

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL
TÍTULO EN INGENIERÍA**

**DISEÑO E IMPLEMENTACIÓN DE UN ROBOT MÓVIL
AUTÓNOMO Y TELEOPERADO PARA LABORES
AGRÍCOLAS.**

**FASE 1: FUMIGACIÓN PARA PLANTACIONES DE
ROSAS**

**IVÁN ANDRÉS LEÓN VÁSQUEZ
ADRIÁN JONATAN VÁSQUEZ TAPIA**

**SANGOLQUÍ – ECUADOR
2013**

ESCUELA POLITÉCNICA DEL EJÉRCITO

INGENIERÍA ELEÉCTRICA Y ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

DECLARACIÓN DE RESPONSABILIDAD

**LEÓN VÁSQUEZ IVÁN ANDRÉS
VÁSQUEZ TAPIA ADRIÁN JONATAN**

El proyecto denominado “DISEÑO E IMPLEMENTACIÓN DE UN ROBOT MÓVIL AUTÓNOMO Y TELEOPERADO PARA LABORES AGRICOLAS FASE 1: FUMIGACIÓN PARA PLANTACIONES DE ROSAS”, ha sido desarrollado con base a una investigación exhaustiva respetando los derechos intelectuales de terceros, conforma constan al final de cada párrafo correspondiente, según normas APA y cuyas fuentes se incorporan en la bibliografía.

Sangolquí, Junio 2013

León Vásquez Iván Andrés

Vásquez Tapia Adrián Jonatan

ESCUELA POLITÉCNICA DEL EJÉRCITO

INGENIERÍA ELEÉCTRICA Y ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Nosotros, León Vásquez Iván Andrés y Vásquez Tapia Adrián Jonatan

Autorizamos a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la Institución el trabajo “DISEÑO E IMPLEMENTACIÓN DE UN ROBOT MÓVIL AUTÓNOMO Y TELEOPERADO PARA LABORES AGRICOLAS FASE 1: FUMIGACIÓN PARA PLANTACIONES DE ROSAS”, cuyo contenido, ideas y criterio son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, Junio 2013

León Vásquez Iván Andrés

Vásquez Tapia Adrián Jonatan

CERTIFICACIÓN

Certificamos que la elaboración del presente proyecto fue realizada en su totalidad por Iván Andrés León Vásquez con CI: 171483165-6 y Adrián Jonatan Vásquez Tapia con CI: 0604098103, como requisito para la obtención del título en INGENIERÍA ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL.

Sangolquí, Junio 2013.

DIRECTOR
Ing. Hugo Ortiz.

CODIRECTOR
Ing. Alex Ibarra.

AGRADECIMIENTO

A Dios y la vida por darnos la oportunidad de vivir en el tiempo que debimos, con las personas que debimos y los mejores amigos que pudimos.

A nuestros padres por el cariño amor y apoyo incondicional a lo largo de nuestras vidas.

A la Escuela Politécnica del Ejército, por acogernos en sus instalaciones y darnos la oportunidad de recibir una formación integral.

Al Ing. Hugo Ortiz por su apoyo y guía incondicionales a lo largo de nuestra carrera, en especial en momentos difíciles de alta presión.

Al Ing. Manuel Freire, por ser nuestro tutor y mentor profesional, y compartir conocimiento de manera desinteresada.

A la Ing. Amparo de la Vega, por ser la guía en plantaciones de rosas, sin ti no hubiese sido esto posible.

A nuestros profesores por sembrar las bases para convertirnos en profesionales éticos y dignos.

DEDICATORIA

Dedico este logro a mi familia por su apoyo incondicional, guía, y por darme siempre fuerzas para seguir cuando todo parecía perdido, a mis amigos por ser un apoyo incondicional en este proceso y a Dios por ser mi guía siempre.

Iván

A mi madre Nancy Tapia, mi amiga, mi ejemplo a seguir, mi confidente. Usted, también es mi bandera Mamá.

A la mujer que formo mi niñez, que me hizo un hombre de bien, bueno y sencillo, mi ángel de la guarda Mercedes G Paredes G (+), la extraño mucho.

A mi hermano mayor, quien me enseñó a amar la vida a cada instante, gracias por hacerme tan feliz Patricio Constante (+), pronto estaremos juntos de nuevo.

A David, mi entrañable hermano, gracias por llegar a alegrar nuestras vidas de la manera menos esperada mi "Rock" querido.

A mi "ñaño chiquito", mi compañero y compinche de travesuras, Leonidas. Te quiero mucho "ñañote".

A ti Paola, por tantas riñas y discusiones, por besos y canciones, por quererme tal como soy. Gracias gordita.

Adrián

PRÓLOGO

El presente proyecto documenta los procesos desarrollados para el diseño y construcción de un robot prototipo autónomo y tele-operado capaz de realizar la fumigación en plantaciones de rosas.

Este documento busca servir de base para que a futuro se desarrollen prototipos similares de mayor versatilidad, y así aportar al desarrollo agrario y robótico del país.

Con el fin de cumplir con los objetivos del proyecto, se ha realizado una investigación de campo en plantaciones de rosas, para evidenciar los problemas presentes en ellas, que se detallan en el Capítulo 1 de este documento. Para la construcción mecánica que corresponde al Capítulo 2, se diseñó un sistema móvil con ruedas en configuración diferencial para proveer mayor estabilidad a la plataforma, y que vaya acorde con las dimensiones del terreno de desplazamiento.

En el Capítulo 3 se desarrolla el diseño de las distintas etapas de hardware, las cuales se diseñaron en base a la construcción mecánica. Para ello se utilizan elementos eléctricos y electrónicos necesarios para realizar el proceso de fumigación. Posteriormente, el Capítulo 5 detalla el diseño de software del sistema, el cual permite realizar el control de la plataforma mediante el uso de programación secuencial, a través de una interfaz humano maquina (HMI) que además permite guardar datos históricos en una base de datos.

Finalmente, se ha documentado las pruebas y resultados de la plataforma bajo condiciones ideales, en donde se detalla cuantitativamente el número de pruebas exitosas de la plataforma, y los errores que se presentaron.

ÍNDICE DE CONTENIDO

PRÓLOGO	vii
ÍNDICE DE CONTENIDO	viii
CAPÍTULO 1	1
INTRODUCCIÓN A LA ROBÓTICA AGRÍCOLA.....	1
1.1. ANTECEDENTES	1
1.1.1. Objetivos	3
1.2. EVOLUCIÓN	4
1.2.1. Reseña Histórica de la Robótica	4
1.2.2. Robótica Agrícola	6
1.3. ESTUDIO DE PROCESOS Y NECESIDADES EN LOS CAMPOS DE ROSAS	8
1.4. APLICACIONES	17
CAPÍTULO 2	18
DISEÑO DEL SISTEMA MECÁNICO	18
2.1. CONCEPTUALIZACIÓN	18
2.1.1. Arreglos de Ruedas	19
2.1.2. Distribución del Espacio Físico	24
2.2. SÍNTESIS	26
2.3. ANÁLISIS	27
2.4. EVALUACIÓN	37
2.4.1. Ecuaciones Para La Cinemática Diferencial	38
2.5. REPRESENTACIÓN	40
CAPÍTULO 3	42
DISEÑO DE HARDWARE	42
3.1. ETAPA DE POTENCIA	42
3.1.1. Movilidad	42
3.1.2. Control De Fumigación	46
3.2. ETAPA DE COMUNICACIÓN DE DATOS	54
3.2.1. Módulos Xbee Serie 1	54
3.3. ETAPA DE CONTROL	62
3.4. ETAPA DE INSTRUMENTACIÓN	63
3.4.1. Medición De Distancia	63
3.4.2. Medición de nivel	68

3.5. SISTEMA DE ALIMENTACIÓN	69
CAPÍTULO 4	70
DESARROLLO DE SOFTWARE	70
4.1. LÓGICA DE CONTROL	70
4.1.1. Distribución y Configuración De Puertos	71
4.1.2. Esquema Principal Del Firmware	73
4.1.3. Modo Manual	75
4.1.4. Modo Automático	83
4.2. HMI	86
4.2.1. Programación En Netbeans	87
4.2.2. Distribución Gráfica	91
4.3. DESARROLLO DE LA BASE DE DATOS	98
CAPÍTULO 5	101
IMPLEMENTACIÓN	101
5.1. CONSTRUCCIÓN MECÁNICA	101
5.1.1. Selección de Materiales	101
5.1.2. Montaje de Piezas	102
5.2. CONSTRUCCIÓN E INTEGRACIÓN ELÉCTRICA Y ELECTRÓNICA	111
5.2.1. Conexión de Alimentación General	114
5.2.2. Conexión de Motores	114
5.2.3. Conexión de Transmisión de Datos	114
5.2.4. Conexión de Electroválvulas y Bomba de Succión	115
5.2.5. Conexión de Sensores	115
5.3. PUESTA EN MARCHA	115
5.3.1. Campo de pruebas	115
5.3.2. Plataforma	119
CAPITULO 6	125
PRUEBAS Y RESULTADOS	125
6.1. PRUEBAS	125
6.1.1. Pruebas del Sistema Mecánico	125
6.1.2. Pruebas del Sistema de Fumigación	128
6.1.3. Pruebas de Sensores	130
6.1.4. Pruebas de los Módulos Serie Inalámbricos Xbee Serie 1	138
6.1.5. Pruebas de la Tarjeta de Control	140
6.1.6. Pruebas de Interfaz Humano Máquina	142
6.2. RESULTADOS	145
6.2.1. Modo Manual	145
6.2.2. Modo Automático	154
CAPÍTULO 7	173
CONCLUSIONES Y RECOMEDACIONES	173

7.1. CONCLUSIONES	173
7.2. RECOMENDACIONES	175
REFERENCIAS BIBLIOGRÁFICAS	177
ANEXO A	178
<i>CODIGO FUENTE DEL CONTROLADOR EN LENGUAJE C</i>	178
ANEXO B	1
<i>CODIGO FUENTE LA INTERFAZ JAVA</i>	1
ANEXO C	2
<i>HOJAS TÉCNICAS DE LOS DISPOSITIVOS UTILIZADOS</i>	2
ANEXO D	1
<i>DIAGRAMAS ELECTRÓNICOS</i>	1
ÍNDICE DE FIGURAS	B
ÍNDICE DE TABLAS	E
GLOSARIO	F

CAPÍTULO 1

INTRODUCCIÓN A LA ROBÓTICA AGRÍCOLA

1.1. ANTECEDENTES

La robótica es la ciencia y la tecnología de los robots, los cuales son un ejemplo bastante completo de un sistema artificial inteligente, puesto que, cuentan con sus propios receptores y efectores que les permiten captar características de su entorno, ganar experiencia y cumplir con un determinado objetivo para el que fueron diseñados.

El vertiginoso avance que ha tenido la tecnología en los últimos años, ha permitido vislumbrar un panorama favorable en el ámbito de la robótica, ya que, estos dispositivos constarán cada día con nuevas y mejores herramientas para su desarrollo. Esto conllevará sin lugar a duda a obtener robots de gran calidad y cada vez a menores precios, y con distintos tipos de aplicación (Jorge Valverde Rebaza-2009).

“Las empresas nacionales dedicadas a la robótica son pequeñas y están dirigidas por

jóvenes que terminan sus estudios y que tratan de abrirse un espacio en el mercado” (Explored-2009).

“Estas compañías tienen que enfrentar el hecho de que los robots que existen en los procesos de automatización industrial del país son importados, especialmente desde Japón, Alemania y los Estados Unidos” (Explored-2009). Actualmente en el agro la población robótica es prácticamente nula, esto a pesar de que allí se realizan tareas repetitivas, y algunas incluso sumamente peligrosas para el ser humano, por las sustancias tóxicas que se esparcen en los campos.

Según datos de la FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura), la explosión demográfica ha causado que los empresarios agrícolas, busquen formas de brindar a sus trabajadores nuevas herramientas de tal manera que faciliten su trabajo e incrementen la producción mensual, además de salvaguardar su salud.

Varias compañías agrícolas han buscado por largo tiempo, la manera de mecanizar procesos dados los altos niveles de competitividad que se presentan en el mercado, pero no se ha logrado encontrar soluciones efectivas a esta medida.

La empresa floricultora HILSEA, admite que necesita ayuda robótica para la fumigación de sus campos, con el fin de salvaguardar la integridad física, y biológica de sus operarios, dado que los fungicidas e insecticidas son sumamente tóxicos para la salud, por este motivo muchos de sus operarios se muestran renuentes a realizar esta actividad continuamente.

La compañía de exportación de rosas Magnolia, indica que necesita optimizar el tiempo de sus operarios, dado que generalmente este tiempo se gasta en tareas repetitivas de alto desgaste físico. La fumigación es una parte de estas, dado que, se necesita recorrer

grandes distancias sosteniendo las herramientas de fumigación que se realizan varias veces por semana e incluso algunas por día, lo cual admite que se podría mejorar utilizando una herramienta robótica.

La necesidad de automatizar procesos en la agricultura como es la fumigación, constituye un método para optimizar los niveles de eficiencia tanto de mano de obra como de la producción, que implica mayores ingresos para las compañías, mejor calidad en el producto, lo que conllevaría a un mayor desarrollo del país.

1.1.1. Objetivos.

En base a lo anteriormente expuesto, se han planteado los siguientes objetivos:

1.1.1.1. *General:*

- Diseñar e implementar un robot móvil, autónomo y tele operado que ejecute la fumigación en plantaciones de rosas, utilizando herramientas y dispositivos electrónicos, mecánicos e informáticos, para que sirva de herramienta a los trabajadores agrícolas.

1.1.1.2. *Específicos:*

- Realizar una investigación técnica – bibliográfica para estructurar el marco teórico científico que fundamente al proyecto.
- Realizar una investigación de campo para determinar las condiciones óptimas necesarias para la implementación del robot (Área y campo de operación).
- Diseñar e implementar la plataforma móvil, así como las distintas piezas y

artefactos mecánicos, que influirán en la movilidad del robot.

- Diseñar e implementar el hardware de control y transmisión de datos del robot.
- Desarrollar una interfaz entre el operador y el robot, para que sea fácil de operar remotamente.
- Realizar pruebas de movilidad y fumigación del robot en el campo de rosas.

1.2. EVOLUCIÓN

1.2.1. Reseña Histórica de la Robótica.

La concepción en sí de un autómatas para realizar distintas tareas, se remonta muchos siglos atrás, incluso antes de Cristo. Por ejemplo, en Alejandría se tienen descripciones de más de 100 máquinas y autómatas, incluyendo un artefacto con fuego, un órgano de viento, una máquina operada mediante una moneda, una máquina de vapor. De allí nace la idea de evolucionar dichas máquinas a fin de que ellas sirvan de herramientas al ser humano en distintas aéreas y campos de aplicación.

A partir de allí, los avances robóticos fueron avanzando, con herramientas y dispositivos rústicos, pero que en cierto modo tenían un alto nivel de efectividad en la tarea que realizaban. Fue así que, a través de los siglos se dieron nuevos inventos en esta rama de la tecnología, con nombres tan ilustres como el de Leonardo da Vinci, quien ya vaticinaba un futuro prometedor para estos artefactos.

Es así, que el gran auge de la robótica que vivimos hoy por hoy, empieza en el siglo XX, motivado por la Revolución Industrial que en este siglo tomó parte. Como un modo de mejorar la eficiencia de las industrias, se introdujeron robots que disminuían el tiempo de fabricación de diversos bienes a fin de elevar la productividad y por ende las ganancias, en un mercado cada vez más competitivo.

Es así que partiendo de esta breve reseña histórica podemos definir el avance de la robótica utilizando generaciones de evolución en la tabla 1.01:

Tabla 1.01. Generaciones Robóticas (Jorge Valverde Rebaza-2009).

GENERACIONES	BREVE DESCRIPCIÓN
Primera Generación	Empezaron como robots manipuladores industriales, con sistemas de control sumamente elementales y que siempre se operaban manualmente.
Segunda Generación	Robots que introdujeron el aprendizaje, es decir un operador humano ejecuta una secuencia que posteriormente el robot debía repetirla por sí mismo.
Tercera Generación	En esta generación se introdujeron los sensores a los robots, a más de tener un control avanzado por medio de programas computarizados.
Cuarta Generación	Robots “inteligentes”, capaces de reconocer mediante sensores el estado actual del proceso que se está ejecutando. De esta forma logran retroalimentar a su controlador para que tome las decisiones

	adecuadas ante los inconvenientes que se presenten, los cuales deben ser previamente programados en el controlador.
--	---

1.2.2. Robótica Agrícola.

La robótica tiene una gran capacidad de aplicación en la agricultura, especialmente por el nivel de precisión con la que cuentan los robots actuales, sin embargo, esto aún no es suficiente para entornos reales en donde, uno de los principales problemas es la movilidad de estos sobre los terrenos agrarios.

La aplicación de la robótica en la agricultura recibe el nombre de *Robótica Agrícola*, la cual se define como la tecnología de la automatización aplicada a bio-sistemas tales como la agricultura, la silvicultura, las industrias pesqueras y otras afines. Esta rama de la robótica ha venido desarrollándose con fuerza en los últimos años, logrando resultados muy significativos en robots que desarrollan labores agrícolas (Jorge Valverde Rebaza-2009).

La *Robótica Agrícola* tiene como uno de sus más importantes objetivos equipar los campos de cultivo con pequeños robots, ligeros y baratos, que especialmente sean concebidos para realizar tareas agrícolas y sustituir a las pesadas y costosas maquinarias que se emplean actualmente para sembrar, fumigar, recolectar y arar la tierra, tareas fundamentales en el agro (Jorge Valverde Rebaza-2009).

El objetivo es lograr que estos robots abarquen la totalidad de las tareas agrícolas. Para cristalizar este proyecto, el desarrollo y producción de robots agrícolas se ha dividido en 3

generaciones que se convertirán a la vez en 3 fases de penetración de agro tecnología al mundo (Jorge Valverde Rebaza-2009).

La **Primera Generación** de la Robótica Agrícola es la que estamos viviendo y se caracteriza por que los robots agricultores son pequeños y están dotados principalmente de un sistema de visión artificial. Son capaces de realizar tareas de búsqueda y transmisión de información sobre el terreno de una forma totalmente nueva en el sector agrícola: distribuidos por hectáreas de terreno, son capaces de orientar sus búsquedas, de intercambiar información con otras unidades y de detectar epidemias e insectos peligrosos, advirtiéndolo a los demás robots desplegados sobre el terreno (Jorge Valverde Rebaza-2009).

La **Segunda Generación** pretende llegar más lejos y acometer tareas agrícolas más complejas, como roturar mecánicamente un terreno y fumigarlo con la ayuda de GPS. Estos robots también serán de pequeño tamaño, excepto los que se dediquen a cosechar, los cuales deberán tener un tamaño equivalente al de las máquinas actuales (Jorge Valverde Rebaza-2009).

La **Tercera Generación** formará parte de un sistema más amplio para gestionar en su conjunto un sistema agrícola íntegramente automatizado, con actividades complementarias a las estrictamente agrícolas, como la ganadería y la gestión comercial (Jorge Valverde Rebaza-2009).

Sin duda que se tiene un plan exacto de los resultados que se desean para el futuro cercano y esto se ve factible dado que, los resultados obtenidos hasta el momento muestran un panorama muy favorable. Dado que en el mundo se han empezado a desarrollar nuevos sistemas robóticos, por ejemplo, en Japón ya existen unos pequeños robots autónomos que plantan arroz en sembríos reales, adaptándose y desarrollando sus actividades a pesar del difícil terreno de los arrozales.

En el MIT (Instituto Tecnológico de Massachusetts) se encuentran desarrollando robots especialistas en plantaciones de tomates, los cuales son hidratados, nutridos, cosechados y hasta polinizados por estos robots (Jorge Valverde Rebaza-2009).

1.3. ESTUDIO DE PROCESOS Y NECESIDADES EN LOS CAMPOS DE ROSAS

Las exportaciones florícolas en el país, están entre los principales ingresos para el Ecuador. Por este motivo es necesario realizar mejoras continuas en los modos de producción y procesos agrarios que conlleva esta actividad. Las flores ecuatorianas son muy apreciadas en países desarrollados, por la diversidad de colores y especialmente por la calidad que estas poseen.

A fin de recolectar datos y verificar el campo de aplicación en el cual se desenvolverá el robot, se realizaron varias visitas a las instalaciones de la exportadora de rosas “Magnolia” ubicadas en Tabacundo, las cuales fueron supervisadas por la Ing. Amparo de la Vega Paredes. Mencionada ingeniera manifestó detalladamente los pasos a seguir en el proceso de cultivo de flores, dentro de los cuales se encuentra el de fumigación de plantas en sus distintas etapas.

El proceso de fumigación es determinado por varios aspectos, los cuales principalmente dependen del estado en que cada planta se encuentre, es decir, totalmente sana o afectada por alguna enfermedad.

A más de esto, las condiciones climáticas para realizar la fumigación son previamente establecidas, puesto que, al no acatar esta regla, se puede echar a perder la cosecha y peor aún matar las plantas, lo que conllevaría un enorme gasto económico y desperdicio de tiempo. Los químicos que se usan en la agricultura para proteger las plantas de las distintas enfermedades y plagas son extremadamente tóxicos. El nivel de toxicidad de cada sustancia química esta medida en un rango que va desde el numero I hasta el numero V, siendo el V el grado más alto de toxicidad.

Por ello es imperativo dar una solución que evite que los operarios entren en contacto directo con estas sustancias extremadamente peligrosas. En la figura 1.01 se puede observar uno de los químicos más tóxicos que se aplican a las rosas a fin de eliminar la plaga de los ácaros. Esta sustancia tiene un grado elevado de toxicidad, lo que quiere decir que su toxicología es de categoría IV.



Figura 1.01. Químico de toxicología IV.

A más de esto, dichos fungicidas tiene un alto nivel de concentración, por lo cual es necesario

disolverlos en grandes cantidades de agua para que su efecto sobre las plantas infestadas sea óptimo y no tenga consecuencias de ningún tipo. Un ejemplo de mezcla se observó el disolver una botella de fungicida de un litro, en una piscina de varios cientos de galones de agua. Esto se puede apreciar en la figura 1.02.

El suministro de estos fungicidas es determinado de acuerdo a la edad de las plantas y al estado en el que se encuentre la plaga en ellas. Para administrar esta solución a las plantas, se utiliza una bomba de 3HP, (figura 1.03) la cual provee la solución con una presión ideal, luego de lo cual es transportada por medio de una manguera de alrededor 130 m de largo como se muestra en la figura 1.04.



Figura 1.02. Químico categoría IV disuelto en agua.



Figura 1.03. Bomba distribuidora.



Figura 1.04. Manguera de transporte.

Al otro extremo de la manguera se conecta un aguilón, el cual es una barra larga de 1.45 m de longitud, en cuyo centro se encuentra una válvula de paso para controlar el flujo del fungicida. A más de esto, dicho aguilón está compuesto de 5 pitones a lo largo de su estructura, que están separados 0.33 m uno de otro (figura 1.05). Los pitones generan varios tipos de aspersión, de acuerdo a su altura, ya que cada parte de la planta debe tener un distinto tipo de aplicación de la solución. Un ejemplo de los pitones se observa en la figura 1.06.

La altura del aguilón, y la distancia de separación entre los pitones están determinadas por la altura promedio de una planta adulta que es alrededor de 2 metros de altura. En la parte superior de la planta se encuentra el botón de la rosa, por ello los distintos tipos de pitones que existen a lo largo del aguilón. En la figura 1.07. Se aprecia la altura de una planta de rosas roja adulta.



Figura 1.05. Aguilón de aspersión.



Figura 1.06. Pitón del aguilón.



Figura 1.07. Altura de una planta de rosas adulta.

Finalmente, se observó que cada invernadero posee varias decenas de camas de flores, las cuales estaban separadas unas de otras por aproximadamente 50 cm. En estas separaciones, se forman surcos en los cuales el terreno es generalmente tierra húmeda y blanda, con muchas deformidades en su superficie. En las figuras 1.08 se observa la separación entre camas así como el tipo de terreno que existe en ellas.



Figura 1.08. Separación entre camas.

Además de esto el largo de las camas depende del tipo de plantas que se encuentren en el invernadero, es decir, la longitud de las camas en los distintos invernaderos, varía entre los 20 m y los 38.5 m como se observa en la figura 1.09.



Figura 1.09. Longitud de camas de rosas.

Finalmente, el proceso de fumigación actualmente es realizado manualmente por operarios, los cuales deben sostener el aguilón a una distancia adecuada y tener sumo cuidado con la manguera de distribución ya que esta se puede enredar.

Pese a que usan protección, los exámenes toxicológicos que se realizan con mucha frecuencia, siempre denotan la presencia de cantidades mínimas de tóxicos en su sangre. En la figura 1.10 se aprecia el modo de fumigación que se ha venido realizando hace varios años.

Dado que un campo de rosas posee decenas de camas de varios metros de longitud, el presente proyecto será implementado en un campo de menor extensión. En vista de que se trata de un prototipo, la superficie considerada para las pruebas se muestra en la figura 1.11.



Figura 1.10. Fumigación manual de rosas.

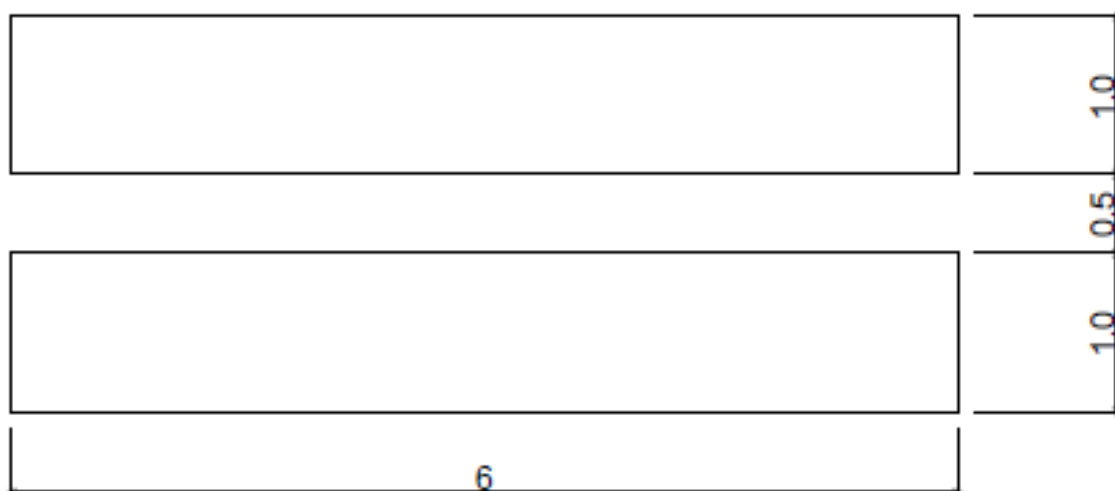


Figura 1.11. Área física para las pruebas del proyecto en metros.

En la Figura 1.11 apreciamos las medidas de las dos camas que constituirán el campo de pruebas del robot fumigador. Las condiciones de este campo serán ideales dado que no se pueden realizar pruebas en un campo de rosas real, porque eso implicaría poner en riesgo varios miles de dólares de producción para las florícolas.

1.4. APLICACIONES

Una vez realizado el estudio de campo, en el cual se observó todo el proceso de fumigación, se determinó las características específicas que debe tener el robot fumigador para que pueda cumplir a cabalidad la actividad para la cual va a ser diseñado.

Para las plantaciones de rosas pueden desarrollarse varias aplicaciones de acuerdo a las necesidades que aquí se presentan. Por ejemplo, se pueden diseñar robots móviles con tracción directa sobre el terreno, es decir, que sus llantas estén en sobre el piso. Por otro lado, también se podrían usar rieles de desplazamiento que guiarán al robot móvil a través de las camas de rosas para su fumigación.

Se puede usar la plataforma móvil a manera de vigía nocturno en la plantación, con el uso de cámaras inalámbricas para prevenir robos o estafas que puedan suscitarse en los invernaderos, o incluso para monitorear el estado de las plantas en los días posteriores a la fumigación donde el ambiente es demasiado toxico para un ser humano.

CAPÍTULO 2

DISEÑO DEL SISTEMA MECÁNICO

2.1. CONCEPTUALIZACIÓN

En el capítulo anterior se explicó las necesidades existentes en las plantaciones de rosas. Por lo cual, se busca implementar un robot que pueda realizar la fumigación de una manera más eficaz que la mano de obra humana. Con el fin de realizar el diseño, primero se estudiarán diferentes tipos de robots, y así implementar la más adecuada.

