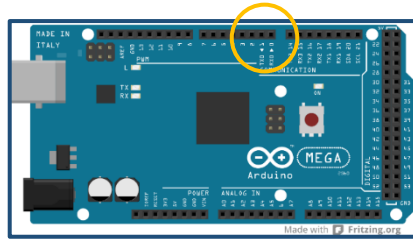


Figura 109: Desconexión del pin de transmisión en T. Arduino.

Procedemos a subir la programación a la tarjeta.

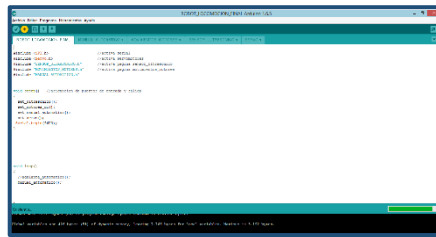


Figura 110: Compilación del programa Arduino.

Una vez subido desconectamos la tarjeta del computador y volvemos a conectar el pin de recepción (Rx) de la tarjeta Arduino.

Luego pasamos a cargar el código de visual Basic studio 2015.

Primero abrimos visual studio 2015.



Figura 111: Ejecución del programa visual studio.

Abrimos el código haciendo clic en nuestro archivo, en nuestro caso lo tenemos en archivos reciente:

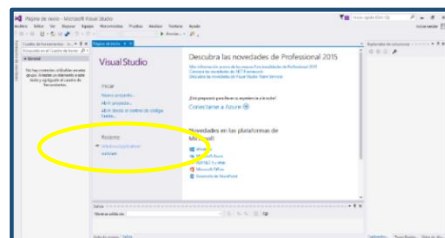


Figura 112: Ítem de acceso a nuestro código.

Se abrirá la siguiente ventana

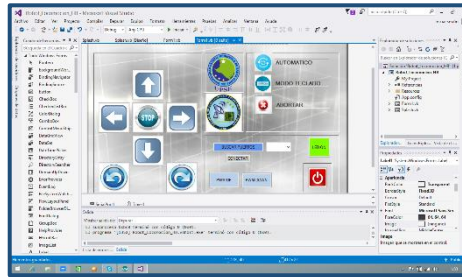


Figura 113: Ventana de desarrollo de aplicación.

Antes de iniciar debemos conectar nuestra antena XBee S2 y luego cambiar el puerto serial de nuestro software Arduino.

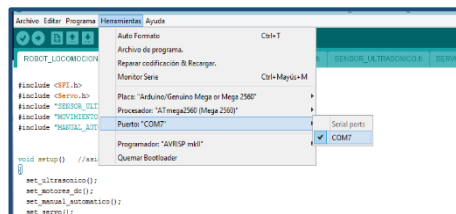


Figura 114: Conexión de puerto serial en Arduino.

También debemos encender nuestro prototipo.

Seguido volvemos a nuestro programa de Visual y seleccionamos el botón iniciar.

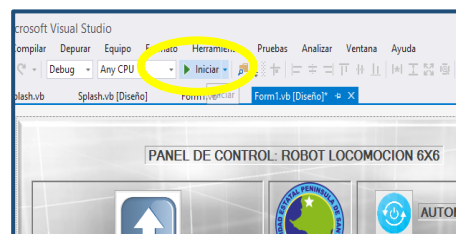


Figura 115: Ubicación del botón inicio.

Y nuestro programa de control comenzará a ejecutarse.



Figura 116: Ejecución del programa.

Aquí podemos observar los diferentes botones de direccionamiento y ejecución.

ADELANTE:	El robot se moverá hacia el frente.
ATRÁS:	El robot se moverá hacia atrás.
IZQUIERDA:	El robot girara hacia la izquierda.
DERECHA:	El robot girara hacia la derecha.
IZQ_ATRAS:	El robot girara hacia atrás en dirección a la izquierda.
DER_ATRAS:	El robot girara hacia atrás en dirección a la derecha.
PARAR:	El robot se detendrá.
MODO AUTOMATICO:	El robot ejecutara la ruta especificada anteriormente.
MODO TECLADO:	Este botón activara el teclado de nuestra pc.
SALIR:	El programa terminara de ejecutarse.

Tabla 23: Descripción de direccionamiento.

También cuenta con el icono de ejecución rápida.



Figura 117: Acceso rápido de nuestra aplicación.

PROCESO DE CARGA

Después de su uso en un máximo de 4 horas la carga en la batería comienza a disminuir y por ende sus movimientos son más lentos. Para esto el prototipo tiene una conexión de carga externa la cual nos ayudará a cargar completamente nuestra batería en un periodo de 2 horas. Lo único que debemos hacer es colocar el Switch en modo “carga” y conectamos a nuestro prototipo el cable adaptador con el cargador. El voltaje de nuestro cargador debe ser mayor de 12 voltios y 8 Amp.

A continuación se puede observar este proceso.



Figura 118: Conexión de carga de batería.