En el presente proyecto, dadas las condiciones explicadas en el Capítulo 1, se debe implementar un sistema robótico móvil con ruedas de tracción terrestres. Ya que un sistema guiado por rieles no sería viable en un campo de rosas, porque interferiría con la cosecha de las mismas. Por este motivo, es necesario diseñar una plataforma móvil que posea un sistema hidráulico de distribución y aspersión de fungicida hacia los rosales.

Existen robots móviles terrestres cuyas aplicaciones son de diversos tipos en la industria y el hogar. Algunas de estas son: tareas de estibación, rastreo y seguridad. A más de esto son populares las aplicaciones en tareas cotidianas como la limpieza del hogar. Se han logrado desarrollar robots autómatas que realizan la aspiración de suelos e identifican la suciedad para ser aspirada con el fin de ahorrar tiempo y esfuerzo a los habitantes del hogar.

CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

Un robot móvil puede contar con varias ruedas, y existe un punto alrededor del cual cada rueda sigue una trayectoria circular. A este punto se lo denomina CCI (Centro de Curvatura Instantáneo) en el caso de una trayectoria recta el CCI es infinito (Ruiz del Solar, Salazar, 2011).

Los robots móviles, pueden tener varias configuraciones en cuanto a la distribución de sus ruedas. Estas configuraciones, se las denomina arreglos de ruedas, y existen varios tipos de estas. Se analizará cada una de ellas para poder determinar cuál es la más efectiva para este proyecto (Ruiz del Solar, Salazar, 2011).

2.1.1. Arreglos de Ruedas.

El tipo de configuración de un robot depende de la aplicación a desarrollar. Sin embargo, los tipos más comunes son:

- Diferencial.
- Síncrono.
- Tipo triciclo.
- Tipo carro.

2.1.1.1. Arreglo Diferencial.

Este es uno de los esquemas más sencillos, básicamente consiste de dos ruedas con un eje común, donde cada rueda se controla independientemente. Además de esto sus movimientos son:

CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

- Línea Recta.
- En arco.
- Vuelta sobre su propio eje.

Este esquema utiliza una o dos ruedas adicionales para mantener el balance, estas distribuciones tienen diferentes nombres dependiendo de las ruedas. Por ejemplo: Con 3 ruedas se denominan triángulo, este esquema puede presentar problemas de estabilidad, y con 4 ruedas se conoce como diamante.

Para que el movimiento sea recto se requiere que las ruedas vayan a la misma velocidad, por lo cual se debe realizar el control de los mismos simultáneamente, esto se evidencia en la figura 2.01 (Ruiz del Solar, Salazar, 2011).

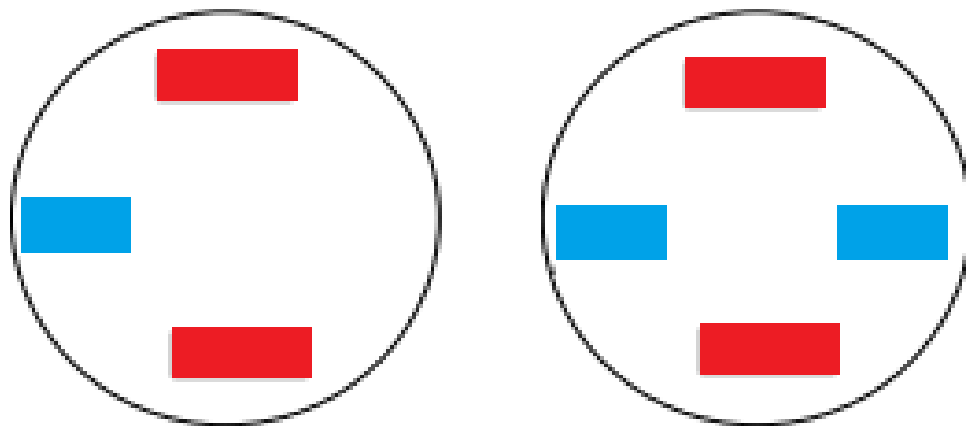


Figura 2.01. Arreglo diferencial de 3 y 4 ruedas (Ruiz del Solar, Salazar, 2011).

2.1.1.2. Arreglo Síncrono.

En este arreglo las ruedas se mueven en forma simultánea. El movimiento síncrono es un caso particular de la configuración diferencial, donde cada eje se mueve en forma dependiente para girar y avanzar. Las ruedas están ligadas de forma tal que siempre apuntan en la misma dirección, son estáticas en el eje perpendicular al movimiento. Por lo que, la dirección de la estructura se mantiene y requiere de un mecanismo adicional para mantener el frente del chasis (estructura del robot) en la dirección de las ruedas (Ruiz del Solar, Salazar, 2011).

En la figura 2.02 se aprecia este movimiento sobre el eje de rotación de las dos ruedas delanteras. Algunas ventajas de los sistemas Síncronos es que se evitan los problemas de inestabilidad, pérdida de contacto del diferencial y mayor complejidad mecánica (eje de giro).

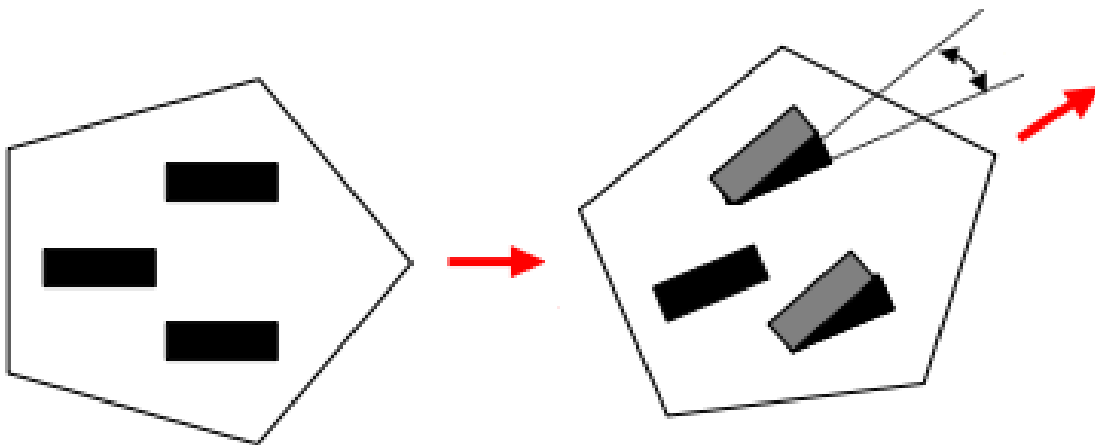


Figura 2.02. Arreglo síncrono de 3 ruedas (Ruiz del Solar, Salazar, 2011).

2.1.1.3. Arreglo Triciclo.

Los triciclos tienen dos ruedas fijas para tracción, además cuentan con una rueda de dirección que normalmente no tiene tracción. Estos sistemas tienen buena estabilidad y simplicidad mecánica, tiene facilidad para desplazarse en línea recta y su cinemática es más compleja, como se observa en la figura 2.03 (Ruiz del Solar, Salazar, 2011).

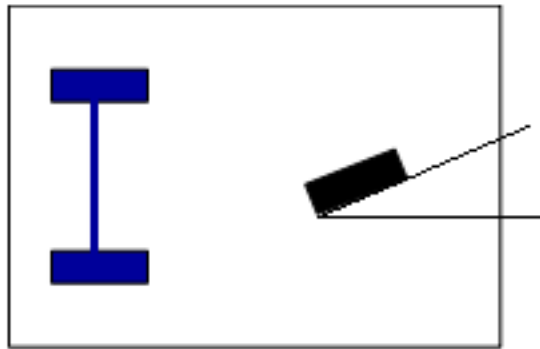


Figura 2.03. Arreglo Triciclo (Ruiz del Solar, Salazar, 2011).

2.1.1.4. Arreglo Carro.

Este tipo de arreglo es similar a la de un automóvil. Se diferencian del triciclo en que cuentan con dos ruedas de tracción y dos ruedas para dirección. Tiene una mayor complejidad mecánica que el triciclo por el acoplamiento entre las 2 ruedas de dirección. Sus principales ventajas son buena estabilidad y facilidad en la rectitud del desplazamiento. La desventaja es su complejidad cinemática, se la puede apreciar en a figura 2.04 (Ruiz del Solar, Salazar, 2011).

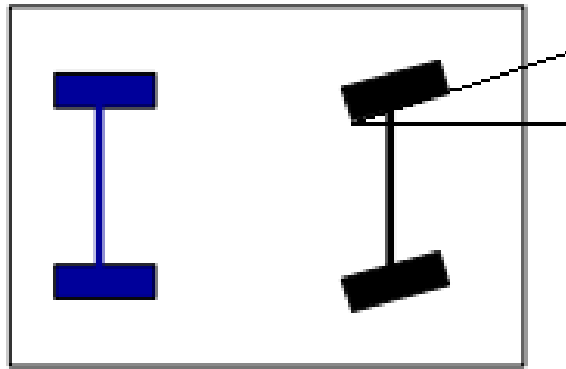


Figura 2.04. Arreglo Carro (Ruiz del Solar, Salazar, 2011).

2.1.1.5. Arreglo Omnidireccional.

Los sistemas omnidireccionales cuentan con 3 ruedas colocadas a 120° tal como se muestra en la figura 2.05. Donde cada una de las ruedas tiene la capacidad de girar en ambos lados y se logra un control lineal más simplificado que en el caso del robot diferencial. Actualmente los campeones mundiales en fútbol de robots F-180 utilizan ruedas omnidireccionales (Ruiz del Solar, Salazar, 2011).

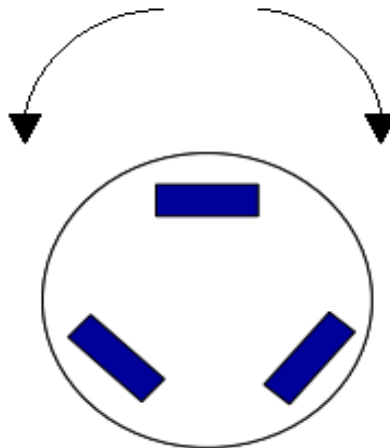


Figura 2.05. Arreglo Omnidireccional (Ruiz del Solar, Salazar, 2011).

CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

2.1.1.6. Selección del Arreglo más Adecuado.

Después de haber analizado los arreglos de ruedas más importantes se tomó la decisión de escoger una configuración similar al Arreglo Tipo Diferencial de Tres Ruedas(Figura 2.01), para la tercera rueda se usará una garrucha o rueda loca. En este punto hay que señalar que dadas las condiciones de peso que el sistema robótico debe soportar, la garrucha debe ser de tamaño considerable a fin de que soporte el peso del sistema sin afectar a la movilidad y estabilidad del mismo. Otro detalle a tomar en cuenta es el radio de giro que debe tener el robot. Este tiene que ser lo más cerrado posible, toda vez que la superficie donde realizará el giro es escasa. El giro que este tendrá que realizar es de casi 90 grados, y por tanto es esencial que gire sobre su propio eje. Esta es una ventaja que el mencionado arreglo posee.

2.1.2. Distribución del Espacio Físico.

El presente diseño tiene varias etapas a considerar para la distribución del espacio físico. Estas etapas son: motores de propulsión, garrucha de dirección, contenedor de fungicida, alimentación eléctrica, sistema electrónico y sistema hidráulico de distribución-aspersión. Por ello se considera hacer un sistema vertical con varios pisos, en donde se ubicarán las etapas antes mencionadas. Tomando en cuenta las características de cada una de ellas.

2.1.2.1. Motores de Propulsión.

Estos dispositivos deben ir colocados bajo la base del prototipo, porque serán los que propulsen las ruedas, que a su vez aplicaran la tracción necesaria para mover todo el sistema. Como se mencionó anteriormente, estos serán ubicados de acuerdo al arreglo diferencial de tres ruedas.

CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

2.1.2.2. Garrucha de Dirección.

Este elemento será colocado en la mitad del extremo opuesto a los motores, con el fin de dar más soporte a toda la estructura metálica. Además, en esta posición se logra tener una mayor estabilidad al momento de realizar el giro.

2.1.2.3. Contenedor de Fungicida.

Este dispositivo será un recipiente que llevará en su interior el líquido fungicida. Por ello, sus dimensiones deben ser proporcionales al área de cobertura del prototipo. Esto significa que el volumen del recipiente debe ser suficiente para fumigar el área de trabajo mostrada en la figura 1.11. Las dimensiones que se consideran adecuadas para el recipiente son de: 0.30x0.36x0.25 m. toda vez que el volumen de líquido necesario se ha estimado en aproximadamente 20 litros.

Este contenedor para evitar el excesivo movimiento de líquido, y un posible contacto de este con las etapas eléctricas, se considera ubicarlo en el primer piso del prototipo.

2.1.2.4. Alimentación Eléctrica.

Esta etapa constituye la fuente de energía eléctrica que alimentará a los dispositivos eléctricos y electrónicos del sistema. Este estará constituido de baterías de corriente continua, las cuales son de gran peso y se deben colocar en el centro del segundo piso, con el objetivo de balancear el peso total de la plataforma.

CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

2.1.2.5. Sistema Electrónico.

Este sistema irá en la parte superior, sobre las baterías, para optimizar el cableado de alimentación y de control con las distintas etapas. Esta etapa estará aislada del exterior por una cubierta metálica con puerta móvil, de manera que se pueda tener fácil acceso a la circuitería. En el sistema electrónico también se consideran los sensores para el modo automático, estos serán colocados en el constado del segundo piso de la plataforma.

2.1.2.6. Sistema Hidráulico de Distribución-Aspersión.

Este grupo de elementos serán los responsables de transportar el fungicida desde su contenedor hacia las plantas a ser fumigadas. Estará conformado por mangueras de distribución, electroválvulas de control y aspersores.

Las mangueras serán colocadas de manera que no compliquen el acceso a otros dispositivos del sistema. Es decir, irán juntas a los ángulos que sostendrán los distintos pisos. Las válvulas de control se colocarán en la parte superior de la plataforma para lograr un aspecto estético y funcional. Los aspersores deben ir colocados en el último piso del robot uno a cada lado para que logren fumigar las plantas.

2.2. SÍNTESIS

Al presente sistema robótico se lo denominará MI-T-NATOR y en la presente sección se enfocará en dar una idea generalizada acerca del sistema mecánico que deberá tener para

CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

desplazarse sobre el terreno de fumigación. De tal forma que provea la suficiente estabilidad para evitar fallos al momento de aplicar presión hidráulica para la aspersión.

El terreno es irregular, por lo cual se colocará un sistema de tracción de dos ruedas, cada una de ellas impulsada por su respectivo motor eléctrico y una caja de reducción que aumente el torque. Con ello, se logrará tener una mayor estabilidad y fuerza para evitar posibles obstáculos que se presenten en el terreno de desplazamiento.

Las pruebas se realizarán sobre condiciones ideales, dado que el presente proyecto se trata de un prototipo para la aplicación agrícola mencionada. No es aconsejable realizar pruebas en un campo real por los altos costos que tiene la producción de rosas, y los riesgos que implicaría realizarlos en una situación real.

La morfología ideal para la aplicación sería la cartesiana, debido a que la fumigación solo requiere un movimiento en el eje x, pero debido a que esto implicaría colocar guías en el terreno, lo cual no es factible, se ha decidió usar una configuración de dos ruedas móviles mencionada en la sección anterior. Dado que los componentes del sistema son varios, se decidió implementar una morfología escalonada de pisos verticales, y de esta manera lograr una distribución adecuada y eficiente, a más de brindar estabilidad al robot.

2.3. ANÁLISIS

Este proyecto se inicia con la idea de solventar ciertos problemas que se presentan en las plantaciones de rosas. Y en las secciones anteriores de este capítulo, se hizo una introducción al sistema mecánico a diseñarse. A partir de aquí se realizará un análisis de todos los componentes mecánicos de MI-T-NATOR.

CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

La parte esencial del sistema mecánico es lograr la movilidad del mismo. Por ello se realiza el análisis de la tracción necesaria para lograrlo. Dado que se requiere acoplar las ruedas de tracción con los motores propulsores, se debe diseñar un sistema de acoplamiento entre ambos.

Las ruedas de tracción deben ser de un diámetro de aproximadamente 0.2 metros, para lograr elevar la plataforma. La Figura 2.06 muestra una representación de las llantas requeridas y su respectivo plano de construcción en la figura 2.07.



Figura 2.06. Ruedas de Tracción.

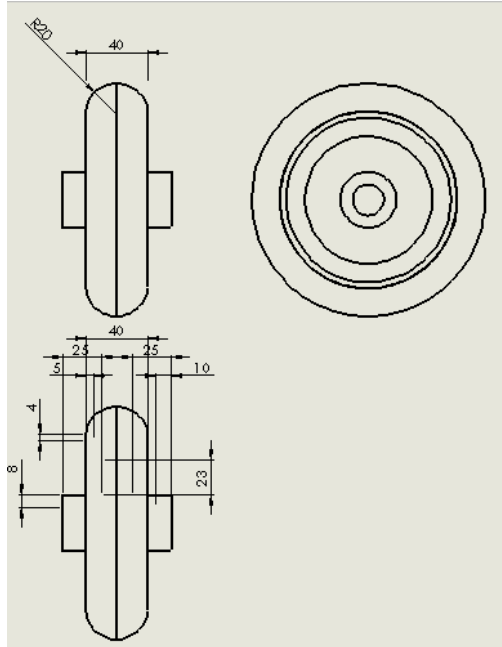


Figura 2.07. Plano Ruedas de Tracción (mm).

Además de esto se usará una garrucha que será la encargada de dar la dirección de giro. Esto basado en el movimiento que los motores tengan en determinado instante. Esta llanta permite que el sistema mecánico incluso gire sobre su propio eje. En la figura 2.08, se observa una representación de la garrucha, y en la figura 2.09 su plano de construcción.

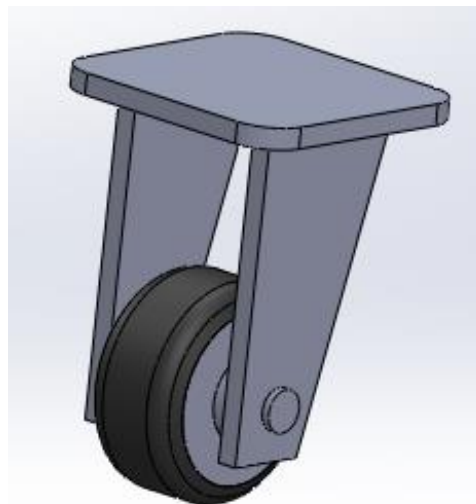


Figura 2.08. Garrucha de Dirección.

CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

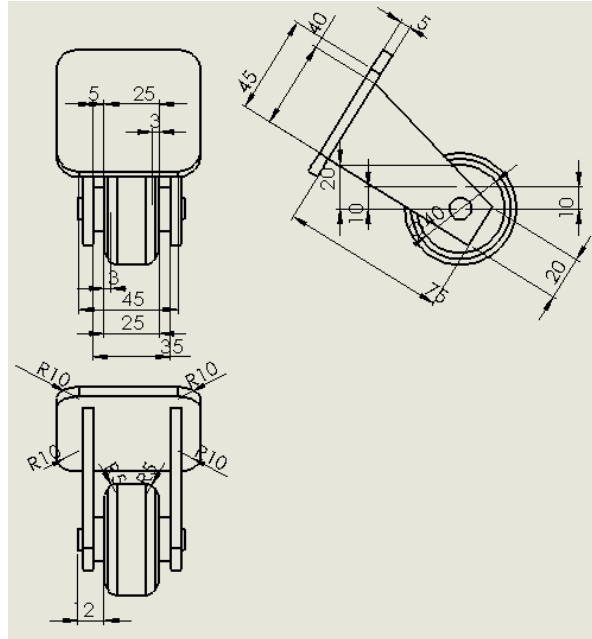


Figura 2.09. Plano Garrucha de Dirección (mm).

El acoplamiento entre la llanta y el motor requiere tener una suficiente sujeción para no tener pérdidas de potencia al momento de realizar la transferencia de movimiento desde el motor hacia las llantas. Se utilizará un sistema de acople metálico sólido mediante un prisionero que sujete los ejes de los motores a la rueda. Esto se aprecia en las figuras 2.10 y 2.11.

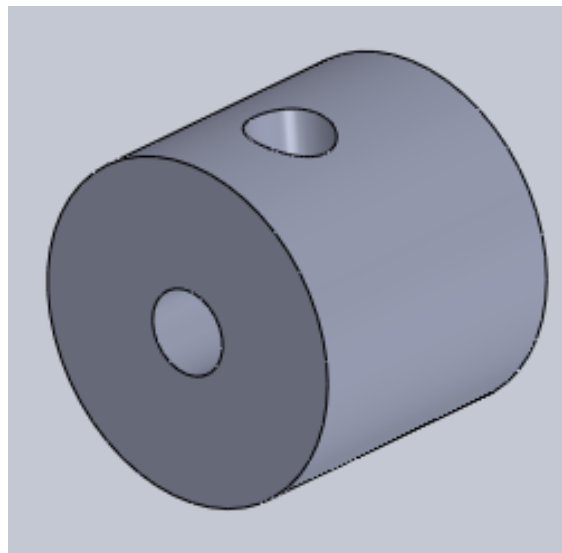


Figura 2.10. Acople motor-rueda.

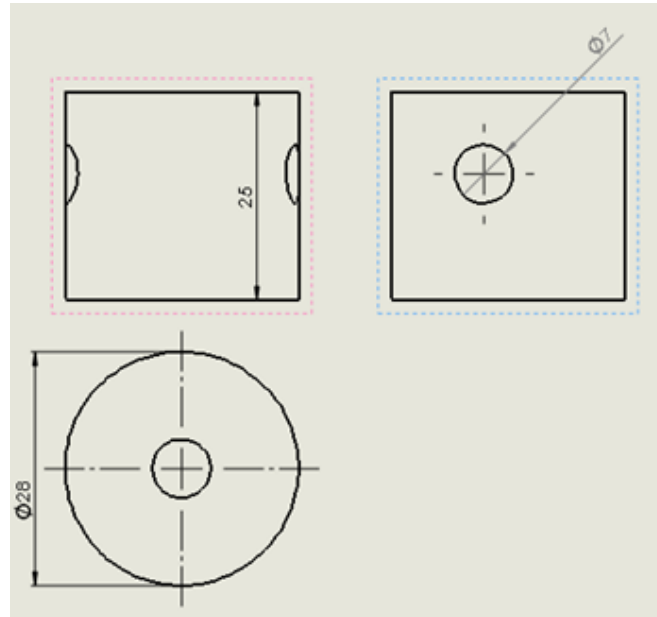


Figura 2.11. Plano Acople motor-rueda (mm).

Los motores deben ir sujetos a la base, por lo que se usarán abrazaderas metálicas ajustables para retirarlos cuando necesiten mantenimiento. El ajuste de estos elementos se realizará mediante un sistema de tuerca tornillo como se muestra en la figura 2.12.

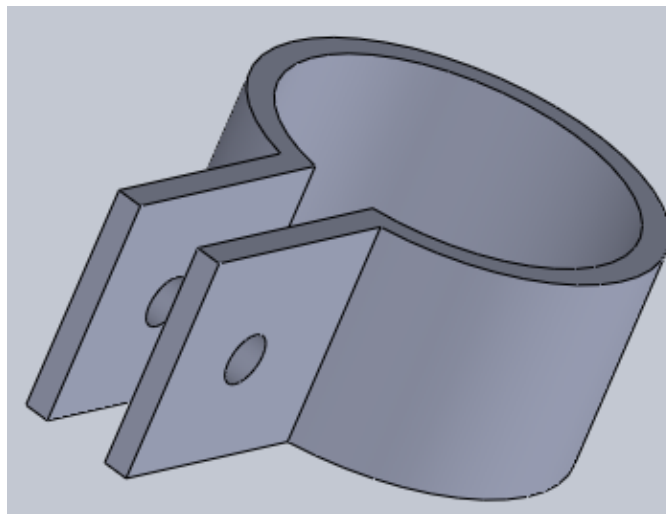


Figura 2.12. Abrazadera para motor.

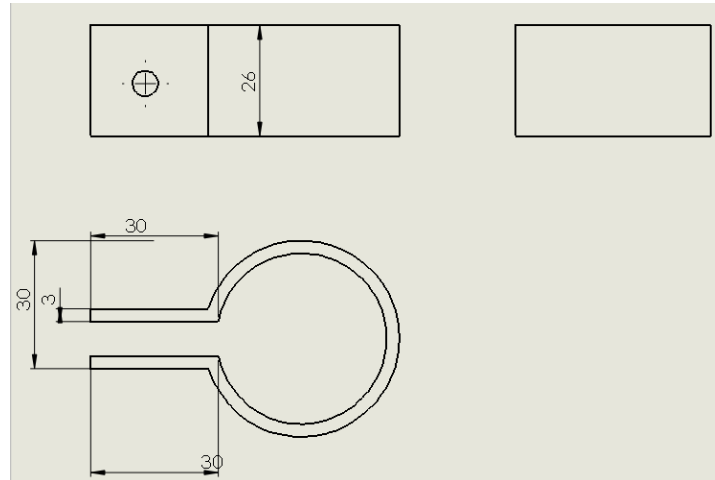
CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

Figura 2.13. Plano Abrazadera para motor (mm).

Para sujetar las baterías de alimentación eléctrica se usarán ángulos a 90 grados en cada vértice de ellas. Cada uno de ellos tendrá una altura adecuada con el objetivo de apretar las baterías, pero que permitan retirarlas para cargarlas. En la figura 2.14 se muestra una representación de dichos ángulos y sus planos en la figura 2.15.

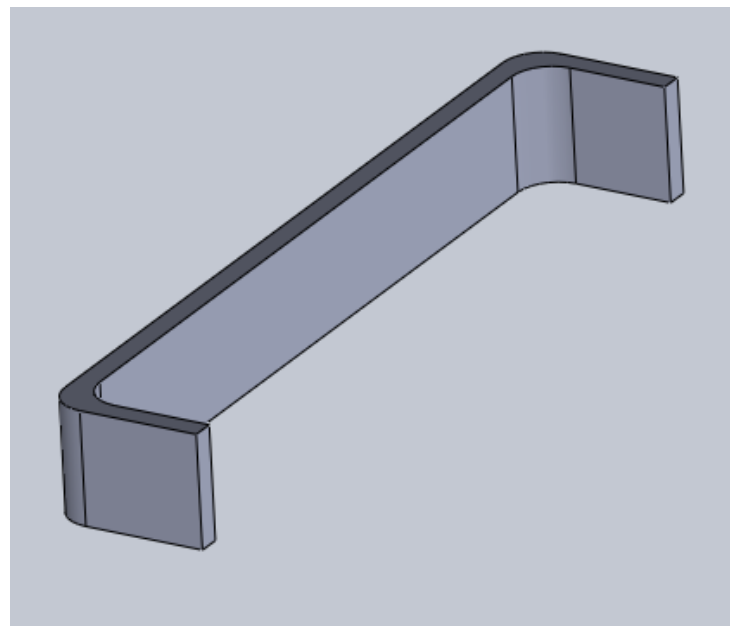


Figura 2.14. Ángulos Sujetadores de Baterías.

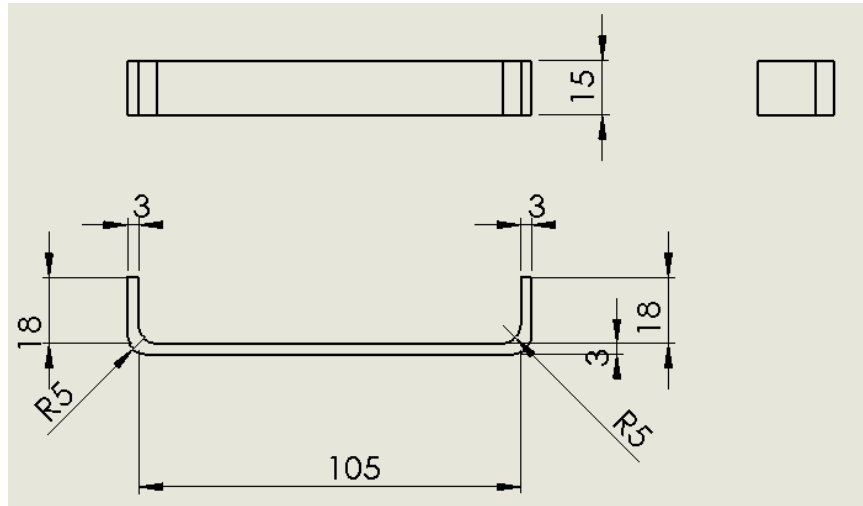


Figura 2.15. Plano Ángulos Sujetadores de Baterías (mm).

Los aspersores serán sujetos mediante un soporte metálico soldado en la mitad de cada lado en la parte superior de la plataforma. Estos soportes estarán ubicados de tal forma que los aspersores rocíen adecuadamente a las rosas sin perder eficiencia. En las figuras 2.16 y 1.17 se evidencia la representación y vistas de esta pieza.

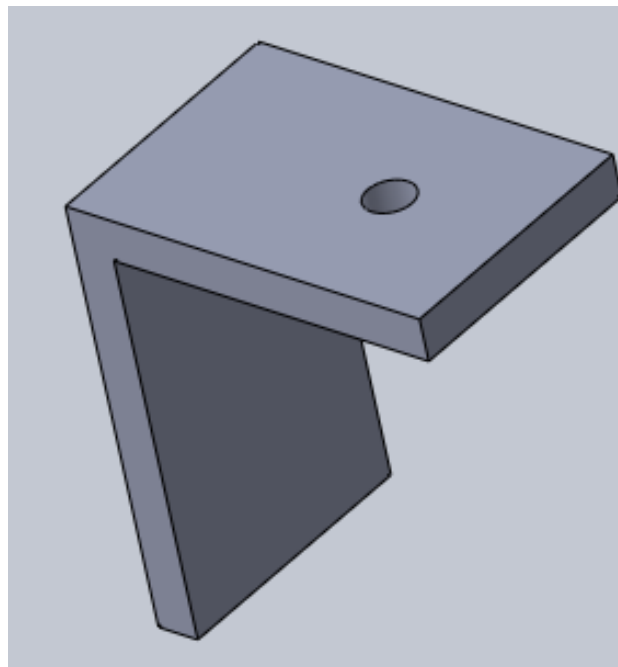


Figura 2.16. Soporte de Aspersores.

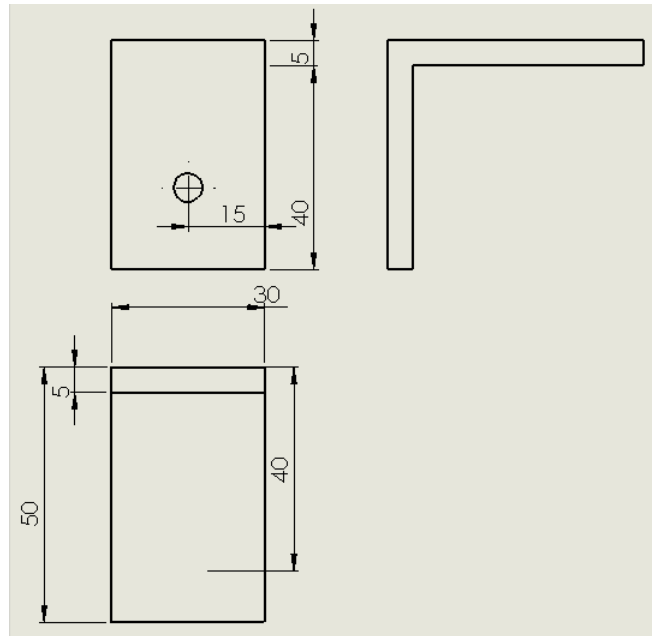


Figura 2.17. Plano Soporte de Aspersores.

La plataforma móvil contará con tres pisos, los cuales serán soldados a ángulos metálicos verticales, estos constituirán las columnas de soporte de las plataformas. Cabe mencionar que el último piso actuará como aislante del segundo piso. Toda vez que el segundo piso debe llevar los dispositivos eléctricos y electrónicos. En la figura 2.18 se puede observar la representación de las plataformas, y en la figura 2.19 su plano de construcción.

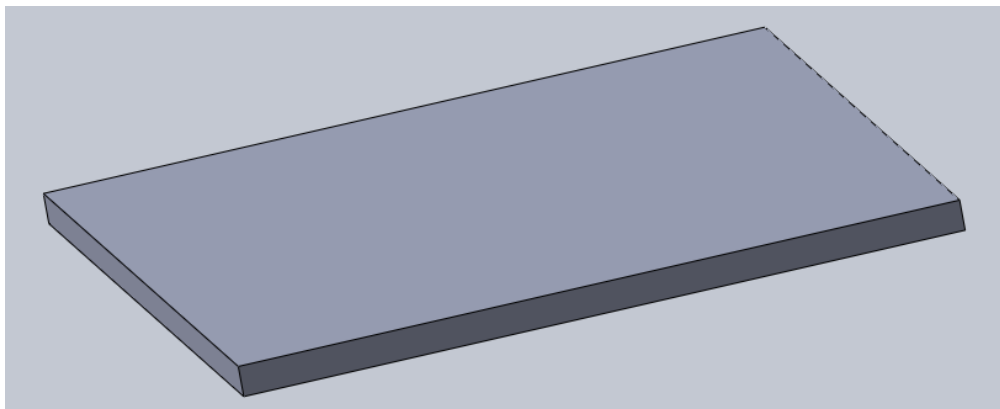


Figura 2.18. Plataforma.

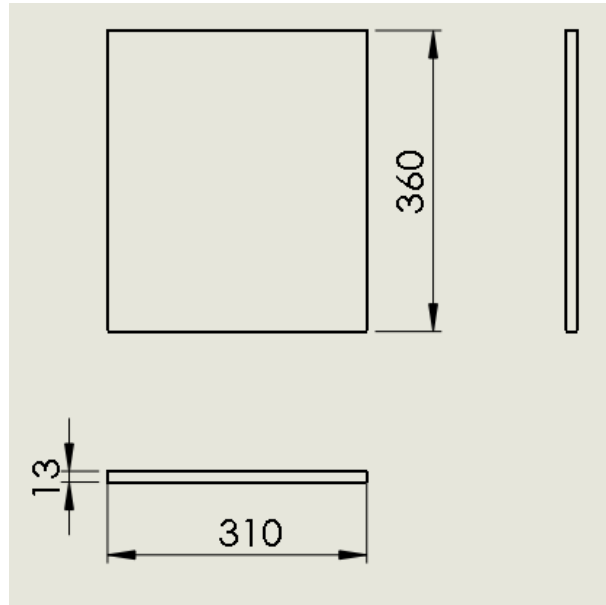


Figura 2.19. Plano Plataforma (mm).

Como se ha mencionado con anterioridad, el sistema mecánico de MI-T-NATOR, requiere de soportes que fijen a una altura determinada cada plataforma. Estos dispositivos serán ángulos metálicos, constituyéndose en columnas de sostén de todos los dispositivos. En la figura 2.20 y 2.21 se aprecian tanto la representación de los ángulos, como sus planos de construcción.

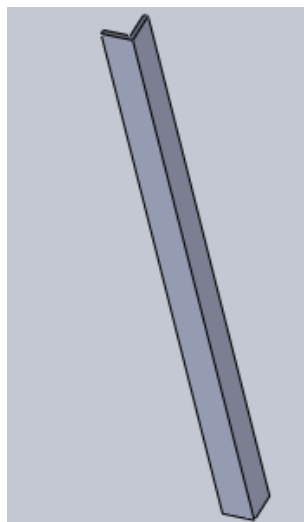


Figura 2.20. Ángulos de soporte.

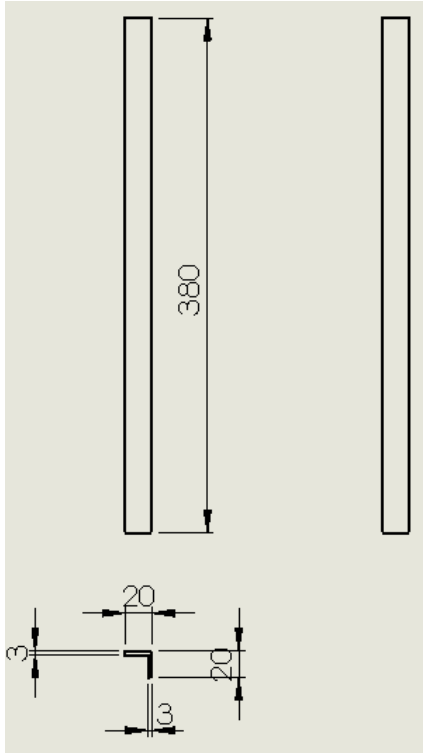
CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

Figura 2.21. Plano Ángulos de soporte (mm).

Con el fin de aislar los elementos electrónicos del exterior, y evitar que el fungicida tenga contacto con los mismos, se considera adecuado encerrar a estos elementos dentro de una caja hermética de metal, con un único acceso a través de la parte superior. En la figura 2.11 se observa la representación de la caja aislante, y en la figura 2.23 el plano de construcción de esta.

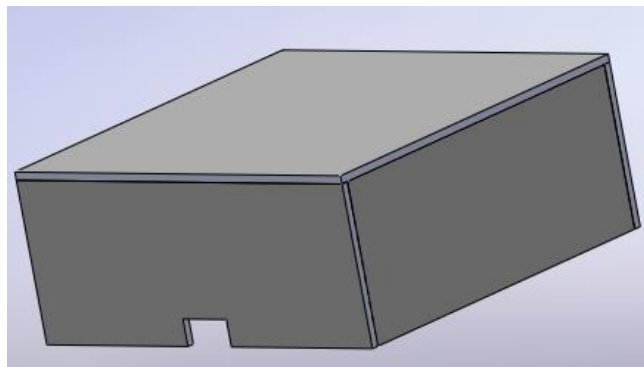


Figura 2.22. Caja Aislante.

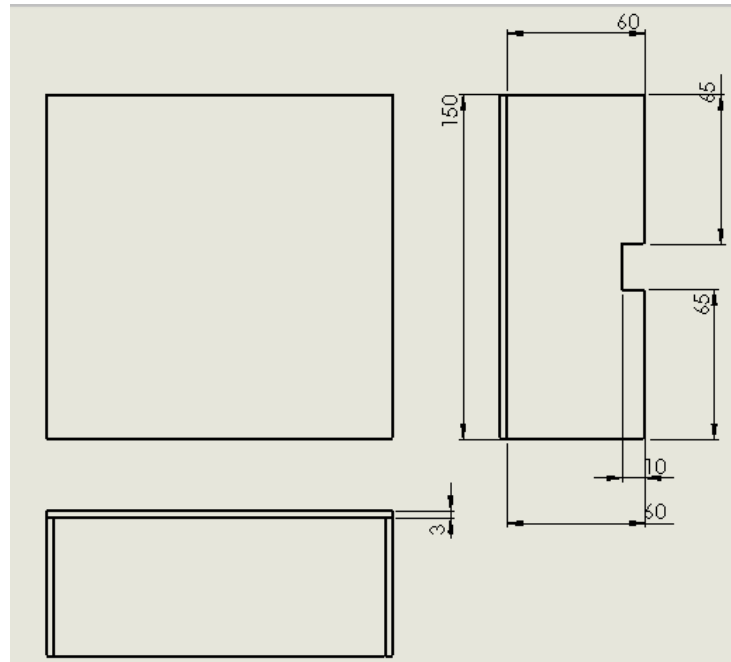


Figura 2.22. Plano Caja Aislante (mm).

2.4. EVALUACIÓN

De acuerdo a las investigaciones y análisis que se han realizado para determinar el diseño mecánico más óptimo de MI-T-NATOR, se ha escogido un diseño por pisos y un arreglo de ruedas tipo triciclo con garrucha de dirección. Con esto se logra que la aplicación sea lo más versátil posible, ya que la distribución de peso y la utilización del espacio físico no afectan la movilidad del robot. Además, el ángulo de giro que constituye un factor crítico, se soluciona por las características de la configuración tipo triciclo.

Esto quiere decir, que el MI-T-NATOR, es capaz de girar sobre su propio eje si así lo requiriera, lo que es óptimo dado el reducido espacio que va a tener para realizar este movimiento.

CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

Otro factor fundamental es la capacidad de realizar los movimientos hacia atrás y adelante lo más recto posible, y ya que se usan motores de características similares a ambos lados, no se genera una desviación crítica en estos movimientos.

2.4.1. Ecuaciones Para La Cinemática Diferencial.

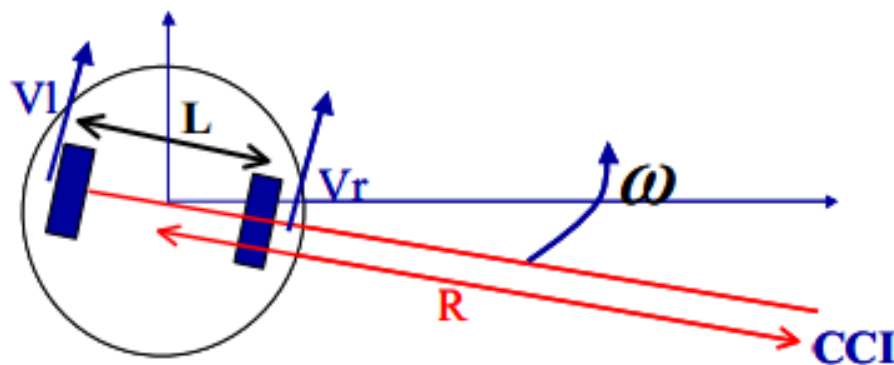


Figura 2.23. Representación Cinemática Arreglo Diferencial (Ruiz del Solar, Salazar, 2011).

Dónde:

ω : Velocidad Angular.

R: Radio de Giro.

L: Distancia entre las Ruedas.

V_r : Velocidad Rueda Derecha.

V_l : Velocidad Rueda Izquierda.

Formula del radio de curvatura:

$$R = \frac{L}{2} * \left(\frac{V_r + V_l}{V_r - V_l} \right)$$

Ecuación 2.1. Radio de Curvatura Instantánea.

CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

Formula de velocidad angular:

$$\omega = \frac{Vr - Vl}{L}$$

Ecuación 2.2. Velocidad Angular.

Datos del robot:

L: 42cm

Giro hacia la Izquierda:

R estimado 15cm

Velocidad rueda derecha estimada es 0,39 m/s

Despejando velocidad rueda izquierda de la ecuación 2.1:

$$Vl = \frac{Vr(L - 2R)}{2R + L}$$

Ecuación 2.3. Velocidad Rueda Izquierda.

Reemplazando valores en la Ecuación 2.3:

$$Vl = \frac{0,39(0,42 - 2 * 0,15)}{0,42 + 2 * 0,15}$$

$$Vl = 0,065m/s$$

CAPÍTULO 2: DISEÑO DEL SISTEMA MECÁNICO

Giro hacia la derecha:

Radio estimado 15 cm.

Velocidad rueda izquierda estimada es 0,39 m/s.

Despejando de la ecuación 2.1:

$$Vr = \frac{Vl(L - 2R)}{2R + L}$$

Ecuación 2.4. Velocidad Rueda Derecha.

Remplazando valores en ecuación 2.3:

$$Vr = \frac{0,39(0,42 - 2 * 0,15)}{2 * 0,15 + 0,42}$$

$$\mathbf{Vr = 0,078m/s}$$

2.5. REPRESENTACIÓN

Una vez realizado todo el análisis requerido para el diseño mecánico de MI-T-NATOR, se procede a mostrar una representación virtual en tres dimensiones de lo que constituirá todos los parámetros mencionados en el presente capítulo. A continuación se muestra la representación total en tres dimensiones del sistema mecánico. En la figura 2.14 se aprecia una representación de la plataforma.

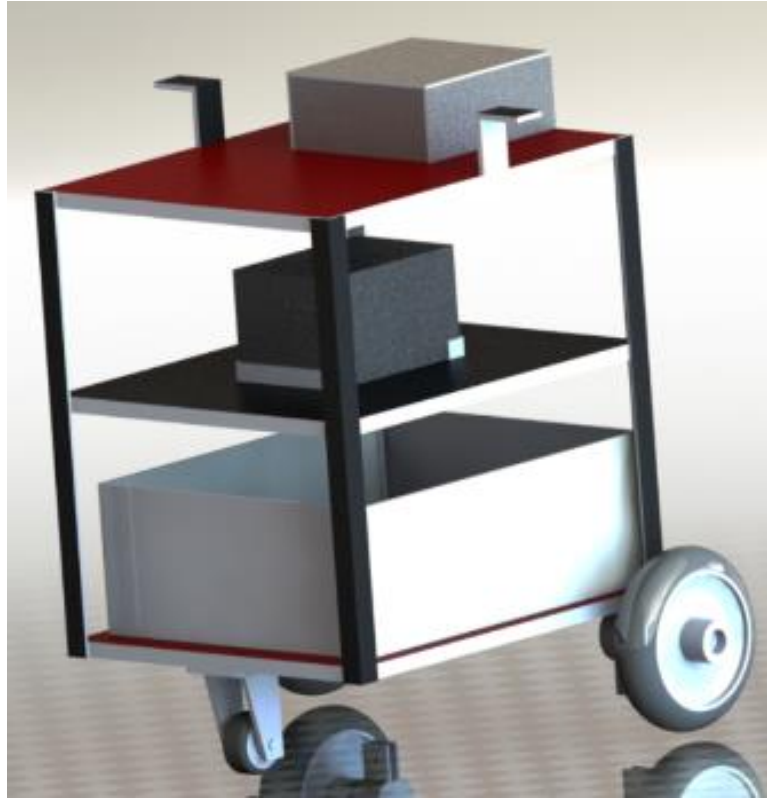


Figura 2.14. Plataforma.

CAPÍTULO 3

DISEÑO DE HARDWARE

3.1. ETAPA DE POTENCIA

3.1.1. Movilidad.

Para saber qué tipo de motor es el más adecuado, se estudió los tipos de motores de corriente continua existentes y que se muestran a continuación. Se los encuentra de distinto tamaño, forma y potencia, pero todos se basan en el mismo principio de funcionamiento. Para la información que se encuentra a continuación, se realizó una investigación en línea, de manera que nos permita seleccionar el tipo de motor más adecuado para la aplicación.

3.1.1.1. Motor de Corriente Continua

Accionar un motor DC es muy simple y solo es necesario aplicar tensión de alimentación entre sus bornes. Para invertir el sentido de giro basta con intercambiar la alimentación y el motor comenzará a girar en sentido opuesto (Hernández, 2012).

A diferencia de los motores a pasos y los servomecanismos, los motores DC no pueden ser posicionados y/o enclavados en una posición específica. Estos simplemente giran a

CAPÍTULO 3: DISEÑO DE HARDWARE

la máxima velocidad y en el sentido que la alimentación aplicada se lo permite (Hernández, 2012).

En el rotor (parte móvil) y el estator (parte fija), se ubican los demás componentes como:

- Escobillas y porta escobillas.
- Colector.
- Eje.
- Núcleo y devanado del rotor.
- Imán Permanente.
- Armazón.
- Tapas o campanas.

Moto-Reductor: Los Reductores y los Moto-reductores son elementos mecánicos muy adecuados para el accionamiento de todo tipo de máquinas y aparatos de uso industrial que necesiten reducir su velocidad de una forma eficiente, constante y segura (Hernández, 2012).

Las ventajas de usar Reductores y/o Moto-reductores son:

- Alta eficiencia de la transmisión de potencia del motor.
- Alta regularidad en cuanto a potencia y par transmitidos.
- Poco espacio para el mecanismo.
- Poco tiempo de instalación y mantenimiento.
- Elemento seguro en todos los aspectos, muy protegido.

CAPÍTULO 3: DISEÑO DE HARDWARE



Con estos datos, se escogió motores de corriente continua que tengan incorporado cajas reductoras para así aprovechar las ventajas mecánicas que se han mencionado con anterioridad.

Para elegir el motor adecuado se estudió algunos tipos de motores de corriente continua (tabla 3.01) para ver el más óptimo y de fácil acceso para la aplicación.

Tabla 3.01. Descripción de Motores de CC.

Tipo	RPM	Torque	Voltaje de operación	Imagen
Servo motor(truncado)MG996R	78	12kg/cm	5v	
Motor con moto-reducción 67:1	150	14kg/cm	12v	
Motor con moto-reducción 50:1	180	12kg/cm	12v	

CAPÍTULO 3: DISEÑO DE HARDWARE

Motor con moto-reducción 29:1	200	9kg/cm	12v	
Motor con moto-reducción AME218	50	30kg/cm	12v	

Considerando la velocidad y el torque se decidió usar un motor con moto-reducción de 67:1 ya que tiene una velocidad y un torque adecuado para la presente aplicación.

Circuito Electrónico Para Motores:

Para manipular los motores se decidió usar un integrado de tipo puente H, ya que los motores deben de ser capaces de girar de manera horaria y anti horaria, para esto se eligió el integrado L298 debido a su buena corriente de salida y su capacidad de manejar voltajes TTL. A continuación algunas características técnicas de dicho circuito integrado:

- Corriente máxima: 2 Amperios por salida (4A en cascada).
- Alimentación motores: 5-46V DC.
- Corriente para lógica: ~36 mA.
- Niveles TTL: Bajo -0.3V~1.5V, alto: 2.3V~Vss.
- Potencia máxima: 25W (75 Grados Celsius).
- Temperatura de funcionamiento: -25C~+130 C.

En la figura 3.01, se muestra el circuito electrónico del puente H y las referencias de los puertos a los distintos circuitos.

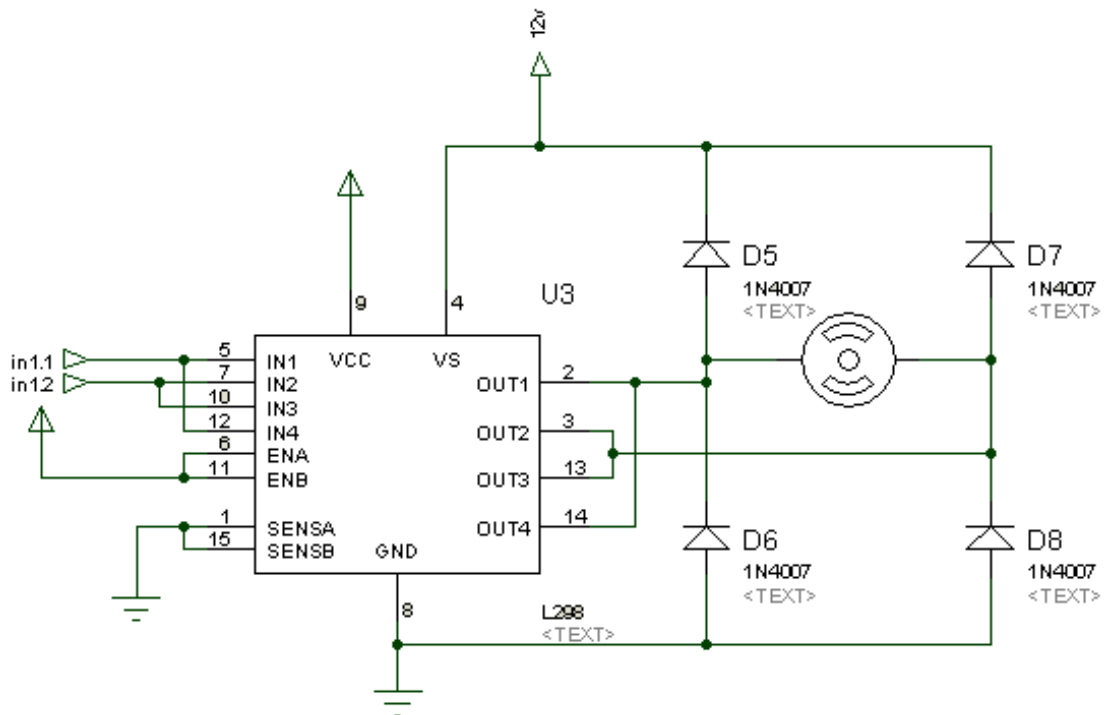


Figura 3.01. Conexión e cascada del Puente H.

3.1.2. Control De Fumigación

Para el control de fumigación se consideró el uso de tres elementos:

- Bomba
- Electro válvulas
- Elementos electrónicos de potencia

3.1.2.1. Bomba

Las bombas son máquinas capaces de proporcionar energía a los fluidos. Estos se mueven en respuesta a un gradiente de energía, y el movimiento provoca el decremento de la energía total. Las bombas están conformadas, en general, por las siguientes partes principales (Blacio, 2009):

- **Cuerpo:** es la coraza externa o caja de la bomba.
- **Impulsor:** el impulsor se encarga de dar movimiento al agua cuando entra a la bomba; consta de palas al igual que una hélice.
- **Eje** el eje de la bomba sostiene y da movimiento al impulsor.
- **Rodamientos** o rulimanes, son sellados o lubricados con agua, y sostienen al eje, el cual debe estar cuidadosamente centrado.
- **Válvula de purga:** es una válvula de tamaño reducido que permite cebar la bomba antes de la operación, si no está provista de válvula cheque.

En el mercado existe gran cantidad de bombas disponibles, cada una diseñada para una aplicación o para un grupo de aplicaciones. Si en acuicultura se necesitan bombas para mover agua dulce, salobre o salada, las más utilizadas son las centrífugas, y en parte, las

CAPÍTULO 3: DISEÑO DE HARDWARE

rotativas. A continuación se presenta información en la cual se hace referencia a la clasificación general de las bombas, con los tipos más importantes (Blacio, 2009):

Bombas CENTRÍFUGAS:

- De Voluta
- Difusoras
- Turbinas para pozos profundos.

Bombas ROTATIVAS:

- De hélice
- De turbina regenerativa
- De aletas deslizantes
- De engranajes
- De lóbulos
- De tubo flexible.

Bombas ALTERNATIVAS.

Bombas ELEVADORAS DE AIRE.

Bombas Centrífugas

Aproximadamente el 90 % de todas las bombas existentes en acuicultura son del tipo Centrífugo. En ellas, la unidad de potencia (motor) mueve el eje de la bomba, haciendo que el

CAPÍTULO 3: DISEÑO DE HARDWARE

impulsor rote. Las aspas del impulsor, con el movimiento de rotación, dirigen cualquier partícula que esté sobre ellas hacia afuera, como resultado de la fuerza centrífuga (Blacio, 2009).

Si la entrada y el cuerpo (cavidad) de la bomba están llenos de agua, el movimiento hacia afuera del agua que está sobre el impulsor disminuye la presión en el centro del mismo, y esto causa que se succione más agua hacia el interior de la bomba para compensar tal pérdida de presión (Blacio, 2009).

Para esta aplicación se usó una bomba centrífuga sumergible, la cual succionara el líquido del contenedor para posteriormente llevarla a los aspersores lo cuales fumigarán a la planta.

En esta aplicación se eligió una bomba sumergible de las siguientes características:

- Presión hasta 70 PSI.
- Alta durabilidad.
- Bajo ruido, alto aislamiento contra interferencias.
- Garantía de flujo constante.
- Perfecto funcionamiento durante el suministro a alta temperatura.
- 2 litros por minuto.

Circuito Electrónico de Salida

CAPÍTULO 3: DISEÑO DE HARDWARE

El circuito utilizado para el encendido y apagado de la bomba centrífuga es un transistor en modo emisor común, el cual funciona en las zonas de corte y saturación haciendo que la bomba se active o desactive. El circuito utilizado se muestra en la figura 3.02.

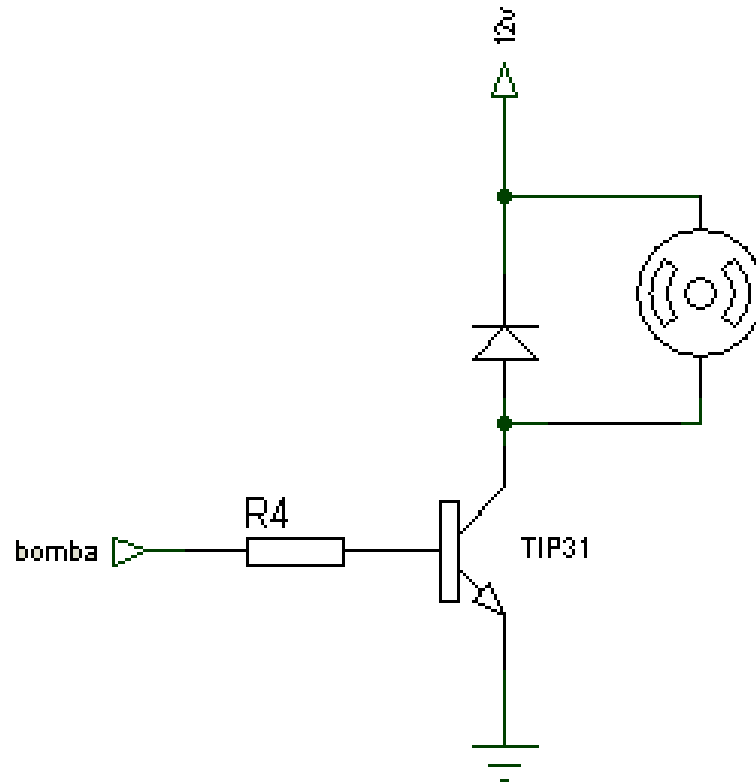


Figura 3.02. Circuito Electrónico de Activación.