ANEXO 8

8Codificación en visual Basic studio 2015

PARTE "A": VENTANA Form1.vb

SPLASH/PRESENTACION

```
Public Class Splash
    Dim contador As Byte = 0
    Private Sub Splash_Load(sender As Object, e
As EventArgs) Handles MyBase.Load
        End Sub
    Private Sub Timer1_Tick(sender As Object, e
As EventArgs) Handles Timer1.Tick
        If ProgressBar1.Value < 100 Then
            contador = contador + 10
            ProgressBar1.Value = contador
        Else
            Me.Hide()
            Form1.Show()
            Timer1.Enabled = False
        End If
    End Sub
End Class
```

FORM1/GENERAL

```
Public Class Form1
    Dim StrBufferEntrada As String
    Dim StrBufferSalida As String
    Private Sub Form1_Load(sender As Object, e
As EventArgs) Handles MyBase.Load
        StrBufferEntrada = ""
        StrBufferSalida = ""
        BtnConectar.Enabled = False
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = False
        Button7.Enabled = False
        Button11.Enabled = False
        Button12.Enabled = False
        Button5.Enabled = False
        Button6.Enabled = False
        Button9.Enabled = False
        Button10.Enabled = False
        PWM_UP.Enabled = False
        PWM_DOWN.Enabled = False
        LED.Enabled = False
        LED_OFF.Enabled = False
        PictureBox1.Enabled = False
        PictureBox2.Enabled = False
        PictureBox3.Enabled = False
        PictureBox5.Enabled = False
        PictureBox6.Enabled = False
    End Sub
    Private Sub Button1_Click(sender As Object, e
As EventArgs) Handles Button1.Click
        SerialPort1.Write("e") 'ADELANTE
    End Sub
    Private Sub Button2_Click(sender As Object, e
As EventArgs) Handles Button2.Click
```

```
        SerialPort1.Write("x") 'ATRAS
    End Sub
    Private Sub Button3_Click(sender As Object, e
As EventArgs) Handles Button3.Click
        SerialPort1.Write("s") 'IZQUIERDA
    End Sub
    Private Sub Button4_Click(sender As Object, e
As EventArgs) Handles Button4.Click
        SerialPort1.Write("d") 'DERECHA
    End Sub
    Private Sub Button7_Click(sender As Object, e
As EventArgs) Handles Button7.Click
        SerialPort1.Write("a") 'PARO
    End Sub
    Private Sub Button11_Click(sender As Object,
e As EventArgs) Handles Button11.Click
        SerialPort1.Write("z") 'IZQ_ATRAS
    End Sub
    Private Sub Button12_Click(sender As Object,
e As EventArgs) Handles Button12.Click
        SerialPort1.Write("c") 'DER_ATRAS
    End Sub
    Private Sub Button5_Click(sender As Object, e
As EventArgs) Handles Button5.Click
        SerialPort1.Write("g") 'AUTOMATICO
    End Sub
    Private Sub Button6_KeyPress(sender As
Object, e As KeyPressEventArgs) Handles
Button6.KeyPress
        If e.KeyChar.ToString = "e" Then 'adelante
            SerialPort1.Write("e")
        End If
        If e.KeyChar.ToString = "x" Then 'atras
            SerialPort1.Write("x")
        End If
        If e.KeyChar.ToString = "s" Then 'izquierda
            SerialPort1.Write("s")
        End If
        If e.KeyChar.ToString = "d" Then 'derecha
            SerialPort1.Write("d")
        End If
        If e.KeyChar.ToString = "a" Then 'paro
            SerialPort1.Write("a")
        End If
        If e.KeyChar.ToString = "z" Then 'atrasizq
            SerialPort1.Write("z")
        End If
        If e.KeyChar.ToString = "c" Then 'atrasdere
            SerialPort1.Write("c")
        End If
        If e.KeyChar.ToString = "o" Then 'posici
            SerialPort1.Write("o")
        End If
        If e.KeyChar.ToString = "p" Then 'posici
            SerialPort1.Write("p")
```

```

End If
If e.KeyChar.ToString = "l" Then 'abortar
    SerialPort1.Write("l")
End If
If e.KeyChar.ToString = "k" Then 'reabortar
    SerialPort1.Write("k")
End If
If e.KeyChar.ToString = "y" Then '+apwm
    SerialPort1.Write("y")
End If
If e.KeyChar.ToString = "u" Then '- pwm
    SerialPort1.Write("u")
End If
End Sub
Private Sub Button9_Click(sender As Object, e
As EventArgs) Handles Button9.Click
    SerialPort1.Write("l") 'abortar rojo
    Button9.Hide() 'se escnde boton rojo
    Button10.Visible = True 'visualiz bv
    Label4.Visible = False 'se escond conectar
    Label5.Visible = True 'vis abortar
End Sub
Private Sub Button10_Click(sender As Object,
e As EventArgs) Handles Button10.Click
    SerialPort1.Write("k") 'conectar verde
    Button10.Hide()
    Button9.Visible = True
    Label5.Visible = False
    Label4.Visible = True
End Sub
Private Sub Button8_Click(sender As Object, e
As EventArgs) Handles Button8.Click
    If MsgBox("¿DESEAS SALIR DE LA
APLICACION?", vbQuestion + vbYesNo,
"PREGUNTA") = vbYes Then
        Me.Close()
        Splash.Close()
    End If
End Sub
Private Sub Timer1_Tick(sender As Object, e
As EventArgs) Handles Timer1.Tick
    hora.Text = Now.ToLongTimeString
    fecha.Text = Now.ToLongDateString
End Sub
Dim pwm1 = 250
Private Sub PWM_UP_Click(sender As
Object, e As EventArgs) Handles
PWM_UP.Click
    SerialPort1.Write("y")
    pwm1 = pwm1 + 10
    If pwm1 < 380 Then
        PWM.Text = pwm1
    End If
End Sub
Private Sub PWM_DOWN_Click(sender As
Object, e As EventArgs) Handles
PWM_DOWN.Click
    SerialPort1.Write("u")
    pwm1 = pwm1 - 10
    If pwm1 > 130 Then
        PWM.Text = pwm1
    End If
End Sub

Private Sub LED_Click(sender As Object, e As
EventArgs) Handles LED.Click
    SerialPort1.Write("n") 'LED on
    LED_OFF.Visible = True
    LED.Hide()
End Sub
Private Sub LED_OFF_Click(sender As
Object, e As EventArgs) Handles
LED_OFF.Click
    SerialPort1.Write("m") 'LED off
    LED.Visible = True
    LED_OFF.Hide()
End Sub
Private Sub BtnBuscarPuertos_Click(sender
As Object, e As EventArgs) Handles
BtnBuscarPuertos.Click
    CboPuertos.Items.Clear
    For Each PuertoDisponible As String In
My.Computer.Ports.SerialPortNames
        CboPuertos.Items.Add(PuertoDisponible)
    Next
    If CboPuertos.Items.Count > 0 Then
        CboPuertos.Text = CboPuertos.Items(0)
        BtnConectar.Enabled = True
    Else
        MessageBox.Show("NINGUN PUERTO
ENCONTRADO")
        BtnConectar.Enabled = False
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = False
        Button7.Enabled = False
        Button11.Enabled = False
        Button12.Enabled = False
        Button5.Enabled = False
        Button6.Enabled = False
        Button9.Enabled = False
        Button10.Enabled = False
        PWM_UP.Enabled = False
        PWM_DOWN.Enabled = False
        LED.Enabled = False
        LED_OFF.Enabled = False
        PictureBox1.Enabled = False
        PictureBox2.Enabled = False
        PictureBox3.Enabled = False
        PictureBox5.Enabled = False
        PictureBox6.Enabled = False
        CboPuertos.Items.Clear()
    End If
End Sub
Private Sub BtnConectar_Click(sender As
Object, e As EventArgs) Handles
BtnConectar.Click
    If BtnConectar.Text = "CONECTAR" Then
        Try
            With SerialPort1
                .BaudRate = 9600
                .DataBits = 8
                .Parity = IO.Ports.Parity.None
                .StopBits = IO.Ports.StopBits.One
                .PortName = CboPuertos.Text
                .Open()
            End With