3.1.2.2. Electroválvulas.

En muchas aplicaciones es necesario controlar el paso de algún tipo de flujo, desde corriente eléctrica hasta gases o líquidos. Esta tarea es realizada por válvulas. En particular, las accionadas por solenoides permiten su implementación en lugares de difícil acceso y facilitan la automatización del proceso al ser accionadas eléctricamente.

Selección De Una Válvula De Solenoide

Al momento de elegir una válvula, debemos tomar en cuenta los siguientes puntos:

- 1.- Fluido a controlar (características de viscosidad y pH)
- 2.- Rangos de Presión con los que se desea trabajar (MOPD y MinOPD)
- 3.- Temperatura del fluido (y del medio externo).
- 4.- Tipo de cañería o conexión (tamaño y estilo).
- 5.- Características eléctricas de la válvula (rangos de voltaje con que trabaja; DC o AC).
- 6.- Opciones específicas para la aplicación:

- Normalmente abierta o cerrada.
- Presión segura de trabajo (SWP)
- Con o sin vástago manual, etc.

Cabe destacar que, al momento de dimensionar la válvula, se debe tener en cuenta la cantidad de flujo a controlar, más que el ancho de la tubería. En este sentido, es recomendado consultar las especificaciones del fabricante para cada válvula. Además es importante tomar en cuenta el tipo de líquido con el que se trabaja, de manera que los materiales de construcción de las válvulas sean compatibles con cada aplicación (Carvallo, Vargas, 2003).

Desde luego, es necesario tener en cuenta el tipo de alimentación con que trabaja la válvula (Rango de Voltaje y Frecuencia). No cumplir con estos rangos puede causar daño permanente en la bobina inutilizando la válvula.

CAPÍTULO 3: DISEÑO DE HARDWARE

En este sentido, a modo de sugerencia, se recuerda que no es conveniente polarizar la bobina del solenoide al mismo tiempo que se conecta a la alimentación un motor de gran potencia (bobina) dado que este causa variaciones importantes en el nivel de voltaje aplicado al solenoide (Carvalho, Vargas, 2003).

Sobre la instalación de la válvula, cabe recordar que la mayoría está diseñada para trabajar en líneas horizontales, con la bobina en la parte superior. Existen válvulas, equipadas con resortes, que permiten su instalación en cualquier posición.

Otra consideración relevante es que se debe respetar la dirección del fluido en la instalación. Este debe coincidir con la flecha que generalmente se encuentra en el cuerpo de la válvula. También, es conveniente instalar un filtro antes de la válvula, con el objetivo de filtrar partículas extrañas (Carvalho, Vargas, 2003).

Elección de la Electroválvula.

Después de un estudio sobre las diferentes electroválvulas y su disponibilidad en el mercado local, se decidió usar una electroválvula con las siguientes características:

- Fluido a controlar: Agua.
- Rangos de Presión con los que se desea trabajar: 60-80 PSI.
- Temperatura del fluido: 10 grados Celsius.
- Tipo de cañería: Manguera blanda ¼ de pulgada de diámetro.
- Características eléctricas de la válvula: 12 V DC.
- Opciones específicas para la aplicación:
- Normalmente cerrada.

CAPÍTULO 3: DISEÑO DE HARDWARE

- Presión segura de trabajo 80 PSI.
- Sin vástago manual.
- Apertura (en línea)/cierre (perpendicular).
- Corriente consumida 400mA.
- IP65.

Las válvulas comerciales seleccionadas, son válvulas genéricas importadas directamente, y fueron seleccionadas por cumplir la mayoría de especificaciones que se han mostrado anteriormente.

Circuito Activación Electroválvula.

El circuito utilizado para el encendido de la electroválvula fue un transistor en modo emisor común, el cual funciona en las zonas de corte y saturación haciendo que la bomba se active o desactive. El circuito utilizado se muestra en la figura 3.03:

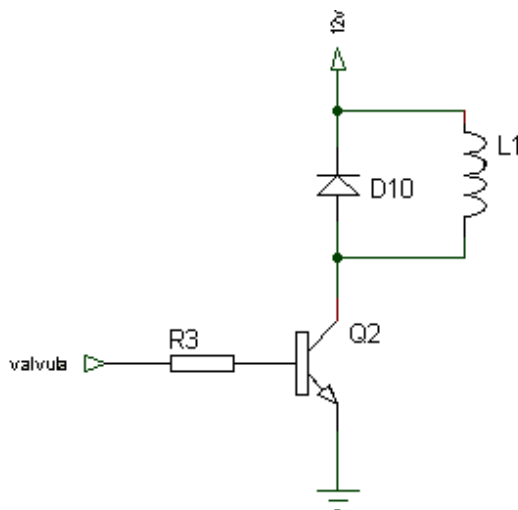


Figura 3.03. Circuito Electrónico de Activación Electroválvula.

3.2. ETAPA DE COMUNICACIÓN DE DATOS

3.2.1. Módulos Xbee Serie 1.

Para la comunicación de datos se utilizó los módulos de comunicación inalámbrica Xbee, los módulos Xbee proveen 2 formas amigables de comunicación:

- Transmisión serial transparente (modo AT).
- Modo API.

Los módulos Xbee pueden ser configurados desde el PC utilizando el programa XCTU o bien desde un micro controlador. Los Xbee pueden comunicarse en arquitecturas punto a punto, punto a multi-punto o en una red.

La elección del módulo Xbee correcto pasa por escoger el tipo de antena (chip, alambre o conector SMA) y la potencia de transmisión (2mW para 300 pies o 60mW para hasta 1 milla), pueden ser usados con adaptadores Xbee Explorer Serial o Xbee Explorer USB. Esta sección se utilizó de base la información que se encontraba en manuales online sobre estos dispositivos (Ingeniería MCI, 2013).

Características:

- De buen Alcance: hasta 300ft (100 metros) en línea vista para los módulos Xbee y hasta 1 milla (1.6 Km) para los módulos Xbee Pro.

CAPÍTULO 3: DISEÑO DE HARDWARE

- 9 entradas/salidas con entradas analógicas y digitales.
- Bajo consumo <50mA cuando están en funcionamiento y <10uA cuando están en modo sleep.
- Interfaz serial.
- 65,000 direcciones para cada uno de los 16 canales disponibles.
- Se pueden tener muchos de estos dispositivos en una misma red.
- Fáciles de integrar.

3.2.1.1. Tipos De Módulos Xbee En Base A Su Serie.



Existen 2 series de estos módulos. La serie 1 y la serie 2 o también conocida como 2.5. Los módulos de la Serie 1 y la Serie 2 tienen el mismo pin-out, sin embargo, no son compatibles entre sí ya que utilizan distintos chipset y trabajan con protocolos diferentes.

La serie 1 está basada en el chipset Freescale y está pensado para ser utilizado en redes punto a punto y punto a multipunto. Los módulos de la serie 2 están basados en el chipset de Ember y están diseñados para ser utilizados en aplicaciones que requieren repetidores o una red mesh. Ambos módulos pueden ser utilizados en los modos AT y API.




MaxStream fabrica más de 70 tipos de módulos Xbee con diferentes antenas, potencia y capacidad.

Muchas de las características de los módulos Xbee tales como velocidad de transmisión y canales por ejemplo pueden ser configurados utilizando el software X-CTU o directamente desde un micro controlador.





Tabla 3.02. Descripción de Tipos de Módulos Xbee.

Xbee	Velocidad de transmisión	Banda de frecuencia	Potencia de transmisión	Antena	Pines IO digitales	Entradas ADC	Rango(ft)
 <p>Xbee 1mW PCB Antenna</p>	115.2 kbp	2.4 GHz	1 mW (+0 dBm)	Built-in	8	(7) 10 bit	300ft (100m)
 <p>Xbee 1mW Wire Antenna</p>	250kbps	2.4 GHz	1mW output (+0dBm)	Built-in	8	(6) 10-bit	300ft (100m)
 <p>Xbee 2mW Chip Antenna - Series 2</p>	250kbps	2.4 GHz	2mW output (+3dBm)	Built-in	8	(6) 10-bit	400ft (120m)
	250kbps	2.4 GHz	2mW output (+3dBm)	Built-in	8	(6) 10-bit	400ft (120m)


CAPÍTULO 3: DISEÑO DE HARDWARE

<p>Xbee 2mW PCB Antenna - Series 2 (ZigBee Mesh)</p>							
 <p>XBee 2mW RPSMA - Series 2</p>	250kbps	2.4 GHz	2mW output (+3dBm)	RPS MA	8	(6) 10-bit	400ft (120m)
 <p>XBee 2mW Wire Antenna - Series 2 (Mesh)</p>	250kbps	2.4 GHz	2mW output (+3dBm)	Built- in	8	(6) 10-bit	400ft (120m)
 <p>XBee Pro 50mW RPSMA - Series 2 (Mesh)</p>	250kbps	2.4 GHz	50mW output (+17dB m).	RPS MA	8	(6) 10-bit	1 mile (1600m)

CAPÍTULO 3: DISEÑO DE HARDWARE

 <p>XBee Pro 50mW Serie 2.5 Wire Antena</p>	250kbps	2.4 GHz	50mW output (+17dB m).	Built- in	8	(4) 10-bit	1 mile (1600m)
 <p>XBee Pro 60mW serie 1 PCB Antena</p>	250kbps	2.4 GHz	60mW output (+18dB m).	Built- in	8	(6) 10-bit	1 mile (1600m)
 <p>XBee Pro 60mW Wire Antenna</p>	250kbps	2.4 GHz	60mW output (+18dB m)	Built- in	8	(6) 10-bit	1 mile (1600m)
 <p>XBee Pro 900 RPSMA</p>	156 Kbps	ISM de 900MH z	50 mW (+17 dBm)	RPS MA	10	(6) 10-bit	6 miles (10 km)

CAPÍTULO 3: DISEÑO DE HARDWARE

 <p>XBee Pro 900 XSC RPSMA</p>	9.6kbps	ISM de 900MH z	100 mW pow er output	RPS MA	none	none	15 miles
---	---------	----------------------	----------------------------	-----------	------	------	----------

Para nuestra aplicación el modulo usado fue el “XBee 1mW Wire Antenna(802.15.4)” debido a su bajo costo, bajo consumo de potencia, tamaño reducido y que su alcance es el indicado para nuestra aplicación.

3.2.1.2. Estándar IEEE 802.15.4

“Las características más importantes en este estándar son su flexibilidad de red, bajos costos, bajo consumo de energía; este estándar se puede utilizar para muchas aplicaciones en el hogar que requieren una tasa baja en la transmisión de datos” (Archundia, 2004).

“La clave la de motivación para el uso de tecnología inalámbrica es la reducción en los gastos de instalación, ya que nunca es necesario cambiar el cableado. Las redes inalámbricas implican un gran intercambio de información con un mínimo de esfuerzo de instalación” (Archundia, 2004).

CAPÍTULO 3: DISEÑO DE HARDWARE

“Esta tendencia es impulsada por la gran capacidad de integrar componentes inalámbricos de una forma más barata y el éxito que tienen otros sistemas de comunicación inalámbrica como los celulares” (Archundia, 2004).

Algunas características de alto nivel del estándar IEEE 802.15.4 se resumen en la tabla 3.03:

Tabla 3.03. Características del Módulo Xbee seleccionado (Archundia, 2004).

Propiedad	Rango
Rango de transmisión	868 MHz: 20kb/s; 915 MHz: 40kb/s; 2.4 GHz: 250 kb/s
Alcance	10 – 20 m
Latencia	Abajo de los 15 ms.
Canales	868/915 MHz: 11 canales. 2.4 GHz: 16 canales.
Bandas de Frecuencia	868/915 MHz y 2.4 GHz.
Direccionamiento	Cortos de 8 bits o 64 bits IEEE
Temperatura	El rango de temperatura industrial: -40° a +85° C

3.2.1.3. Capas de Red

El estándar IEEE 802.15.4 soporta múltiples topologías para su conexión en red, entre ellas la topología tipo estrella y la topología peer-to-peer. La topología a escoger es una elección de diseño de acuerdo a las necesidades de cada aplicación (Archundia, 2004).

CAPÍTULO 3: DISEÑO DE HARDWARE

Algunas aplicaciones en donde se aplica este estándar son aplicaciones como periféricos e interfaces de computador. Estas aplicaciones son conexiones de baja potencia de tipo estrella, mientras que otros como los perímetros de seguridad requieren de una mayor área de cobertura por lo que es necesario implementar una red punto a punto (Archundia, 2004).

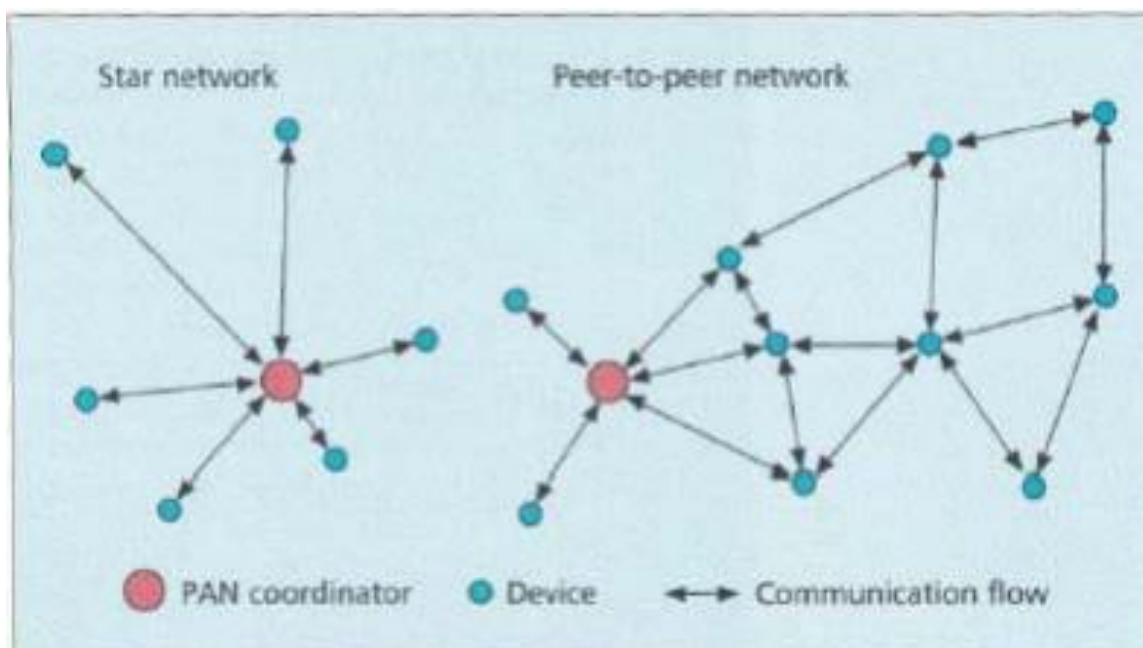


Figura 3.04. Tipos de Redes con Módulos Xbee (Archundia, 2004).

3.2.1.4. XBee 1mW Wire Antenna - Series 1

Este es el módulo XBee de 2.4GHz más popular de Digi. Estos módulos de estándar 802.15.4 son muy fáciles de usar ya que sus comandos seriales son bastantes sencillos. Estos módulos permiten una comunicación muy simple y confiable entre microcontroladores, ordenadores, sistemas, realmente cualquier cosa con un puerto serial, compatible con redes punto a punto y multipunto.

CAPÍTULO 3: DISEÑO DE HARDWARE

Características:

- 3.3V-50mA.
- 250kbps Max velocidad de datos.
- Salida de 1 mW (0 dBm).
- 300 pies (100 m) Rango.
- Cable de antena.
- 6 10-bit ADC pines de entrada.

3.3. ETAPA DE CONTROL

La etapa de control se resume la figura 3.05.

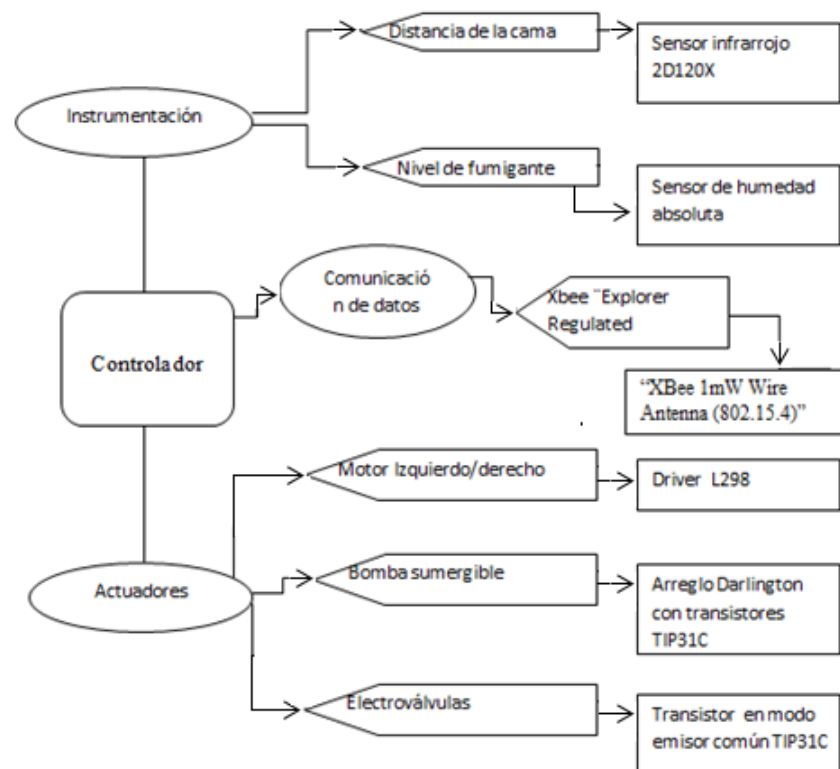


Figura 3.05. Diagrama de Bloques de Control.

CAPÍTULO 3: DISEÑO DE HARDWARE

Para la etapa de control se utiliza un micro controlador PIC de la familia microchip modelo 18f452, debido a su gran capacidad de memoria, a su fácil acceso, precio y debido a que se lo usó en varias aplicaciones a lo largo de la carrera.

Un microcontrolador es un sistema cerrado, lo cual quiere decir que, en un solo circuito, se encierra un sistema digital programable completo. Este dispositivo se destina a gobernar una sola tarea que no se puede modificar. Los microcontroladores disponen de los bloques esenciales: CPU, memorias de datos (RAM) y de programa (ROM), reloj, periféricos de entradas/salidas, entre otros.

Existen muchos modelos de micro controladores los cuales se diferencian por su cantidad de entradas/salidas, si tienen o no conversores analógicos/digitales, puertos de comunicación, temporizadores, etc.

3.4. ETAPA DE INSTRUMENTACIÓN

La etapa de instrumentación se divide en dos partes:

- Medición de distancia
- Medición de nivel

3.4.1. Medición De Distancia.

Para la medición de distancia se usaron los sensores Sharp 2D120X, este sensor permite medir la distancia a una superficie entre 4 a 30 cm, la señal de salida del sensor es

CAPÍTULO 3: DISEÑO DE HARDWARE

analógica que varía de 3.1V a 4cm hasta 0.3V a 30cm (aunque en la practica el sensor es útil de 5cm a 26cm).

“El sensor consta de un LED infrarrojo de alta intensidad que apunta de manera perpendicular al sensor. Un dispositivo opto-electrónico (llamado PSD, Position Sensitive Device) mide entonces el ángulo con el que este rayo se refleja en el obstáculo” (MrGroup - 2009).

La información de la distancia se extrae midiendo el ángulo recibido. Si el ángulo es grande, entonces el objeto está cerca porque el triángulo formado es ancho si el ángulo es pequeño entonces el objeto se encuentra lejos porque el triángulo formado es largo y delgado

En la figura 3.6 se puede ver cómo se lleva a cabo la triangulación en el sensor. El LED infrarrojo emite el haz de luz a través de una pequeña lente convergente que hace que el haz emisor llegue de forma paralela al objeto.

Cuando la luz choca con un obstáculo, una cierta cantidad de luz se refleja; si el obstáculo fuera un espejo perfecto, todos los rayos del haz de luz pasarían, y sería imposible medir la distancia.

Sin embargo casi todas las sustancias tienen un alto grado de rugosidad en la superficie, que produce una dispersión hemisférica de la luz (la llamada reflexión no teórica). Alguno de los haces de esta luz rebota hacia el sensor que es recibido por la lente. La lente receptora también es una lente convexa, pero ahora sirve para un propósito diferente, actúa para convertir el ángulo de posición.

CAPÍTULO 3: DISEÑO DE HARDWARE

Si un objeto se pone en el plano focal de una lente convexa y los otros rayos de luz paralelos en otro lado, el rayo que pasa por el centro de la lente atraviesa inalterado o marca el lugar focal. Los rayos restantes también enfocan a este punto (Simbaña, O.V. (2006). Medidor de Líquidos Controlado por un Microcontrolador. Tesis de tecnología no publicada EPN, Quito, Ecuador).

“Puesto en el plano focal es un Sensor Detector de Posición (PSD). Este dispositivo semiconductor entrega una salida cuya intensidad es proporcional a la posición respecto al centro (centro eficaz) de la luz que incide en él” (MrGroup - 2009).

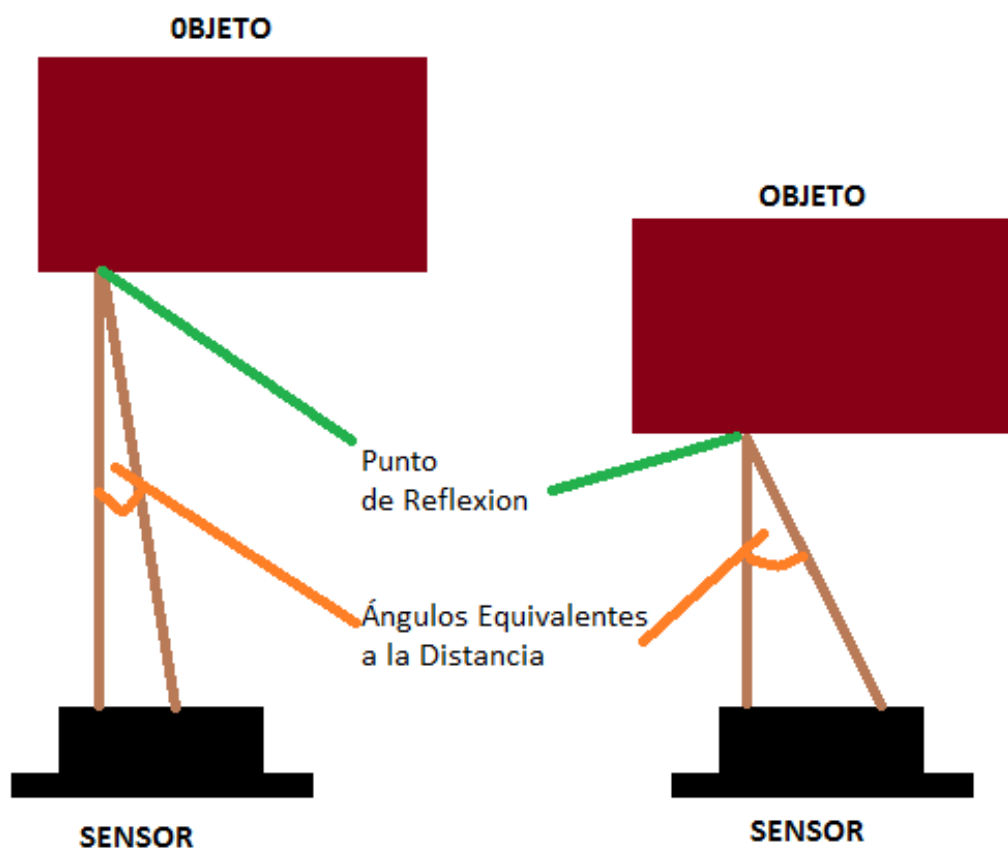


Figura 3.06. Forma de Medición Sensor Sharp.

CAPÍTULO 3: DISEÑO DE HARDWARE

Estos sensores son utilizados para medir la distancia entre la plataforma y la cama, así dependiendo en que cama se encuentre, este podrá mantenerse en una trayectoria lo más recta posible y a la vez llegado el fin de la cama determinar la dirección a la cual este va a girar.

Partiendo de la ecuación de distancia que el fabricante de estos sensores indica en su hoja técnica (Ecuación 3.1); en donde el valor del sensor no representa el voltaje que entrega el sensor, sino un valor en bits de conversión analógica digital (digitalización, representada en un numero entero de 1 a 1023).

Por ello, es necesario definir la distancia en función del voltaje en base de la relación indicada para la digitalización (Ecuación 3.3), esto en base a una regla de tres simple tomando en cuenta los valores de voltaje con su respectivo valor de conversión (Ecuación 3.2).

$$Distancia(cm) = \frac{2076}{Valor\ del\ sensor - 11}$$

Ecuación 3.1. Ecuación de Distancia en Función a la Digitalización.

$$1025 \rightarrow 5v$$

$$Valor\ del\ sensor \rightarrow Voltaje$$

Ecuación 3.2. Relación de Conversión versus Voltaje.

$$Valor\ del\ sensor = 204.6 * Voltaje$$

Ecuación 3.3. Ecuación de Distancia en Función a la Digitalización.

Donde reemplazando 3.3 en 3.2 se obtiene:

$$Distancia(cm) = \frac{2076}{204.6 * Voltaje - 11}$$

Ecuación 3.4. Ecuación de Distancia en Función a la Digitalización.

En la ecuación 3.4 se puede observar la relación existente entre la distancia que mide el sensor, y el voltaje que entrega al microcontrolador para acondicionarla.

Características:

- Rango: 4cm-30m.
- Campo de detección: 8°.
- Alimentación: Desde 4.5V hasta 5V.

Ventajas:

- Cono de detección muy estrecho: Perfecto para mapear.
- Buena resolución a distancias cortas.

Inconvenientes:

- Cada uno de los sensores debe ser calibrado por separado.

CAPÍTULO 3: DISEÑO DE HARDWARE

3.4.2. Medición de nivel

Para la medición de nivel se usaron sensores de humedad, los cuales al detectar presencia de líquidos producen el continuidad entre dos contactos. Esto usando las características físicas que poseen los líquido, mas aun tomando en cuenta que el fungicida tiene componentes salinos que mejoran ;a conductividad del agua.

Estos sensores son cables solidos cuyos terminales son de aluminio y van sumergidos por pares dentro del contenedor de fungicida, de manera que se produzca continuidad al momento de que ambos terminales se encuentren sumergidos. En la figura 3.07. se muestra la conexión de dichos elementos.

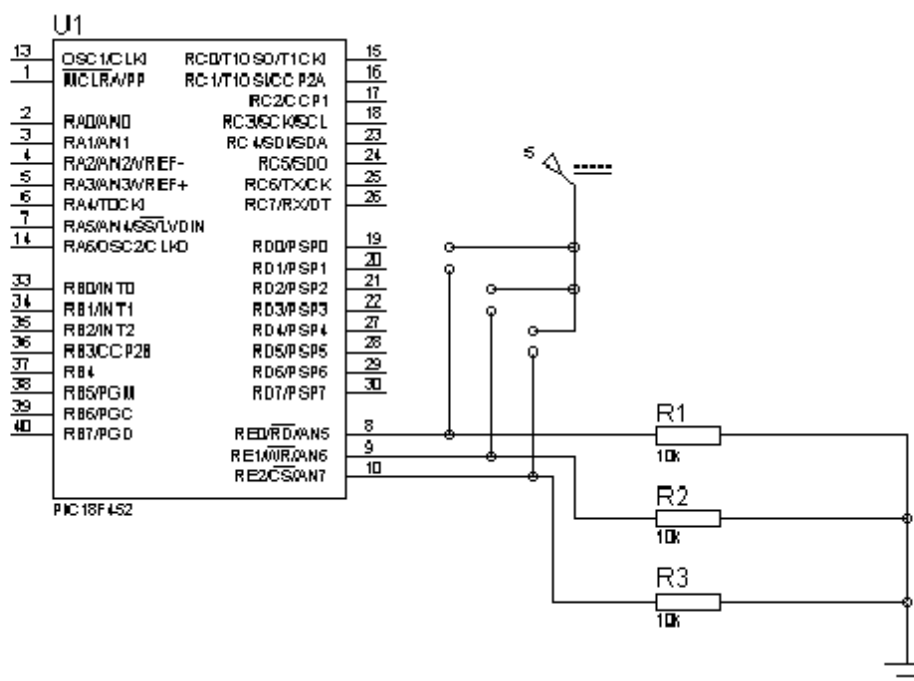


Figura 3.07. Conexión sensores de Nivel.

CAPÍTULO 3: DISEÑO DE HARDWARE

Como se observa en la figura 3.7, existen divisores de tensión a la salida de los pines del microcontrolador, esto con la finalidad de que si existe continuidad entre los cables que van a 5 voltios (cable sumergido), establezca un estado alto en las entradas del controlador. Mas si no existe esta conduccion, el estado es bajo por encontrarse conectados los demas extremos de los divisores a tierra. Se colocaron tres de estos sensores para determinar si el liquido se encuentra en un nivel alto, medio y bajo, según el nivel se procede a deerminar si sepuede o no proceder a la fumigacion ademas de la velocidad que va a tener la plataforma.

3.5. SISTEMA DE ALIMENTACIÓN

La plataforma fue alimentada mediante una bateria recargable de 12Vdc/10Ah y otra de 9Vdc, posteriormente, en el tablero decontrol se generan distintos voltajes por medio de reguladores para alimentar a los distintos circuitos, como se muestra en la figura 3.08.



Figura 3.08. Distribucion de Voltajes.

CAPÍTULO 4

DESARROLLO DE SOFTWARE

En el presente capítulo se describen todos los componentes que constituyen el sistema de software de control de MI-T-NATOR. Para esto, se realizan diagramas de flujo que indican las secuencias necesarias para la lógica de control, tomando en cuenta ciertos problemas que el sistema pueda encontrar a lo largo de su operación.

Además de esto, se muestra la distribución gráfica de la Interfaz Humano Máquina (HMI) que permitirá al operario determinar los movimientos adecuados que MI-T-NATOR debe realizar para cumplir con el objetivo de fumigar las camas de rosas. Por último, se explica el diseño de una pequeña base de datos, de tal forma que sirva de bitácora para determinar la fecha, hora y número de camas en las que se realizó la fumigación. Esto con el fin de llevar un registro histórico de las actividades realizadas por MI.T.NATOR.

4.1. LÓGICA DE CONTROL

La lógica de control es una secuencia de comandos, que se representa en diagramas de flujo, de manera que se pueda entender a cabalidad las instrucciones a ser cargadas al microcontrolador. Estas secuencias de comandos determinan las distintas tareas que el sistema

debe realizar, tomando siempre en cuenta las condiciones existentes a su alrededor, mediante la etapa de instrumentación que se encuentre conectada al controlador.

Esta lógica de control se conoce como el firmware del sistema, y no es más que una serie de comandos programados al microcontrolador para realizar determinada secuencia de comandos. Estas secuencias dependen directamente de las etapas de potencia, control, comunicaciones y sensamiento. Este firmware está programado en lenguaje C, el cual es compilado a lenguaje hexadecimal, para que de esta manera pueda ser interpretado por el microcontrolador.

El firmware consta de varias rutinas y subrutinas, con lo que se consigue una programación en bloques, de modo que no se generen confusiones al momento de interpretar el código secuencial que debe realizar el controlador. Dicho software ya compilado, es cargado mediante un programador de PIC hacia el controlador, que en este caso es un microcontrolador con numeración PIC 18F452.

Se ha escogido este controlador por la funcionalidad de memoria que este tiene, la gran cantidad de puertos disponibles, comunicación serial RS-232, y varios puertos que se usan como conversores análogos digitales.

4.1.1. Distribución y Configuración De Puertos

En esta sección se detalla la distribución y configuración de los puertos. En la Tabla 4.1, se aprecia la distribución de puertos del microcontrolador, a fin de dar una idea clara de las funciones que cada puerto debe tener dentro de la programación. Además, esta tabla hace referencia al tipo de configuración que cada puerto tendrá, a más de dar ciertas observaciones para determinados puertos.

Tabla 4.01. Distribución de Puertos.

Función de Puerto	Núm.	Nombre de Puerto	Área de Programación	Configuración de Puerto	Observaciones de Software
Sensor Delantero Izquierdo	0	A	Posición sobre Campo	Entrada Analógica	Calibración Requerida
Sensor Delantero Derecho	1		Posición sobre Campo	Entrada Analógica	Calibración Requerida
Sensor Trasero Izquierdo	2		Posición sobre Campo	Entrada Analógica	Calibración Requerida
Sensor Trasero Derecho	3		Posición sobre Campo	Entrada Analógica	Calibración Requerida
No Usado	4				
No Usado	5				
Bomba de Succión	0	B	Control Aspersión	Salida Digital	
Válvula Izquierda	1		Control Aspersión	Salida Digital	
Motor Izquierdo	2		Movilidad Campo	Salida Digital	
Motor Izquierdo	3		Movilidad Campo	Salida Digital	
Válvula Derecha	4		Control Aspersión	Salida Digital	
No Usado	5				
Motor Derecho	6		Movilidad Campo	Salida Digital	
Motor Derecho	7		Movilidad Campo	Salida Digital	
No Usado	0	C			
No Usado	1				
No Usado	2				
No Usado	3				
No Usado	4				
Recepción	5		Comunicación	Entrada Digital	Requiere

USART RS232			con PC		Configuración
Transmisión USART RS232	6		Comunicación con PC	Salida Digital	Requiere Configuración
No Usado	7				
No Usado	0	D			
No Usado	1				
No Usado	2				
No Usado	3				
No Usado	4				
No Usado	5				
No Usado	6				
No Usado	7				
Nivel Bajo de Líquido	0		E	Indicador de Nivel	Entrada Digital
Nivel Medio de Líquido	1	Indicador de Nivel		Entrada Digital	Acondicionamiento de Señal
Nivel Alto de Líquido	2	Indicador de Nivel		Entrada Digital	Acondicionamiento de Señal

4.1.2. Esquema Principal Del Firmware.

Una vez que se ha determinado la configuración de software adecuada para los puertos del controlador, se debe dar una idea general de la secuencia de control. La Figura 4.01, muestra el diagrama principal del firmware que el controlador debe seguir. Cabe mencionar que dentro de este esquema existirán rutinas, subrutinas, y funciones que el microcontrolador debe realizar de acuerdo al estado en el que se encuentre el sistema.

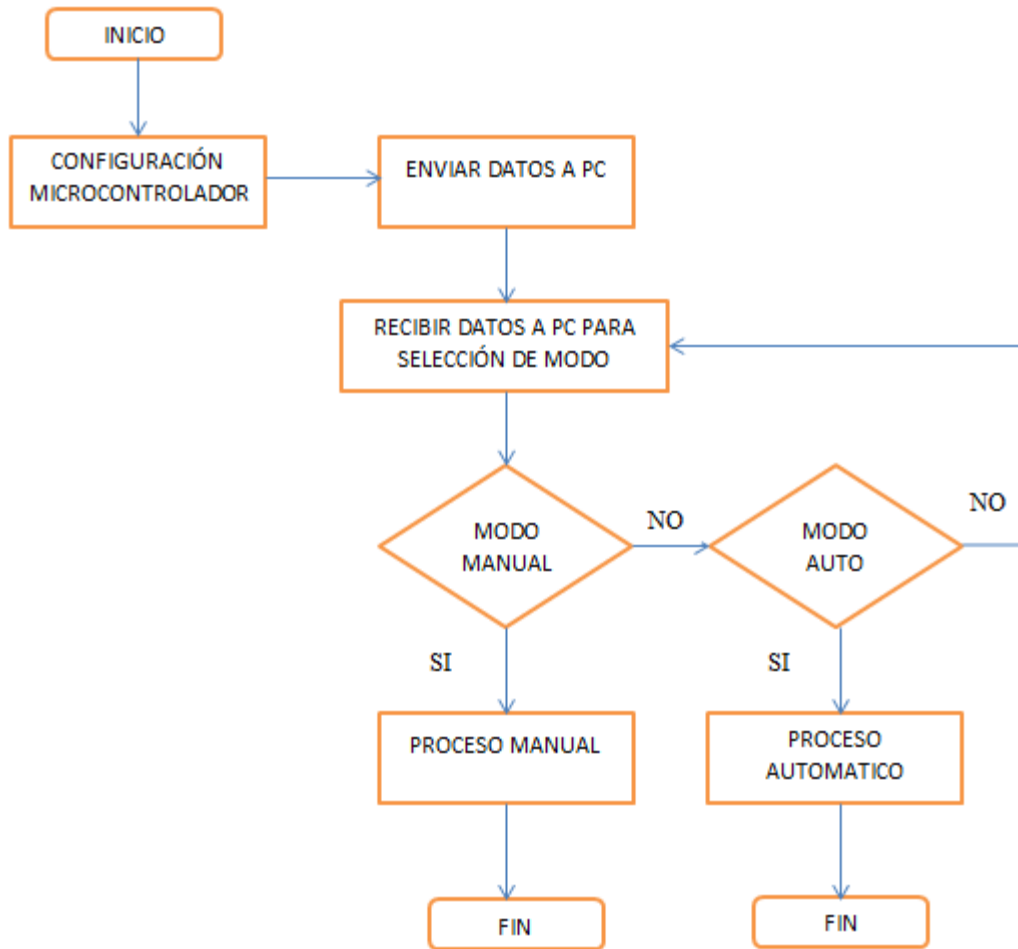
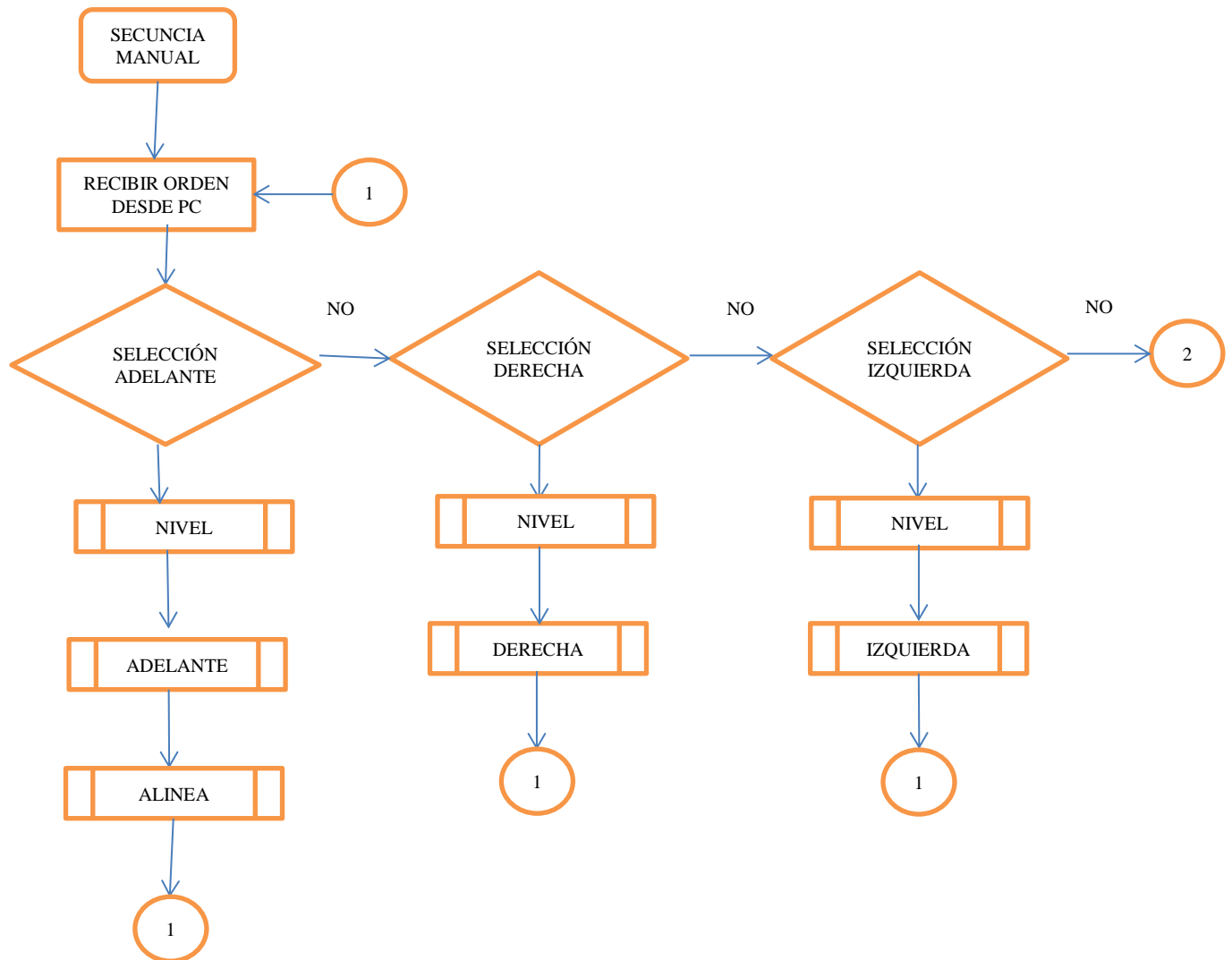


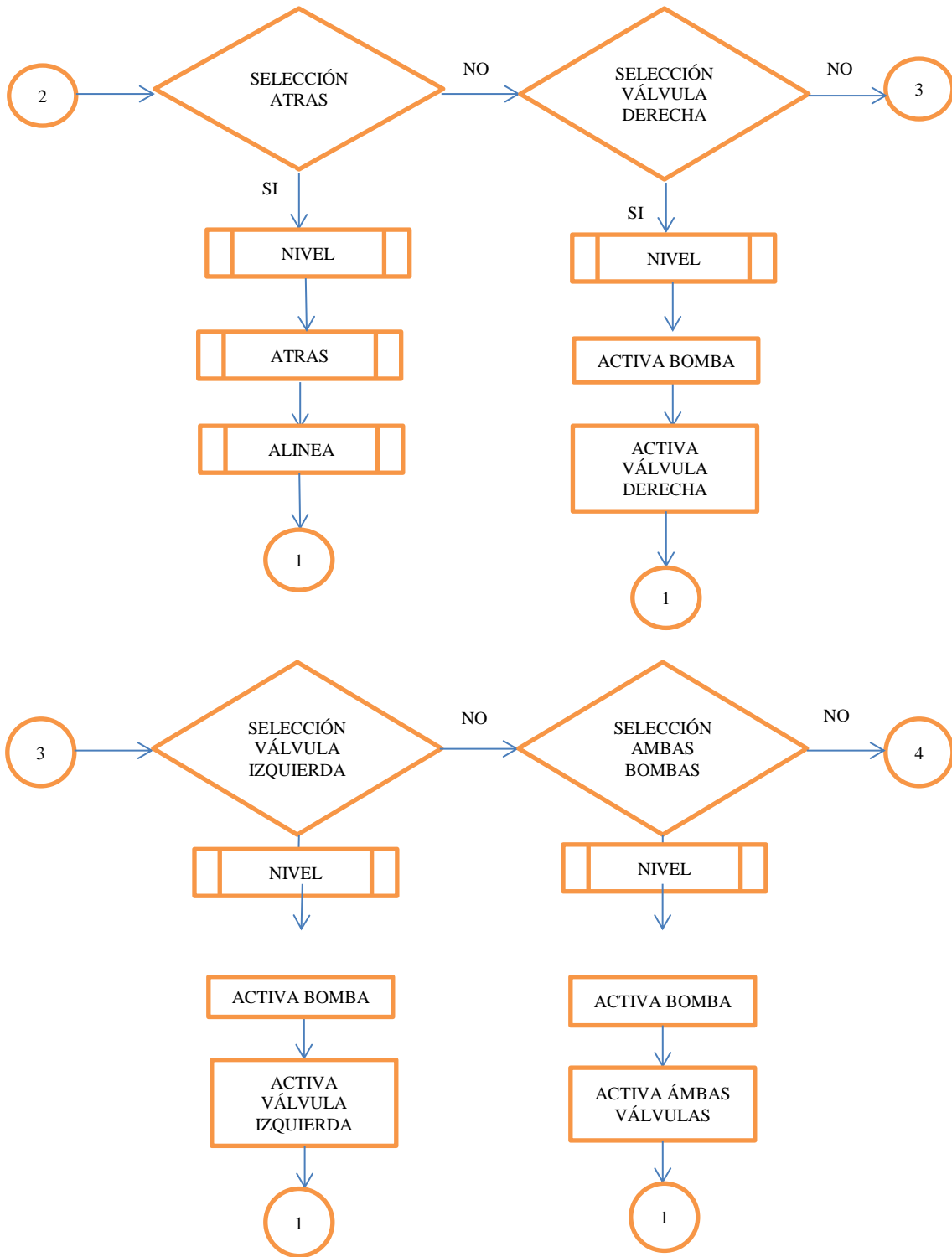
Figura 4.01. Diagrama de Flujo Principal.

Como se puede observar en la Figura 4.01, luego de cargar la palabra de control hacia el PIC, es necesario indicar bajo qué modo de operación MI-T-NATOR debe funcionar. Por ello, se escoge entre dos modos de operación, los cuales son: modo manual y modo automático. Cada uno de ellos tiene su propio proceso, y dentro de ellos existirán determinadas rutinas y funciones específicas de cada uno de estos modos.

4.1.3. Modo Manual.

El modo manual, es el modo bajo el cual MI-T-NATOR es controlado por un operario mediante un computador. En esta sección nos enfocaremos en analizar las secuencias lógicas que controlan al robot, dependiendo de las necesidades que el operador tenga en determinado momento. En primera instancia, se detalla el comportamiento general de este modo, para luego hacer referencia a cada una de las subrutinas que este modo posee, y que son necesarias para el correcto funcionamiento del mismo. En la figura 4.02 se observa el diagrama de flujo del modo manual.





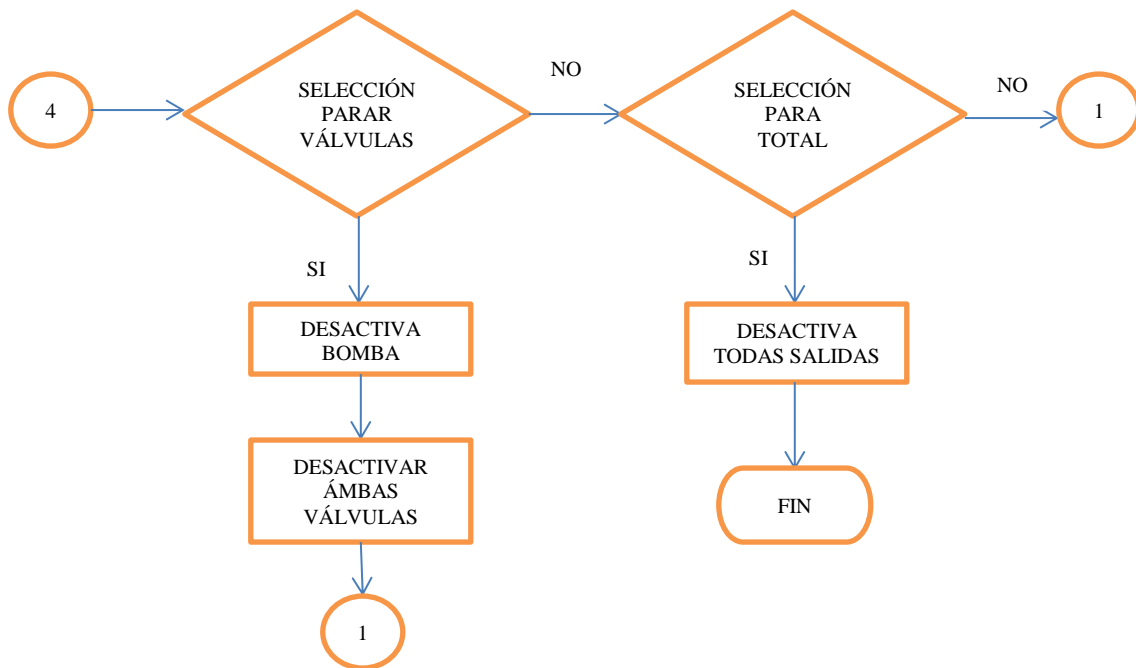


Figura 4.02. Diagrama de Flujo Modo Manual.

Se puede observar en la Figura 4.02, la secuencia lógica que el controlador sigue al momento de seleccionar el modo manual. En este caso siempre se recibe un dato por parte del computador para realizar una determinada acción. Dentro de este modo, se encuentran varias *subrutinas* que se detallan a continuación.

4.1.3.1. Función Nivel.

La función nivel es la encargada de tomar datos del estado del contenedor de fungicida. Lo que se busca es validar estos datos, y prevenir así que se inicie el funcionamiento sin que exista el suficiente volumen de líquido para empezar la fumigación.

Además, el peso de toda la estructura cambia acorde se realiza la fumigación, por lo cual levanta banderas de acuerdo al estado para futuros controles. En la figura 4.03 se muestra el diagrama lógico que usa esta función.

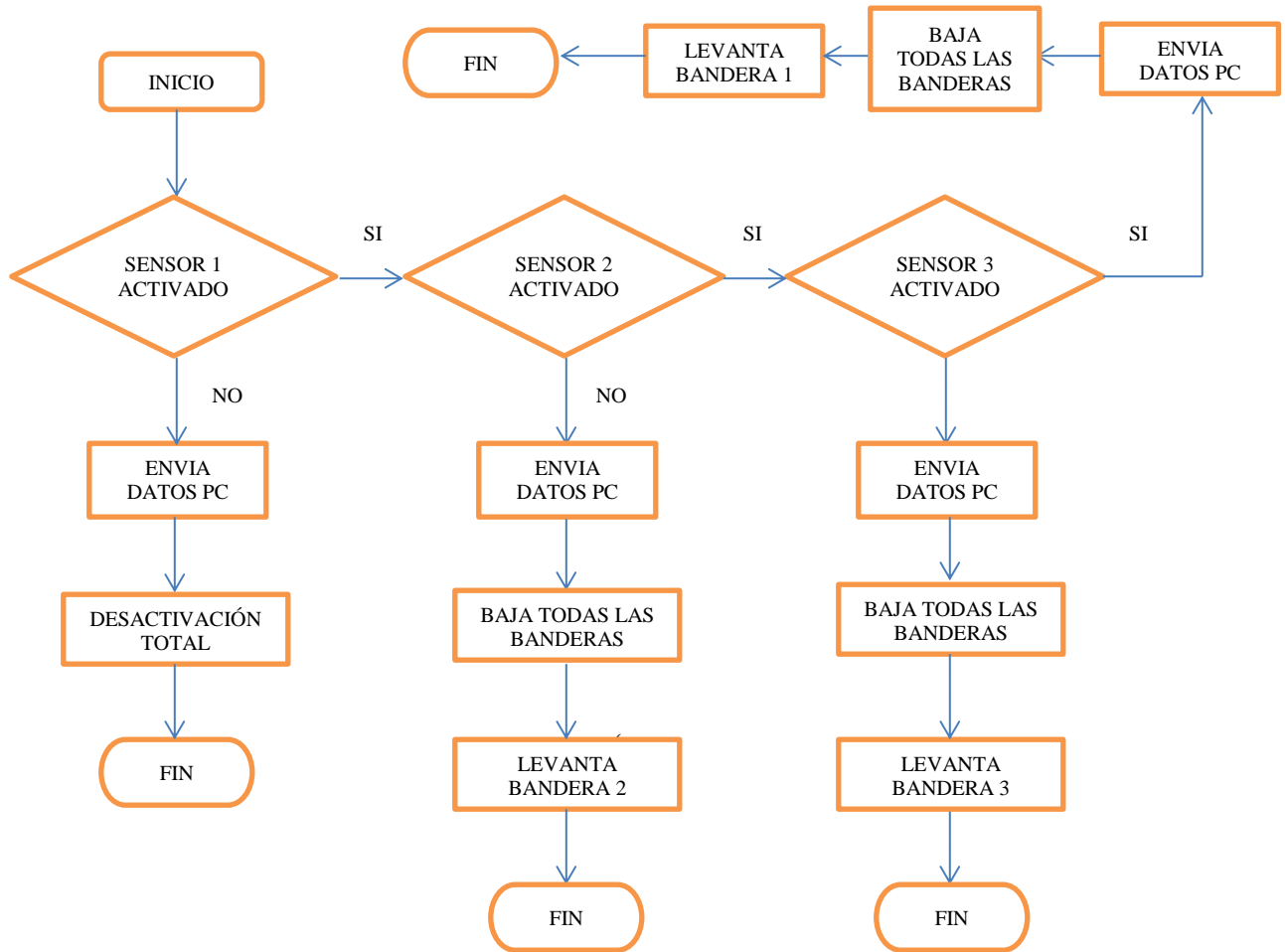


Figura 4.03. Método Nivel.

4.1.3.2. Función Adelante.

La función adelante es la encargada de activar los motores en el mismo sentido de giro, de manera que ambos muevan a la plataforma hacia adelante. Para esto previamente se debe verificar el estado de las banderas activadas por nivel, de tal forma que se controle la velocidad de giro mediante un porcentaje de los módulos PWM del controlador. En la figura 4.04 se muestra el diagrama de flujo de la presente función.

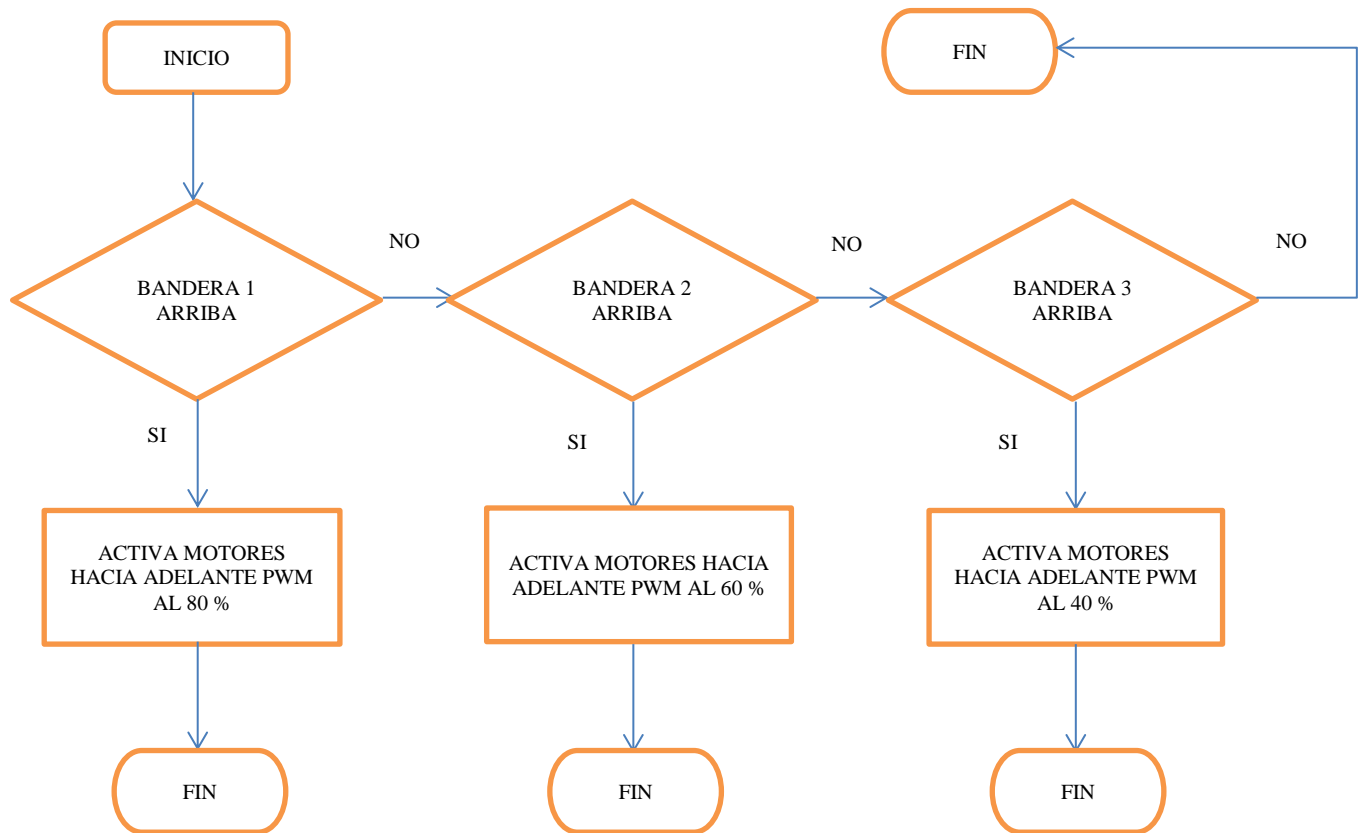


Figura 4.04. Método Adelante.