```

```
    BtnConectar.Text = "DESCONECTAR"  
    Button1.Enabled = True  
    Button2.Enabled = True  
    Button3.Enabled = True  
    Button4.Enabled = True  
    Button7.Enabled = True  
    Button11.Enabled = True  
    Button12.Enabled = True  
    Button5.Enabled = True  
    Button6.Enabled = True  
    Button9.Enabled = True  
    Button10.Enabled = True  
    PWM_UP.Enabled = True  
    PWM_DOWN.Enabled = True  
    LED.Enabled = True  
    LED_OFF.Enabled = True  
    PictureBox1.Enabled = True  
    PictureBox2.Enabled = True  
    PictureBox3.Enabled = True  
    PictureBox5.Enabled = True  
    PictureBox6.Enabled = True  
    Catch ex As Exception  
        MsgBox(ex.Message,  
MsgBoxStyle.Critical)  
    End Try  
    ElseIf BtnConectar.Text =  
"DESCONECTAR" Then
```

```
    BtnConectar.Text = "CONECTAR"  
    Button1.Enabled = False  
    Button2.Enabled = False  
    Button3.Enabled = False  
    Button4.Enabled = False  
    Button7.Enabled = False  
    Button11.Enabled = False  
    Button12.Enabled = False  
    Button5.Enabled = False  
    Button6.Enabled = False  
    Button9.Enabled = False  
    Button10.Enabled = False  
    PWM_UP.Enabled = False  
    PWM_DOWN.Enabled = False  
    LED.Enabled = False  
    LED_OFF.Enabled = False  
    PictureBox1.Enabled = False  
    PictureBox2.Enabled = False  
    PictureBox3.Enabled = False  
    PictureBox5.Enabled = False  
    PictureBox6.Enabled = False  
    SerialPort1.Close()  
    End If  
    End Sub  
End Class
```