4.1.3.3. Función Alinea.

El método alinea es el responsable de corregir la dirección al momento que MI-T-NATOR se traslada hacia adelante o atrás. Debido a que los motores usados, pese a ser del mismo tipo, no son idénticos, la plataforma se desvía un poco hacia los lados.

Para corregir esto se realiza un control en lazo cerrado de manera que se pueda corregir la dirección aumentando la velocidad del motor hacia el cual se dirige la desviación. Para ello se utilizan los sensores infrarrojos que se mencionaron en el capítulo 3. En la figura 4.05 se aprecia el diagrama de bloques de esta función.

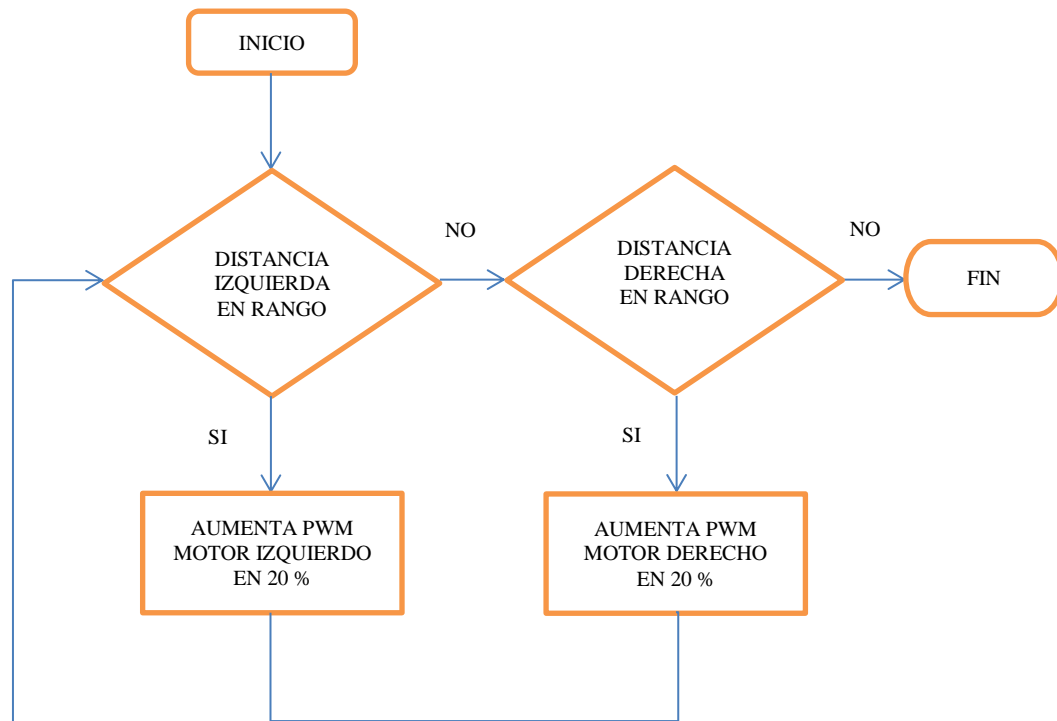


Figura 4.05. Método Alinea.

4.1.3.4. Función Atrás.

La función atrás determina el movimiento de la plataforma en reversa, para lo cual el controlador ordena a los motores girar hacia el mismo sentido, pero en dirección opuesta al método adelante. En la figura 4.06 se detalla el comportamiento lógico de esta función.

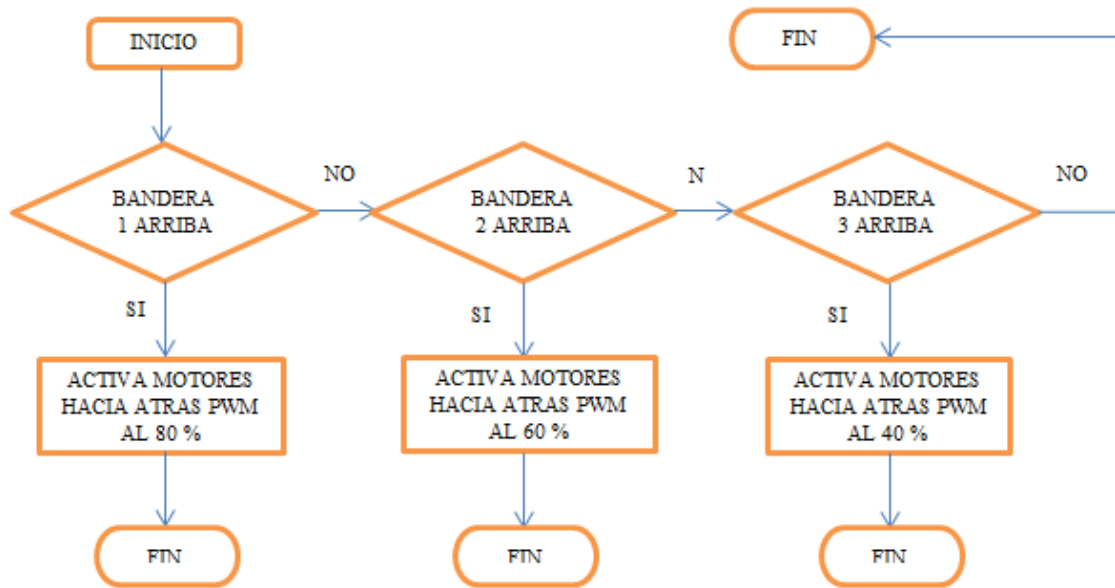


Figura 4.06. Método Atrás.

4.1.3.5. Función Izquierda.

Esta función es la encargada de realizar el giro de la plataforma hacia la izquierda, para lo cual activa solamente el motor de izquierdo. El motor derecho por otro lado permanecerá sin giro, provocando así que MI-T-NATOR gire adecuadamente sobre su eje. Cabe mencionar que para realizar este giro.

Dado el peso que debe mover en el giro de acuerdo a la activación de banderas de la función Nivel. Si la bandera 1 se encuentra levantada, significa que el recipiente está lleno de fungicida, el motor izquierdo ira hacia adelante y el motor derecho hacia atrás. De esta manera se logra dar la suficiente fuerza para que la plataforma gire su propio eje. Al igual que la función adelante, la velocidad de giro de los motores estará establecida por la bandera que se encuentre activada. Su lógica se muestra en la figura 4.07.

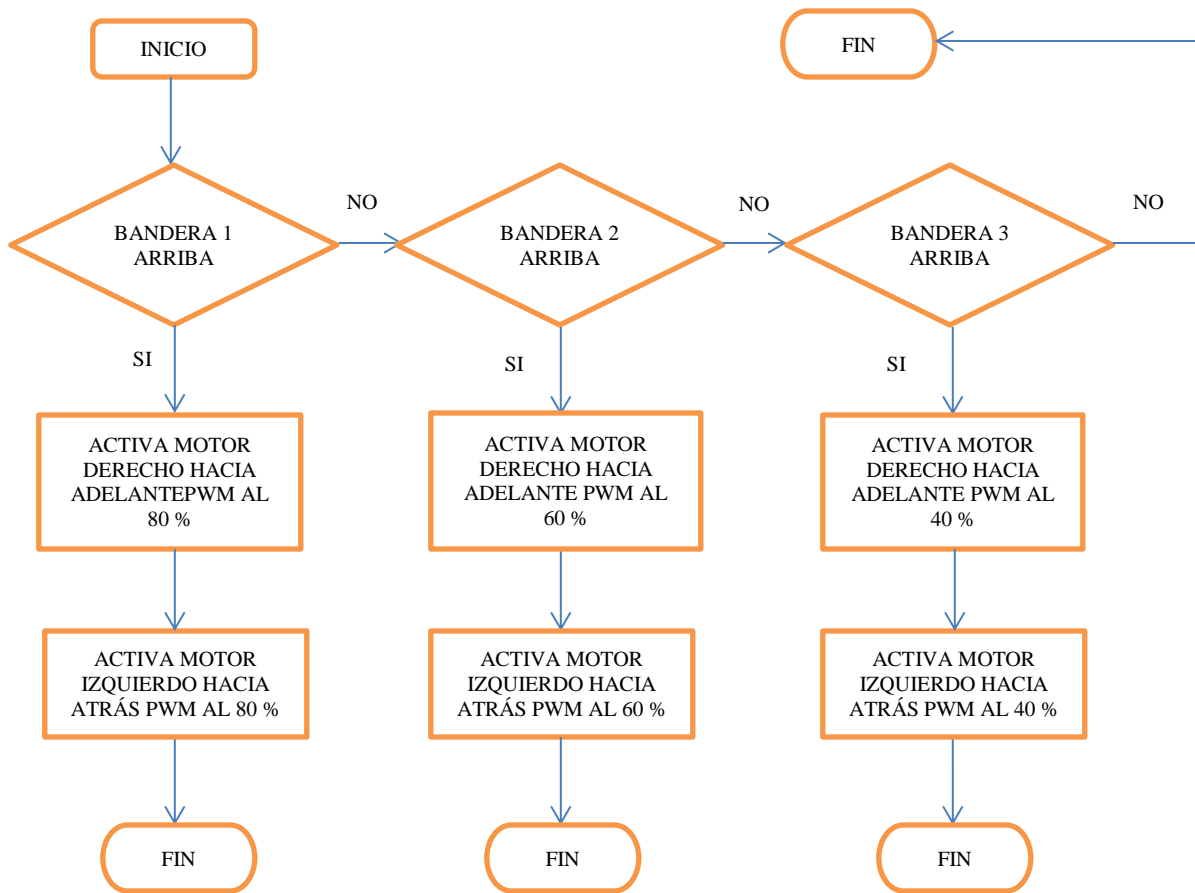


Figura 4.07. Método Izquierda.

4.1.3.6. Función Derecha.

La función derecha tiene una estructura similar a la función izquierda, mas difiere en invertir el sentido de giro de los motores, con lo cual la plataforma gira sobre su propio eje hacia la derecha cuando se realiza la llamada a esta función, que se detalla en la figura 4.08.

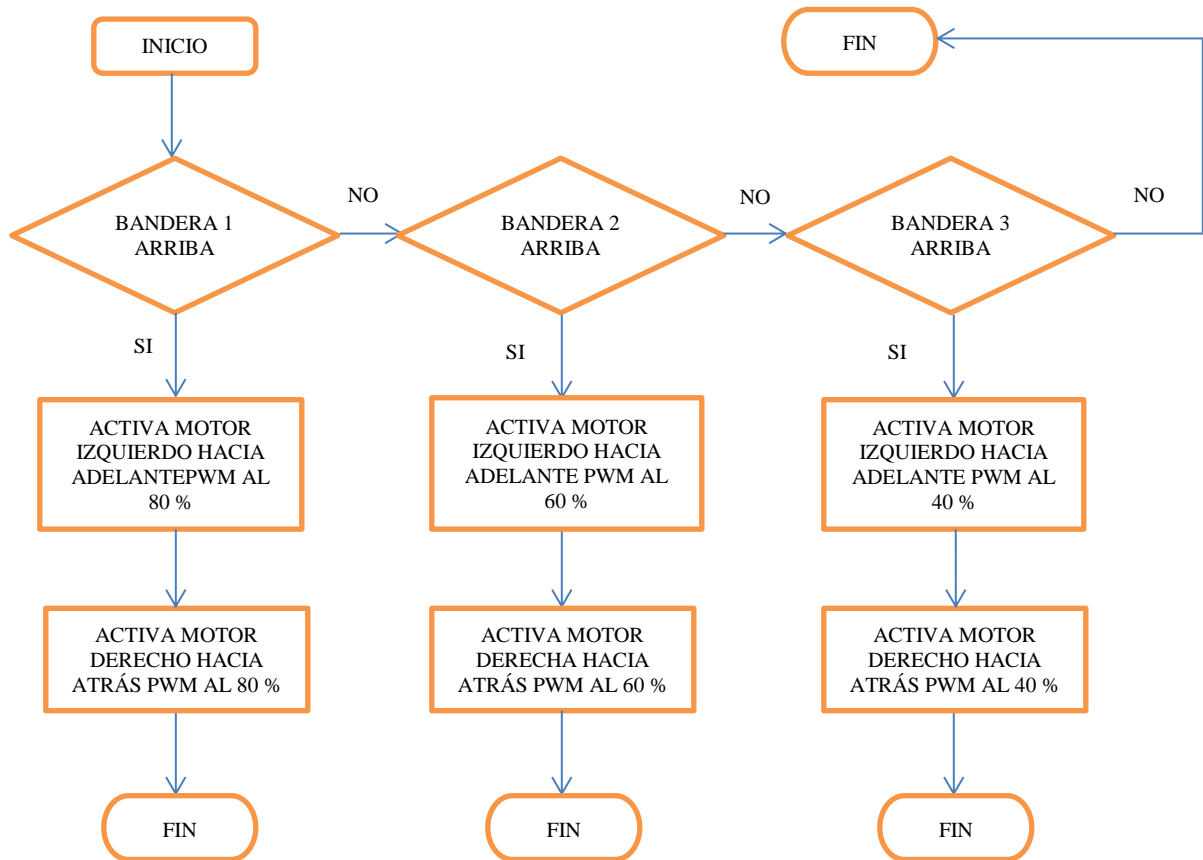


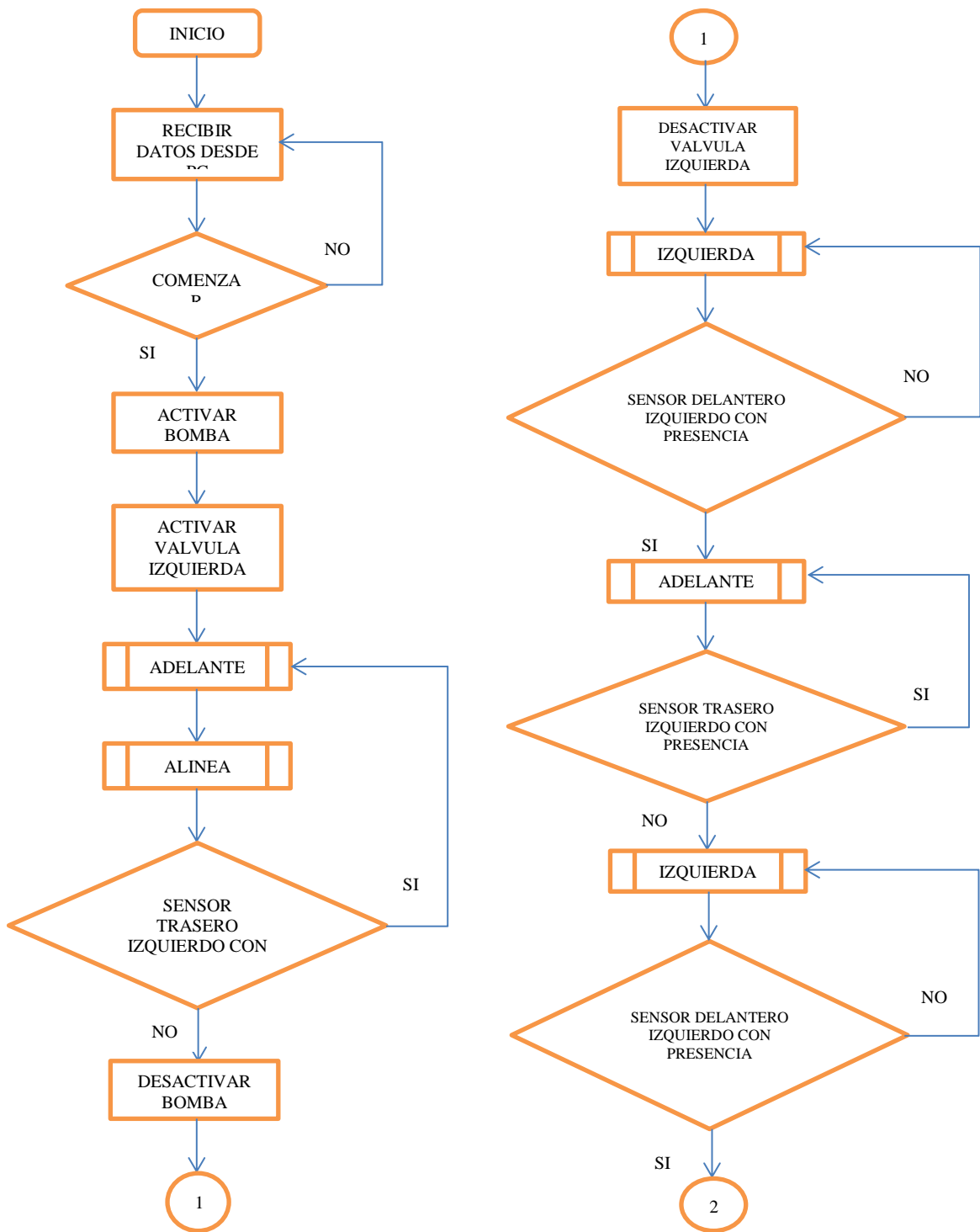
Figura 4.08. Método Derecha.

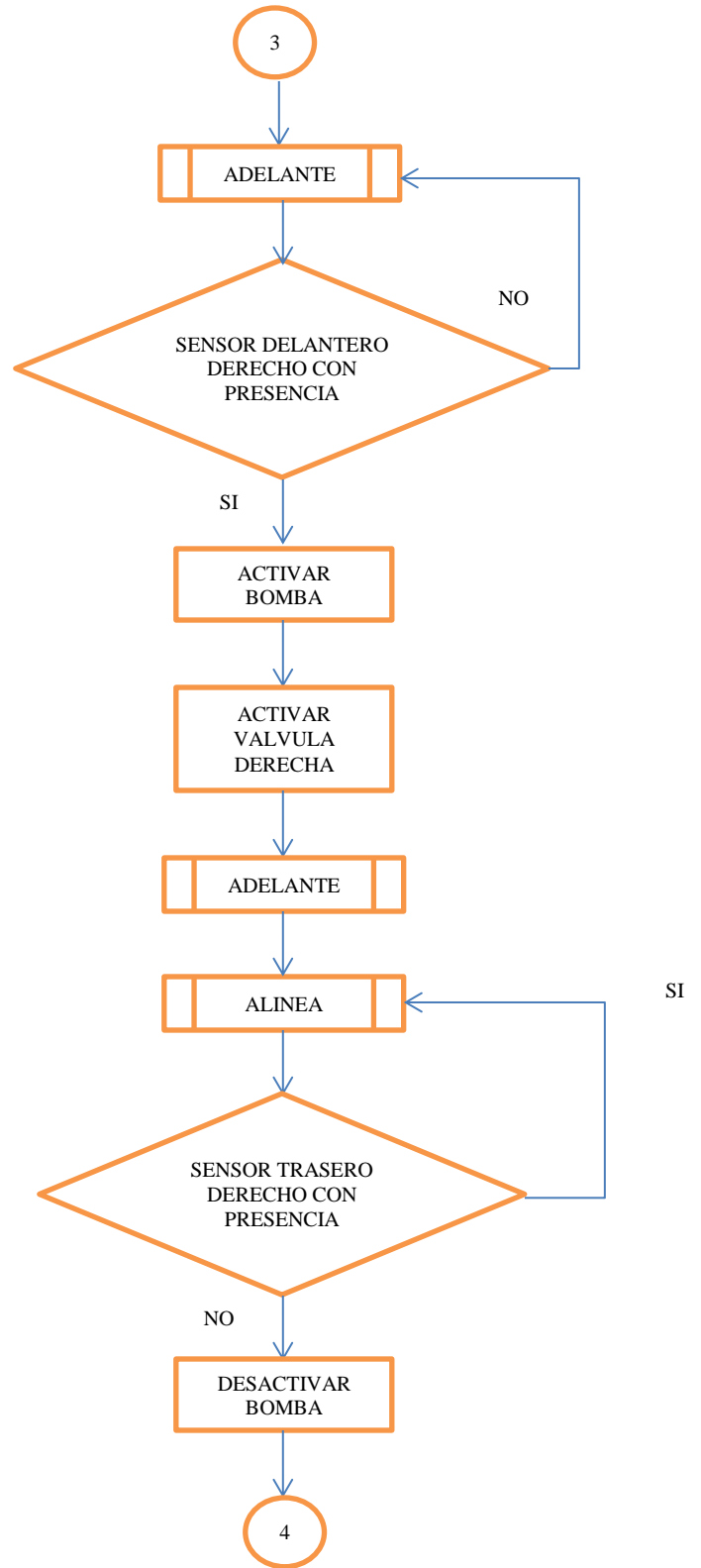
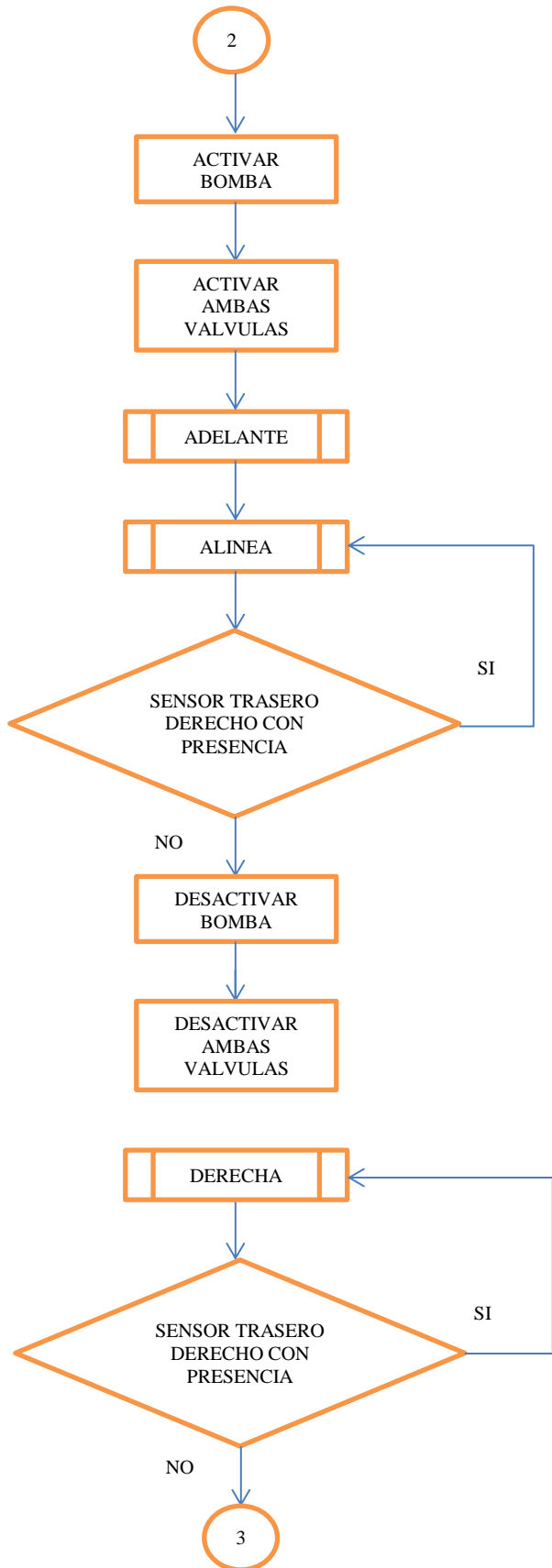
De esta manera es como el modo manual realiza su secuencia lógica, con una programación estructurada en funciones, de manera que se ahorre espacio de memoria y no se confunda líneas de programación.

4.1.4. Modo Automático.

El modo automático a diferencia del modo manual, realiza la fumigación sobre las rosas de manera autónoma. Es decir, MI-T-NATOR únicamente necesita recibir la selección de modo y la orden de inicio para comenzar la fumigación, además este modo tiene una parada de emergencia que se ejecuta en cualquier instante; por lo que, para simplificar el

espacio en el diagrama no se lo considera. El modo automático, usa algunas de las funciones que se explicaron en el modo manual, dichas funciones tienen una programación versátil para evitar usar líneas de código innecesarias, como se muestra en la figura 4.09.





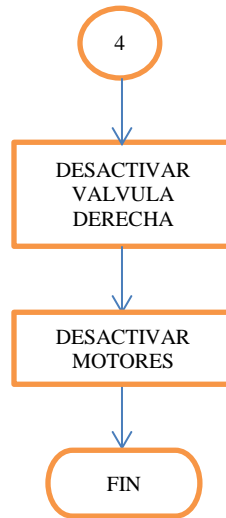


Figura 4.09. Modo Automático.

Esta es la secuencia lógica que debe seguir el modo automático para cumplir con la fumigación requerida. Como se observa en la figura 4.09 es una lógica secuencial que cumple el controlador para realizar una actividad después de otra. Si en el entorno no se presenta ningún factor externo que perjudique la labor, todo el proceso se llevara a cabo satisfactoriamente.

4.2. HMI

La Interfaz Humano Máquina o HMI, por sus siglas en inglés, es el medio por el cual un operador se puede comunicar con MI-T-NATOR para de esta forma controlar las acciones que la plataforma va a realizar. Dicha interfaz se caracteriza por la sencillez que tiene, de manera que sea entendible para cualquier operador que deba supervisar el proceso de fumigación.

Existen muchas herramientas de programación que permiten desarrollar interfaces gráficas para realizar control. Pero se decidió escoger la plataforma de desarrollo Java, ya que es

una plataforma libre que no necesita licencia para su uso. A más de esto, Java tiene la característica de ser un software multiplataforma, que incluso puede cargarse en dispositivos móviles.

Esto permite que por medio de Java se dé una amplia versatilidad a la aplicación HMI, para que en futuros desarrollos de la aplicación, se pueda realizar el control de MI-T-NATOR mediante un teléfono celular o una tableta personal.

El software Java utilizado para el desarrollo de la HMI es Netbeans IDE 7.2.1, y el compilador es el JDK (Java Development Kit) en su versión 7. Ambos programas son de libre acceso, y se los puede encontrar en internet en sus versiones más actuales.

4.2.1. Programación En Netbeans

La programación en Netbeans es una programación orientada a objetos, lo que quiere decir que se deben crear objetos de las distintas clases para poder acceder a los datos y métodos de dichas clases. Cabe mencionar además que en el proyecto se usan varias librerías que son externas a las clases básicas de Netbeans.

Estas librerías requieren ser importadas al proyecto de desarrollo para poder ser usadas, y sus funciones son específicas para el desarrollo de HMI. A continuación se especifican las distintas funciones que se usan para determinadas acciones en la programación.

A continuación mencionamos ciertas propiedades y el modo de funcionamiento de los métodos más importantes utilizados en el proyecto. Para mayor información remitirse a los anexos, donde se encuentra el total del código.

4.2.1.1. Comunicación Serie (Giovynet).

La comunicación serie es el protocolo por el medio del cual se realiza la transferencia de datos entre MI-T-NATOR y el computador, es una comunicación RS-232 con la salvedad que se usan módulos inalámbricos para la transferencia de datos. Para abrir el puerto de comunicaciones en Netbeans, se utiliza la librería giovynet que permite manejar el puerto, y configurar los parámetros de comunicación como son la velocidad de transmisión, el bit de parada, el nombre del puerto libre y la paridad.

Para utilizar esta librería es necesario importar en la cabecera del programa algunos métodos que esta posee, y que facilitan la utilización de sus propiedades. Tales funciones importadas son:

```
importgiovynet.serial.Baud;  
import giovynet.serial.Com;  
importgiovynet.serial.Parameters;
```

Posteriormente, en cada evento del programa, en donde sea necesario enviar órdenes hacia la plataforma, se llaman a las funciones de configuración y de acción de la librería giovynet. A continuación se muestra la configuración que se usa a lo largo del programa en cada evento que así lo requiera.

```
try {
```

```
        Parameters settings = new Parameters();
settings.setPort("COM5");           //numero de puerto
settings.setBaudRate(Baud._9600);   //velocidad en baudios
settings.setMinDelayWrite(10);      //retraso de espera
Com com1 = new Com(settings);
com1.sendSingleData('u');           //envía un caracter por el puerto
        String u1=com1.receiveSingleString(); //recibe un dato
com1.close();                       //cierra el puerto
        } catch (Exception ex) {           //método excepciones
ex.printStackTrace();
        }
```

Todas las sentencias de ejecución se encuentran encerradas en un método try, lo cual genera excepciones si se presenta un error. El momento que se presenta un error, el ciclo de programación entra en el método catch, que captura la excepción y nos muestra el motivo por el que se presentó el error.

Como se puede observar luego de la sentencia try, se coloca el método Parameters que instancia un objeto de esta clase, lo que nos permite dar los parámetros de configuración del puerto. En las líneas subsiguientes utilizamos el objeto denominado settings para establecer los parámetros de configuración.

Una vez configurado el puerto, usamos la clase Com para crear el objeto com1, el que nos permitirá usar las funciones de apertura, envío, recepción y cierre del puerto. Para instanciar este objeto, es necesario cargarlo con el objeto settings, para que el método com1 sepa la ruta para la comunicación.

4.2.1.2. Eventos de Teclado.

Esta sección fue desarrollada pensando en la facilidad que se debe dar a los operarios la interfaz para controlar el sistema móvil. Se consideró adecuado provocar que los movimientos del MI-T-NATOR con solo oprimir teclas. Por ello se usa la función Keyevent. Para ello en un principio, importamos en la cabecera del programa las clases que se necesitan para operar.

```
importjava.awt.KeyEventDispatcher;  
importjava.awt.KeyboardFocusManager;  
importjava.awt.event.KeyEvent;
```

Luego debemos crear una clase dentro del proyecto, de manera que Java permita instanciar los métodos de esta clase como se muestra a continuación:

```
private class Teclado implements KeyEventDispatcher {  
    publicbooleandispatchKeyEvent(KeyEvent e) {  
        if (e.getKeyCode() == KeyEvent.VK_UP) {  
  
            }  
        }  
    }  
}
```

Como podemos ver, se crea una clase denominada Teclado que implementa o hereda las funciones de KeyEventDispatcher. De esta forma podemos crear el método dispatchKeyEvent e instanciar el evento e. La sentencia if verifica cual fue la tecla presionada mediante una comparación. En el ejemplo se compara la tecla presionada con la flecha arriba del teclado.

Luego de esto creamos un objeto de la clase `KeyboardFocusManager` con el fin de arrancar la lectura del teclado. Esta sentencia se debe colocar en el evento desde donde se quiere empezar a leer eventos de teclado. Una vez que se ha instanciado este objeto, le añadimos el método teclado para que realice las acciones pertinentes de acuerdo a lo que se necesite.

```
KeyboardFocusManager manager =  
KeyboardFocusManager.getCurrentKeyboardFocusManager();  
    manager.addKeyEventDispatcher(new Teclado());
```

Con esto, indicamos a la interfaz, el punto desde el cual debe empezar a reaccionar frente a los eventos de teclado, dependiendo de qué tecla fue pulsada.

4.2.2. Distribución Gráfica.

Para realizar la distribución gráfica se usó la herramienta de Netbeans Jform, que permite crear aplicaciones gráficas únicamente arrastrando los distintos componentes que se deseen colocar en el formulario, o pantalla principal. En la figura 4.10 se muestra la distribución de las herramientas de Netbeans.

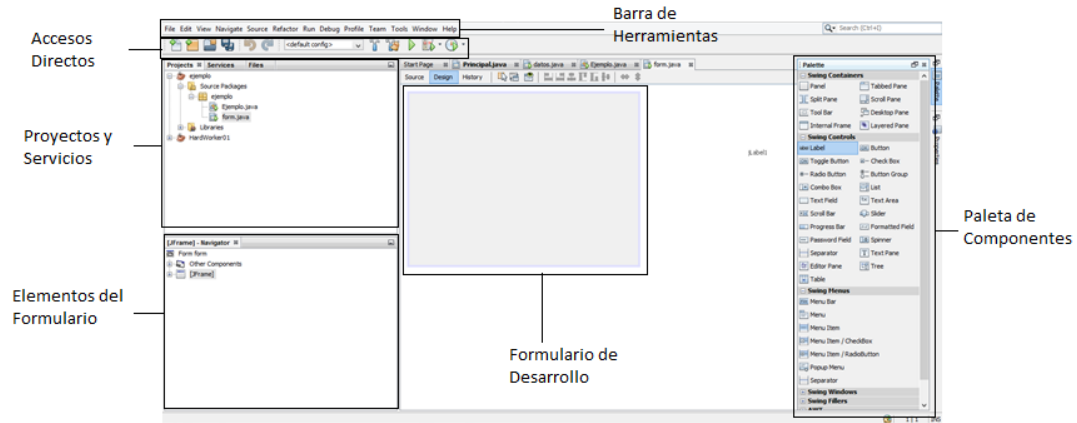


Figura 4.10. Distribución de Herramientas Netbeans.

En la sección derecha de la figura anterior se puede observar la paleta de componentes de Netbeans, los cuales deben ser arrastrados hacia el Formulario de Desarrollo para realizar la distribución adecuada de los mismos. De igual manera, se puede añadir eventos a estos componentes según sea necesario. De esta manera se fue dando forma a lo que hoy es la interfaz gráfica de MI-T-NATOR, buscando sencillez y funcionalidad en cada operación que realice el operador. En la Figura 4.11 se muestra la HMI en su forma completa.



Figura 4.11. HMI Completa.

Como se observa en la figura 4.11 se ha estructurado la aplicación mediante paneles, para de esta forma optimizar la distribución del espacio. Cada panel es un contenedor dentro del cual se colocan los distintos componentes de la aplicación, agrupándolos dependiendo del panel en el que se encuentren. El diseño de configuración es libre de manera que Netbeans no trate de aplicar plantillas de diseño pre-establecidas. A continuación se explican los distintos paneles, así como las funciones que cada uno de los componentes poseen.

4.2.2.1. Panel de Presentación.

En este panel se colocan los datos informativos del proyecto, la institución para la cual es desarrollada la aplicación, nombres de los autores y el título del proyecto. Para esto solo se usan etiquetas alfanuméricas, y se agrega también el logo de la Escuela Politécnica del Ejército a una de ellas.



Figura 4.12. Panel de Títulos.

4.2.2.2. Panel Esquema de Trayectoria.

En este panel se indica la trayectoria por la cual MI-T-NATOR debe realizar la fumigación. Para ello simplemente se carga una etiqueta con la imagen de la trayectoria, como se muestra a continuación.



Figura 4.13. Panel Esquema de Trayectoria.

4.2.2.3. Panel MI-T-NATOR.

Este panel muestra una representación de la plataforma total, para poder visualizar el prototipo en su forma física, y así dar al operario una idea de cómo debe realizar los movimientos del robot. El panel se muestra en la figura 4.14.



Figura 4.14. Panel MI-T-NATOR.

4.2.2.4. Panel Selección de Modo.

En este panel se selecciona el tipo de modo en el que va a operar, a más de indicar e momento en que debe iniciar el modo automático. Cuando en este panel, se selecciona el modo manual, se activan los eventos de teclado que se han mencionado anteriormente en este capítulo. El modo de selección se aprecia en la figura 4.15.

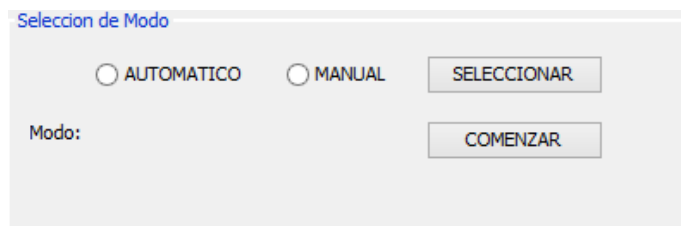


Figura 4.15. Panel Selección de Modo.

La selección del modo de operación se realiza mediante el botón seleccionar, que dependiendo de qué modo este seleccionado, enviara las órdenes hacia el controlador de MI-T-NATOR. Cuando se inicie la interfaz, el botón COMENZAR de este panel estará oculto, y solo se activara cuando la selección sea el modo automático.

4.2.2.5. Panel Control Modo Manual.

Este panel inicialmente está oculto, pero cuando se selecciona el modo manual, se activa y permite controlar el robot mediante los botones que se encuentran en él. Estos botones tienen las mismas acciones que los eventos de teclado. Es decir, cada evento de botón envía un dato al controlador de la plataforma para realizar determinada acción.



Figura 4.16. Panel Control Modo Manual.

En la parte izquierda de este panel se aprecian los controles de movimiento de la plataforma, y al lado derecho aparecen los botones de control de aspersion, como se observa en la figura 4.16.

4.2.2.6. Panel de Control Estado Robot.

Esta pantalla (figura 4.17) nos indica el estado del contenedor de fungicida, datos que son recibidos desde el controlador para ser mostrados al operador. Adicionalmente, se tiene un botón de parada de emergencia, el cual debe ser activado por el operador, sin importar en qué modo de operación se encuentre el prototipo.



Figura 4.17. Panel Estado de Robot.

Este modo nos indicara en qué nivel de fungicida se encuentra el contenedor, para de esta forma saber en qué momento es necesario recargar el contenedor, y así precautelar la integridad de la bomba y no forzarla.

4.2.2.7. Barra de Archivo.

En esta barra se encuentran dos funciones, una de las cuales nos permite desplegar la función reportes para poder visualizar la base de datos, para poder llevar un control del día, hora y modo de operación. También es posible salir de la aplicación con el botón salir.

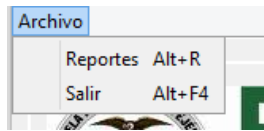


Figura 4.18. Barra de Archivo.

Esta barra de menús también se caracteriza por [permitir acceder a estas funciones mediante accesos directos, como se aprecia en la figura 4.18. Al seleccionar la opción Reportes, se despliega una segunda pantalla en la que se puede realizar la consulta de la base de datos.

4.2.2.8. Pantalla de Reportes.

En esta pantalla es posible visualizar ciertos datos de interés en cuanto a la fumigación del número de camas que se ha estado realizando a través del tiempo, esto a fin de verificar también qué modo es el que más se usa. En la siguiente figura se observa cómo se distribuye la ventana de reportes.

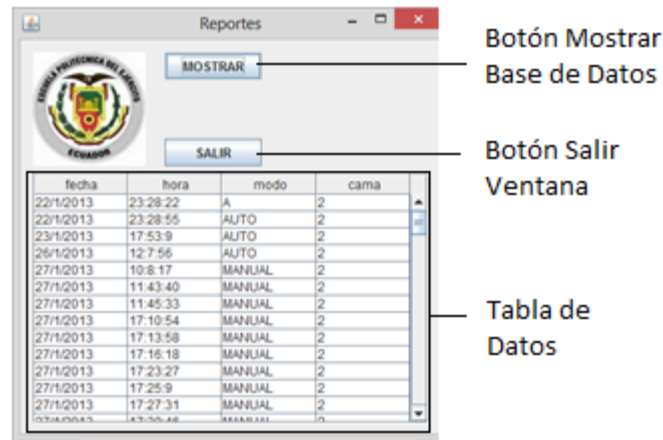


Figura 4.19. Ventana de Reportes.

Esta pantalla tenemos dos botones de acción, con los cuales se puede salir de la ventana o mostrar los datos accediendo a la base donde se encuentran almacenados. A más de esto también posee un área en la cual se carga la tabla de datos de los eventos que han ido sucediendo a lo largo del tiempo. A fin de que sirvan de herramienta para organizar los cronogramas de fumigación en las distintas camas de rosas.

4.3. DESARROLLO DE LA BASE DE DATOS

La base de datos se encuentra elaborada bajo el motor de datos MySQL, esto dado que es un software de libre acceso, y de gran capacidad, idóneo para las características que el presente proyecto debe soportar.

La base de datos es por demás simple, dado que únicamente se usa una tabla a la cual se envían todos los datos que son de interés para el operario. Esta tabla tiene los atributos mostrados en la tabla 4.02.

FECHA
HORA
MODO
CAMA

Tabla 4.02. Atributos de la Base de Datos.

Para realizar la conexión entre Netbeans y MySQL, es necesario agregar a las librerías del proyecto, el conector adecuado, que es una librería .jar que se encargara de dar la ruta de conexión entre Netbeans y la base de datos. A continuación se muestran el modo de realizar la conexión en Netbeans, y cómo enviar los comandos a MySQL.

```
try{
    //Para establecer el mod de la tabla
    DefaultTableModel mod = new DefaultTableModel();
    this.tabla.setModel(mod);

    //Coxexion a base de Datos
    DriverManager.registerDriver(new com.mysql.jdbc.Driver());
    Connection conexion =
    DriverManager.getConnection("jdbc:mysql://localhost/fumigador", "root", "Isaac2004");

    //EjecutarConsulta
    Statement stmt = conexion.createStatement();
    String sentencia ="SELECT * FROM datos;";
    ResultSets=stmt.executeQuery(sentencia);

    //Obteniendo las columnas consultadas
    ResultSetMetaDatarsMd = rs.getMetaData();
```

```
intnumcolumn = rsMd.getColumnCount();

//TituloColumnas
for (inti = 1; i<= numcolumn; i++) {
    mod.addColumn(rsMd.getColumnLabel(i));
}

//Filas de Tabla
while (rs.next()) {
    Object[] fila = new Object[numcolumn];

    for (inti = 0; i<numcolumn; i++) {
        fila[i]=rs.getObject(i+1);
    }

    mod.addRow(fila);
}

//Cierre de Conexion
rs.close();
conexion.close();

}
catch(SQLException ex) {
    System.out.println(ex.getCause());
}
```

Como se puede apreciar en el código anterior, todas las sentencias se encuentran encerradas en la función try, para de esta manera identificar si es que se produjo algún error en determinado punto de la conexión o en la sintaxis de código, ya sea en Netbeans o en MySQL.

CAPÍTULO 5

IMPLEMENTACIÓN

Después de realizar el análisis de los requerimientos que necesita MI-T-NATOR para su operación, se realiza la implementación física de las distintas etapas del prototipo. Todo esto en base a los resultados que se han ido obteniendo a lo largo de los capítulos del presente proyecto.

5.1. CONSTRUCCIÓN MECÁNICA

La construcción mecánica está ligada al Capítulo 2 del proyecto, ya que de él se desprenden los distintos elementos mecánicos a ser implementados. Todo este análisis ya se lo ha realizado anteriormente, por lo cual la presente sección únicamente se enfoca en el modo en el cual se realizó el ensamblaje de las distintas piezas mecánicas, cuya construcción se la realizo en un taller metalmecánico con la ayuda de tornos, fresas y soldadoras.

5.1.1. Selección de Materiales

Como se explicó en el capítulo 2, la estructura mecánica está compuesta de materiales metálicos en su totalidad, esto dado la resistencia que tienen los mismos. En un análisis a priori,

CAPÍTULO 5: IMPLEMENTACIÓN

se tomó en cuenta que el peso de la plataforma cargada de fungicida seria de aproximadamente 30 kg.

Por este motivo, se escogió al aluminio como el material más adecuado para la aplicación, toda vez que sus láminas tienen distintos grosores, es de fácil maleabilidad, bajo costo y sobre todo el poco peso que tiene a pesar de su alta resistencia. Por ello se especifica a continuación el modo de ensamblaje de las distintas piezas de la estructura.

5.1.2. Montaje de Piezas

Como se mencionó en el capítulo 2, la plataforma estará compuesta por 3 láminas de aluminio inoxidable de 4 líneas. Esto nos permite crear una estructura de gran resistencia, de manera que las láminas no se doblen al momento de colocar toda la carga. En la figura 5.01 se muestran las láminas que se usan.



Figura 5.01. Láminas Base.

Estas láminas están soldadas a los vértices de los ángulos que sirven de soporte para la estructura. Cada una de ellas está separada de la otra de una manera adecuada, ya que en cada

CAPÍTULO 5: IMPLEMENTACIÓN

una de estas laminas están colocados el contenedor (primer piso), la batería de alimentación (segundo piso), aspersores y electrónica (tercer piso).



Figura 5.02. Soldadura Angulo Base.

En la Figura 5.02, se puede apreciar el modo en el cual están sujetas las láminas a los vértices de los ángulos de soporte. Cada uno de estos ángulos está colocado en los distintos vértices de las bases, y cada una de estas uniones posee tres puntos de suelda para que las bases se sujeten firmemente a la estructura. Además allí se puede observar uno los orificios que se realizaron en cada lámina, de manera que sirvan de guía del cableado.

Posteriormente se colocaron en la base inferior, los vértices que sirven para sujetar el contenedor en cada una de sus esquinas, como se aprecia en la figura 5.03. Dichos vértices están soldados al primer piso de la plataforma, de manera que el contenedor se ubique centrado.



Figura 5.03. Vértices Sujetadores Contenedor.

Bajo la misma base inferior se colocaron las abrazaderas que sujetaran a cada uno de los motores. Estas están colocadas una a cada lado de la base y en el extremo posterior de la misma, ya que al tener tracción posterior, hará que el sistema sea más estable cuando este en movimiento. En la Figura 5.04 se aprecia cómo se encuentran colocadas físicamente estas abrazaderas.



Figura 5.04. Abrazaderas de motor.

CAPÍTULO 5: IMPLEMENTACIÓN

Las abrazaderas se encuentran fijadas a la base de aluminio por dos puntos de suelda que dan más soporte y fijación a los motores y a los componentes que están sujetos a ellos. Además de esto, las abrazaderas poseen un tornillo roscado con una tuerca para que sirvan de ajuste a los motores. En el eje del motor se encuentra colocado el acople para la llanta. Este acople posee un orificio roscado sobre superficie circular para que un prisionero también roscado hexagonal apriete al eje del motor, e impida que el eje se afloje con el movimiento.

Este acople tiene el mismo diámetro que el orificio de la llanta, haciendo que estas dos piezas se encuentren fijadas entre sí. En la Figura 5.05 se encuentra la conexión entre el eje del motor, el acople y la llanta.



Figura 5.05. Acople Llanta Motor.

En la misma base y en el extremo opuesto al que se encuentran los motores, es necesario colocar la garrucha de dirección. Esta garrucha está colocada en la mitad de la base para que dé más estabilidad a la estructura, y facilite el giro a 90 grados que frecuentemente se debe ejecutar. Esta se encuentra sujeta mediante remaches de presión, para evitar hacer orificios innecesarios en la lámina. En la figura 5.06 se muestra como se encuentra montada la garrucha de dirección.



Figura 5.06. Garrucha de Dirección.

En la segunda base de la plataforma, que está 25 cm por encima de la primera base, se encuentra colocada la fuente de alimentación. Esta fuente de alimentación es una batería de 12 voltios de corriente continua, y está sujeta a la base por dos ángulos metálicos ubicados en el centro.

Todo esto con el fin de no desestabilizar a la plataforma; los ángulos poseen tres puntos de suelda que los fijan a la base, y están separados de manera que la batería se ajusta a ellos y no tenga el más mínimo movimiento. S evidencia esto en la figura 5.07.

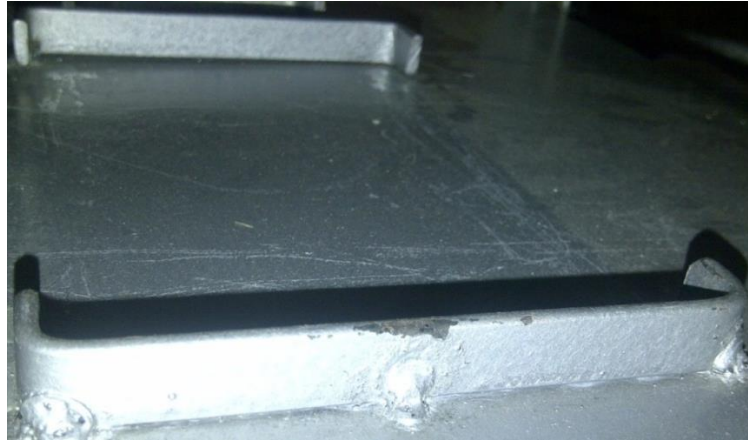


Figura 5.07. Ángulos Sujetadores.

En la base superior están ubicados los soportes de los aspersores, como se observa en la figura 5.08, así como la caja que contiene los dispositivos electrónicos de control. Los soportes de los aspersores están colocados uno a cada lado de la plataforma, a la mitad de longitud de ella, con el fin de que el fungicida se distribuya de manera uniforme a los rosales. Estos soportes están sujetos con remaches de presión a la base, dado que los puntos de suelda generarían un aspecto antiestético a la plataforma.



Figura 5.08. Soportes de Aspersores.

CAPÍTULO 5: IMPLEMENTACIÓN

En la parte posterior se encuentra ubicada a la caja de control del sistema, la cual contiene el controlador de todo el sistema. Por obvias razones, el sistema se encuentra asilado del exterior, para evitar contacto con el fungicida, y así evitar cortocircuitos. Este contenedor tiene una tapa en la parte superior del mismo, ya que el controlador requiere tener un acceso rápido y fácil en caso de que se requiera realizar algún tipo de mantenimiento, figura 5.09.



Figura 5.09. Caja de control.

Finalmente, el contenedor de fungicida está hecho de placas de acrílico blanco, cada una de las planchas están unidas a la otra por un compuesto químico especial para acrílico, y estas uniones están selladas con silicona industrial de tubería, ya que se debe evitar la filtración de fungicida hacia el exterior.

La bomba de succión, que es una bomba sumergible de gran potencia, se encuentra sujeta a uno de los vértices verticales del contenedor, de manera que uno de sus extremos succione el agua a través de una manguera plástica, y el otro extremo distribuya el fungicida hacia los aspersores. Esto se evidencia en la figura 5.10.



Figura 5.10. Bomba de Succión Sumergible.

Como se puede apreciar en la Figura 5.10, las mangueras plásticas requieren de una abrazadera metálica en toda unión que ellas tengan, dado que así se evitan fugas de líquido entre dichas uniones.

Para realizar la distribución de fungicida hacia los aspersores, se usaron conectores plásticos y metálicos con sus respectivas abrazaderas. Los neoplos usados fueron de $\frac{1}{4}$ de pulgada y se cambiaban a $\frac{1}{2}$ pulgada en la T de distribución, dado que esa es la medida de las electroválvulas que se usan para activar los aspersores.

Para impermeabilizar este sistema, a más de las abrazaderas, se usó sellador de rosca para eliminar cualquier filtración el momento de aplicar presión hidráulica a las mangueras. En la figura 5.11 se muestra como se encuentra ubicada la T de distribución.



Figura 5.11. Sistema de Distribución de Fungicida.

Cada una de estas mangueras provenientes de la T de distribución están conectadas a las electroválvulas por medio de nepllos metálicos sellados con aislador de rosca y abrazaderas en cada una de las uniones neplo-manguera, como se muestra en la figura 5.12.

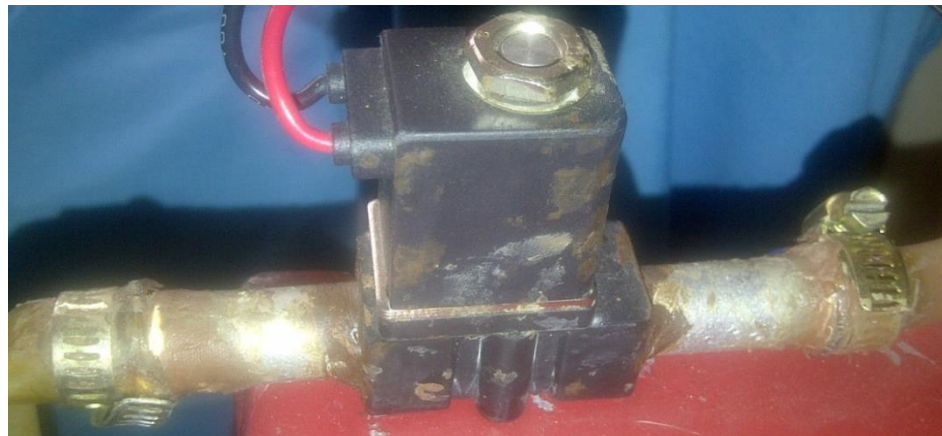


Figura 5.12. Conexión Electroválvula.

Luego de la conexión de la válvula, la manguera de salida se conecta con los aspersores. En ellos se utiliza nuevamente el sellante de rosca y abrazaderas, que son la manera más

CAPÍTULO 5: IMPLEMENTACIÓN

económica y efectiva de impermeabilizar las conexiones hidráulicas. En la figura 5.13 se muestra la conexión entre las mangueras y los aspersores.



Figura 5.13. Conexión Aspersor Manguera.

Como se puede apreciar en la Figura 5.13, los aspersores son de tipo plano movibles de manera que sea fácil dar una dirección específica de aspersión. La configuración plana permite que la aspersión se realice en forma de sombrilla, ayudando a cubrir más rosales con menos líquido.

5.2. CONSTRUCCIÓN E INTEGRACIÓN ELÉCTRICA Y ELECTRÓNICA

Para la integración Eléctrica y Electrónica, se decidió usar una placa profesional con pistas en ambos lados, y utilizando borneras de conexión, ya que usar “proto board” a más de ser

CAPÍTULO 5: IMPLEMENTACIÓN

antiestético, representaba un desorden por la cantidad de cables que se debía usar. Por ello, el diseño de la placa de control básica se detalla en la figura 5.14.

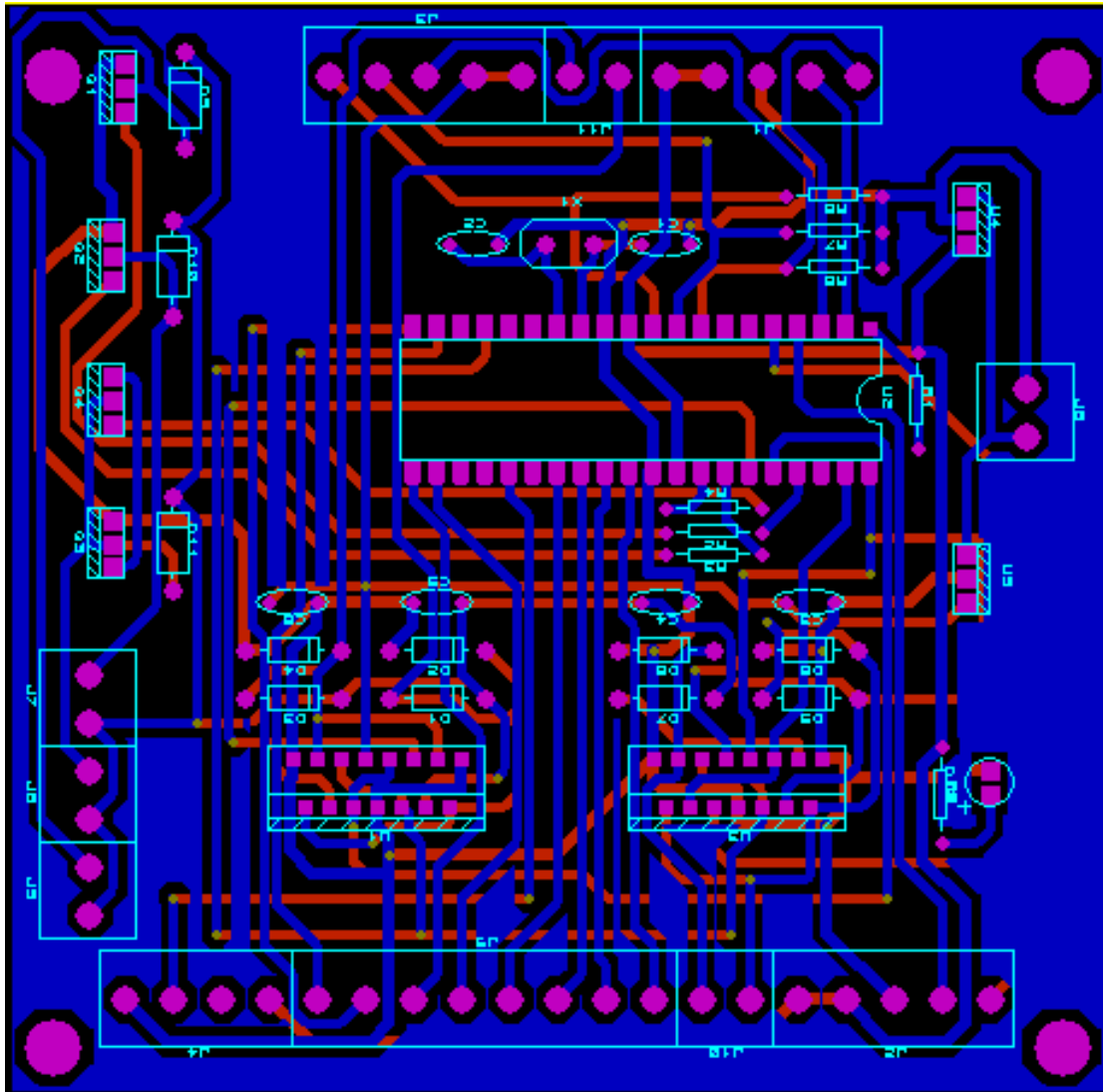


Figura 5.14. Diseño Placa de Control.

Luego de realizar el diseño de la placa de control, se procedió a la implementación de la misma, la cual contiene una película plástica protectora y puntos de suelda de tamaño adecuado, dada la corriente que debe usar.

CAPÍTULO 5: IMPLEMENTACIÓN

El diseño posee borneras extras en el caso de que se requiriera expandir el sistema y conectar más entradas o salidas al controlador. En la Figura 5.15 se muestra la placa de control fabricada y con todos sus elementos colocados en su sitio.

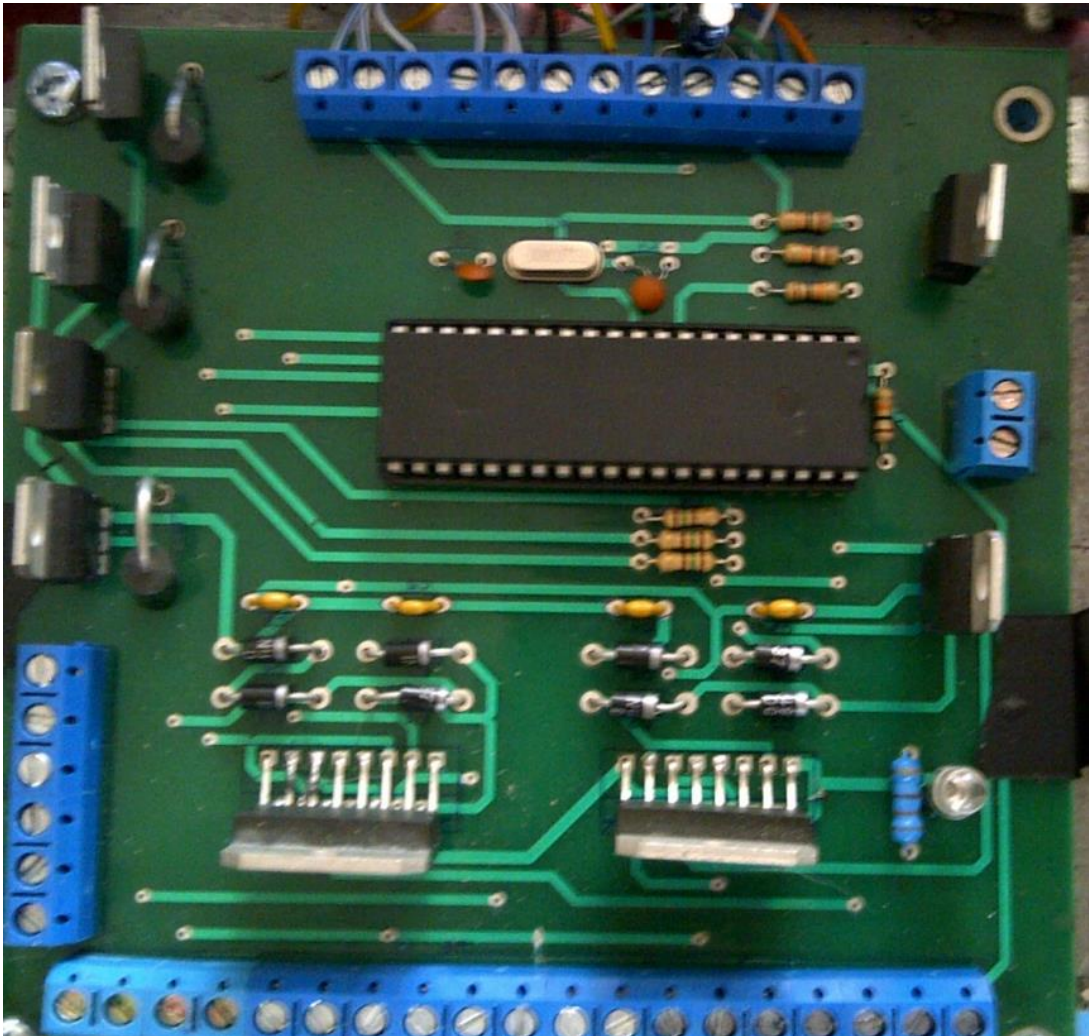


Figura 5.15. Placa de Control Implementada.

A esta tarjeta de control se encuentran conectadas las etapas eléctrica y electrónica del sistema robótico, todo el conjunto de etapas hace posible el funcionamiento integral de MI-T-NATOR. Por ello a continuación se detalla el dimensionamiento de los tipos de conectores eléctricos que se deben usar para cada una de las conexiones.

CAPÍTULO 5: IMPLEMENTACIÓN

5.2.1. Conexión de Alimentación General.

Este tipo de conexión es fundamental, y constituye la de mayor importancia dado que de ella depende la alimentación de todos los componentes eléctricos y electrónicos del sistema. Además se debe considerar la cantidad de corriente que entrega la fuente de alimentación, que es de 10 Amperios/Hora, lo cual es una cantidad de corriente considerable, tomando en cuenta todos los dispositivos que debe activar y la corriente total que necesita distribuirla a través de los cables.

El cable adecuado para estas conexiones se decidió que sea un cable multifilar AWG 22, ya que este cable soporta sin problemas la cantidad de corriente que suministrara la batería, sin causar pérdidas por calor dado su espesor y su constitución multifilar.

5.2.2. Conexión de Motores.

Los motores están conectados a la placa de control por medio de cable sólido AWG 25, ya que los motores requieren una gran cantidad de corriente que es entregada por los drivers, y que si el medio conductor no posee ciertas características, provocaría pérdidas por calentamiento. Este tipo de cable generalmente es usado en la conexión de timbres caseros, y dado que ellos soportan 110 voltios a una gran cantidad de corriente, se decidió usarlos para alimentar los motores, tomando en cuenta el bajo costo que el metro de este cable tiene.

5.2.3. Conexión de Transmisión de Datos.

Para esta sección se usa el cable de conexión de par trenzado UTP categoría 2, ya que este tipo de cable se lo utiliza en transmisión de datos de baja tasa de transferencia en datos.

CAPÍTULO 5: IMPLEMENTACIÓN

Cabe recalcar que a este cable se retira su cubierta ya que no se necesita el total de pares trenzados ni un conector RJ-45.

5.2.4. Conexión de Electroválvulas y Bomba de Succión.

Para la conexión de las válvulas, se utiliza nuevamente el cable AWG 22 multifilar, dadas las condiciones de corriente que cada uno de estos elementos necesitan, especialmente de la bomba de succión cuya corriente nominal bordea los 2 amperios.

5.2.5. Conexión de Sensores.

Los sensores entregan señales analógicas, y transportan rangos de voltaje entregados por dichos sensores que no tienen gran cantidad de corriente. Por lo cual, no es necesario sobredimensionar estos conectores, y se usa los mismos cables UTP de categoría 2 usados en la comunicación de datos.

5.3. PUESTA EN MARCHA.

5.3.1. Campo de pruebas.

Para la puesta en marcha del proyecto, se debió iniciar por la construcción del campo de pruebas en condiciones ideales, para que el robot se desplace por un terreno sin muchas irregularidades y que las condiciones ambientales se estabilicen.

CAPÍTULO 5: IMPLEMENTACIÓN

En primera instancia, se construyeron estructuras metálicas que sirven de soporte para las plantas que simularan los rosales. Estas estructuras además sirven como un elemento a ser medido, ya que su presencia indica al robot si se ha acercado demasiado a ese costado y deba corregir su dirección.

En las figuras 5.16 y 5.17 se muestran los planos de construcción y la representación de dichos soportes.

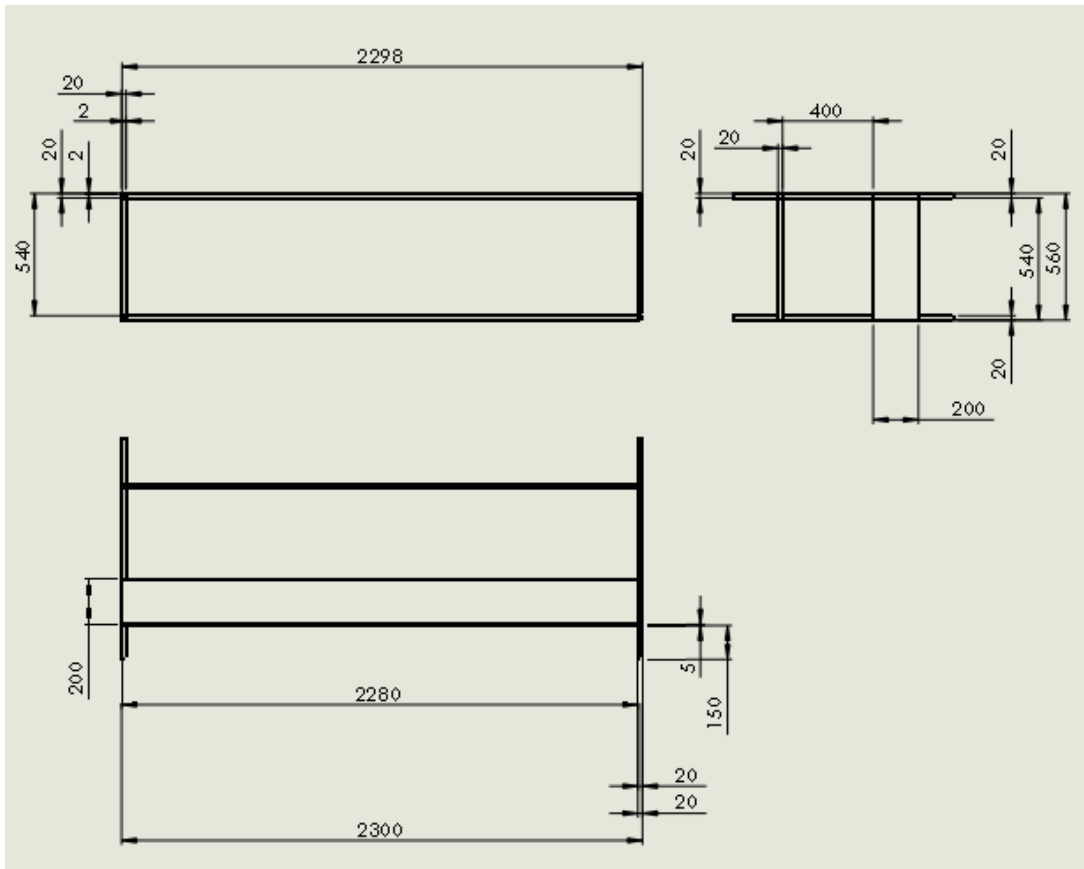


Figura 5.16. Plano de Construcción Estructuras Metálicas (mm).



Figura 5.17. Representación Estructuras Metálicas.

Se construyeron cuatro estructuras de este tipo, de manera que pueda cubrir los 5 m de largo que tendrán las camas simuladas. A más de esto, se encuentran colocadas dos masetas paralelas con las dos restantes, de manera que el campo tenga dos camas de 5 metros cada una.

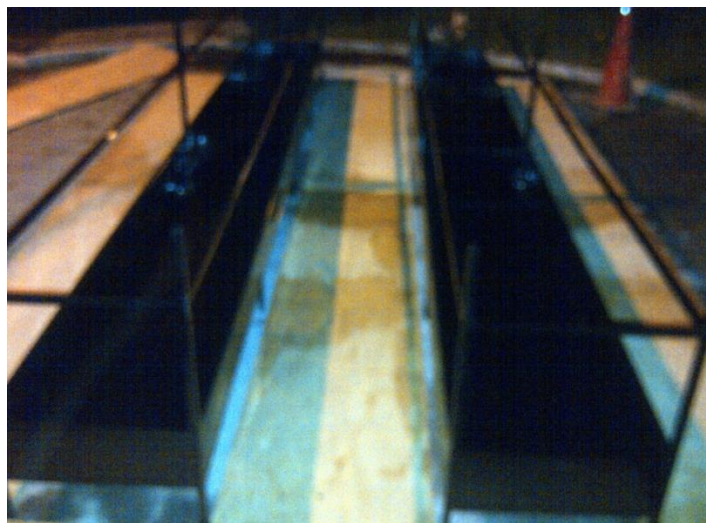


Figura 5.18. Distribución de las Masetas en el Campo de Pruebas.

CAPÍTULO 5: IMPLEMENTACIÓN

En la Figura 5.18 también se puede observar la colocación de tablas de madera sobre el piso del campo, esto es para simular las condiciones ideales de un piso liso por donde debe moverse el robot. Estas tablas son de tipo industrial y en la figura 5.20 se muestra su plano de dimensiones y su representación en tres dimensiones.

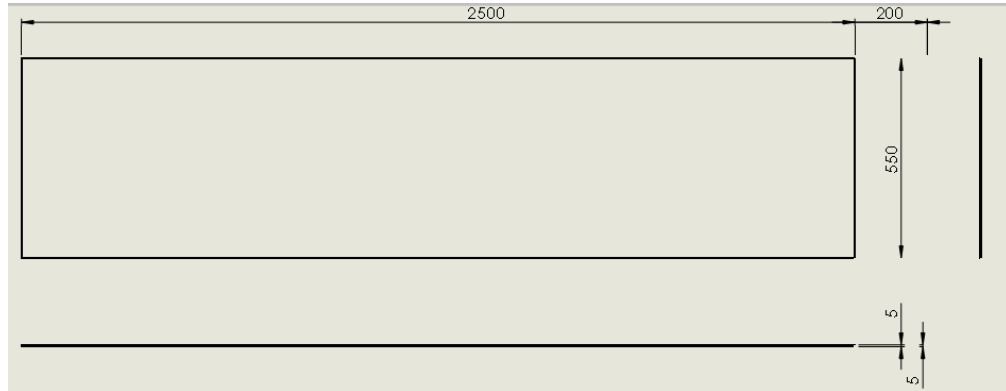


Figura 5.19. Plano de Construcción Tabla de Madera (mm).

Estas tablas están colocadas entre las camas de rosas, y van sujetas a las masetas metálicas de manera que no se doblen por efecto del líquido fungicida, que en cierta cantidad caerá sobre ellas. También se encuentran colocadas tablas perpendiculares a estas en los extremos de las masetas, dado que el robot realizara giros y desplazamientos entre camas por estos sectores.

A más de esto también se encuentran colocadas tablas verticales en el costado opuesto a las masetas metálicas, para generar un límite que debe ser censado por el robot, y de esta forma sirvan de guía para el movimiento del robot, como se muestra en la figura 5.20.

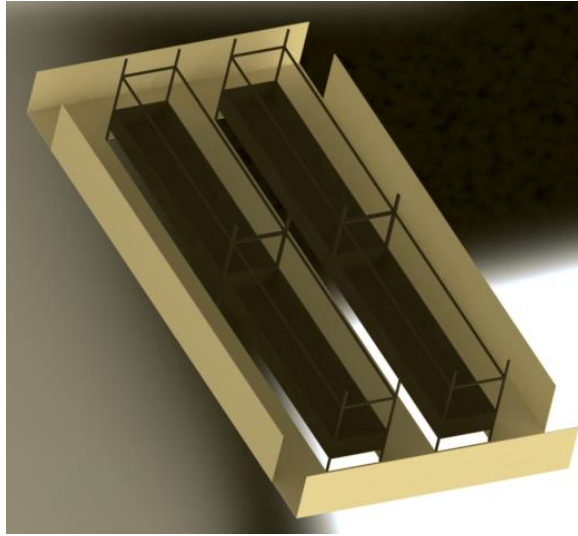


Figura 5.20.Representacion del Campo Total.

5.3.2. Plataforma.

Para la puesta en marcha de la plataforma, se realizaron los pasos antes mencionados, y además se realizaron pruebas reales de las dimensiones de la plataforma con las dimensiones del campo de pruebas, logrando determinar que las medidas son adecuadas para el desplazamiento del robot.

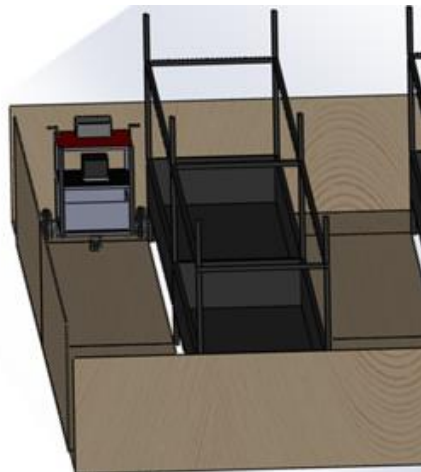


Figura 5.21. Representación de la Plataforma en el Campo.

CAPÍTULO 5: IMPLEMENTACIÓN

En la figura 5.21 se aprecia una representación de las dimensiones de la plataforma versus el campo de pruebas. Como se puede observar, las dimensiones en el modelo tridimensional son adecuadas, por lo cual en la realidad tampoco se presentaron inconvenientes de esta índole, como se evidencia en la figura 5.22.



Figura 5.22. Plataforma Sobre el Campo.

En cuanto a la parte electrónica, luego de realizar el cableado de cada uno de los elementos hacia la tarjeta de control, se verificó la continuidad de cada uno de los cables para de esta forma eliminar posibles fallas por pérdida de comunicación.

Para los sensores de nivel, se usaron láminas de cobre en el extremo que va dentro del tanque para mejorar la estética de los mismos, así como de recubrir sus cables con termo fundente de uso profesional para que sea una capa aislante de los cables. En la Figura 5.23 se observa cómo se colocaron las láminas conductoras dentro del contenedor, y en la figura 5.24 se aprecia la colocación del termo fundente en los cables a ser sumergidos.



Figura 5.23. Láminas de Sensores de Nivel.

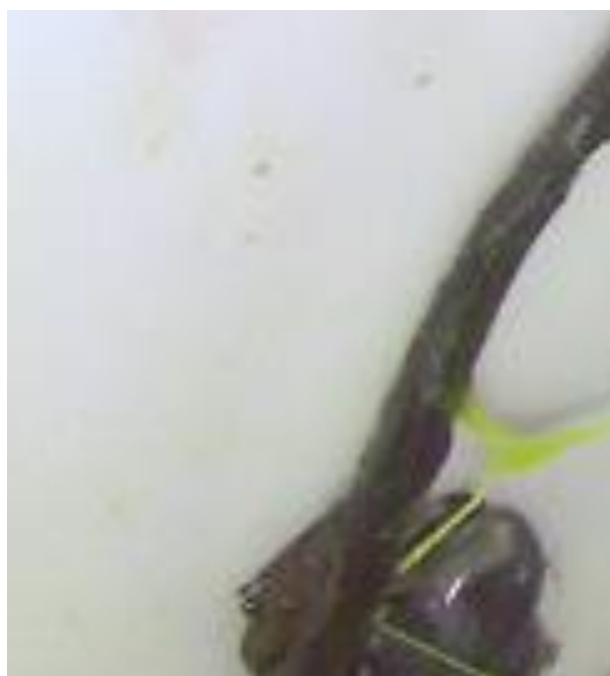


Figura 5.24. Termo Fundente aislante.

En los bornes de la batería de alimentación, se colocaron protectores plásticos para evitar algún contacto con líquido que se pudiese presentar, lo cual ineludiblemente afectaría a todos los

CAPÍTULO 5: IMPLEMENTACIÓN

elementos eléctricos y electrónicos conectados a la red de energía de esta batería. En la figura 5.25 se observa los protectores de bornes de la batería.

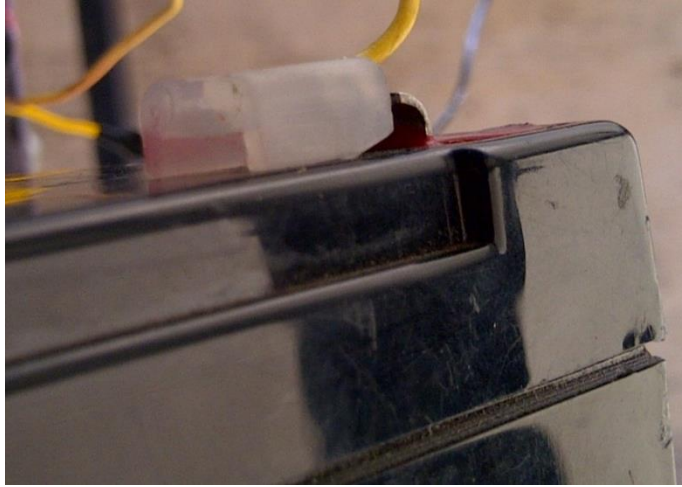


Figura 5.25. Protectores de Bornes de Batería.

Finalmente se colocaron amarras de sujeción en varias partes de la plataforma de manera que sirvan de guía del cableado, y para sujetar la manguera de distribución. Como se aprecia en la figura 5.26.



Figura 5.26. Amarra de Sujeción.

CAPÍTULO 5: IMPLEMENTACIÓN

En aquellas partes en donde el cableado no podía ser ocultado, se colocó espiral plástico para dar una apariencia profesional a la plataforma. Esto se evidencia en las figuras 5.27, 5.28, 5.29.

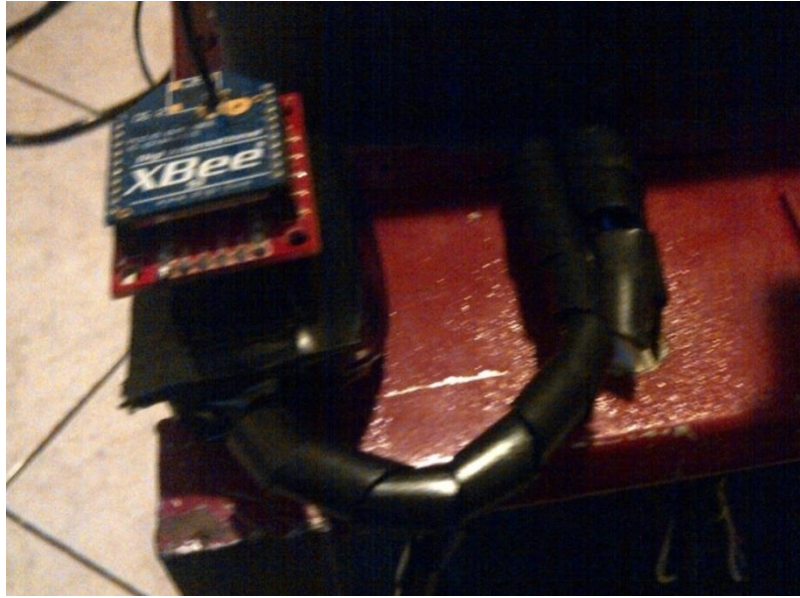


Figura 5.27. Espiral Protector Modulo Xbee PRO y Tarjeta de Control.

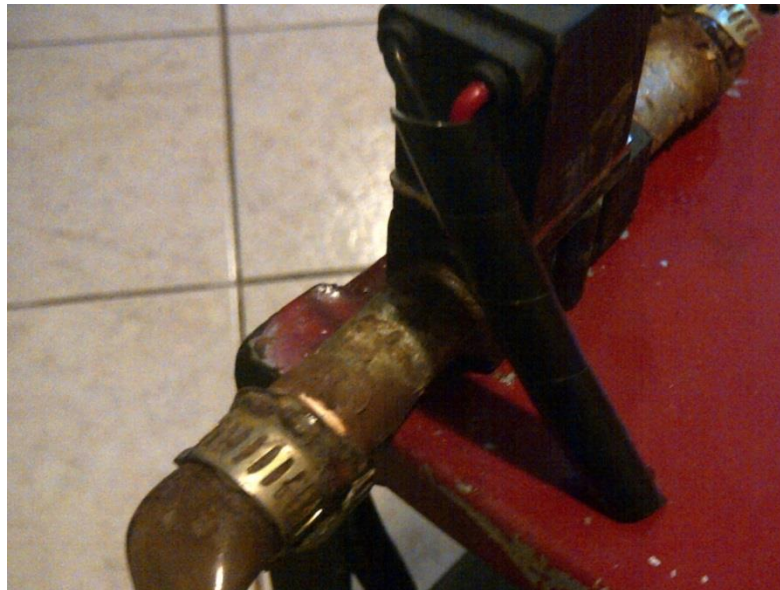


Figura 5.28. Espiral Protector Válvulas de Distribución.



Figura 5.29. Espiral Protector Sensores Infrarrojos.

En la figura 5.30, se puede observar el estado final de MI-T-NATOR.

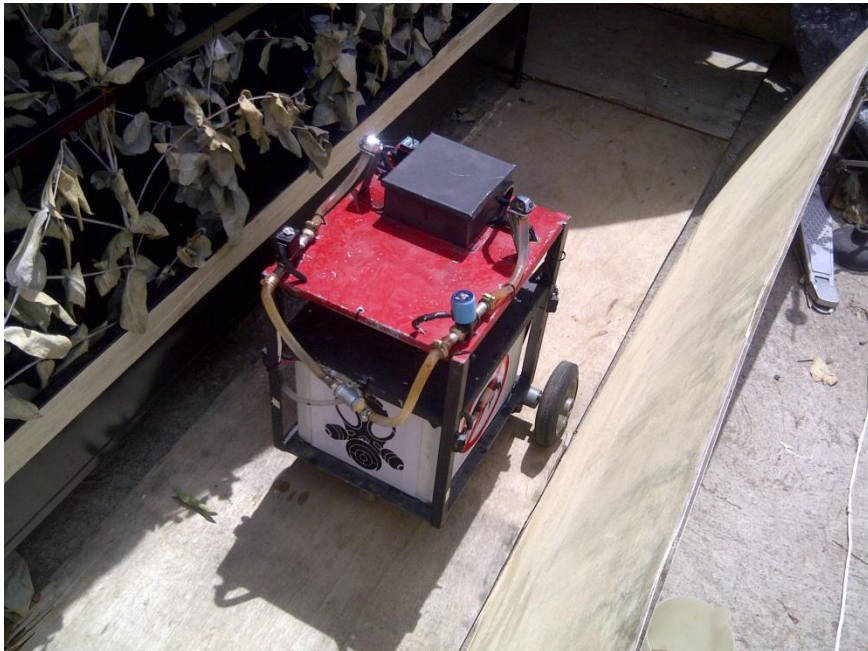


Figura 5.30. Estado Final MI-T-NATOR.

CAPITULO 6

PRUEBAS Y RESULTADOS

6.1. PRUEBAS

Luego de haber implementado tanto el robot como el campo de pruebas, se procedió a realizar las pruebas de rigor para verificar el correcto funcionamiento. Se realizaron pruebas de cada una de las etapas del sistema por separado, para posteriormente realizar pruebas del sistema integrado en uno solo.

6.1.1. Pruebas del Sistema Mecánico.

Luego de implementar el sistema mecánico, era necesario realizar pruebas de movilidad para verificar el correcto desplazamiento en el terreno sobre el cual el robot debe moverse. Para ello, en un principio se realizaron pruebas sin elementos electrónicos de ningún tipo.

De esta forma, se logró evidenciar algunas fallas en el desplazamiento de la plataforma. Pese a ser un terreno casi en su totalidad lizo, la plataforma realizaba un viraje hacia el lado

CAPÍTULO 6: PRUEBAS Y RESULTADOS

derecho, el cual era sumamente notorio y por obvias razones podía desencadenar un problema serio al momento de realizar las pruebas integrales. En la figura 6.01 se evidencia el problema.



Figura 6.01. Falla en la Dirección de la Plataforma sin Carga.

Este inconveniente de dirección también afectaría al giro perpendicular que MI-T-NATOR debe realizar para desplazarse entre las camas. Todo esto pese a que los motores de corriente continua tienen las mismas características tanto de torque como de revoluciones, pero esto en la práctica se evidenció que no era cierto. Es claro que las máquinas eléctricas a pesar de ser construidas bajo las mismas normas y características jamás podrán ser totalmente idénticas.

Por ello, se tomó la decisión de solucionar este inconveniente de manera electrónica, ya que una solución mecánica implicaba un rediseño total del sistema mecánico de la plataforma ya implementada, y con ello una pérdida de tiempo considerable en el proyecto, lo cual bajo ningún punto de vista era posible.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Se evidencio además, que el sistema mecánico reducía esta falla de viraje en el momento en que el contenedor de fungicida estaba con su carga máxima, esto debido a que era más difícil para los motores mover la el peso de la plataforma adicionado con el líquido. En la figura 6.02, se aprecia el viraje que presenta la plataforma cuando está cargada al máximo.



Figura 6.02. Falla en la Dirección de la Plataforma con Carga Máxima.

La velocidad de la plataforma estando cargada al máximo es adecuada, a pesar de no usar el cien por ciento de la velocidad que proveen los motores. Estas pruebas se las realizaron colocando a la plataforma en un terreno lizo en el cual se desplazó a gran velocidad estando con el contenedor lleno.

Por lo cual, para los giros perpendiculares se determinó que la mejor opción era alimentar a los motores en sentido contrario uno del otro, de manera que también se minimice el esfuerzo de ellos.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

6.1.2. Pruebas del Sistema de Fumigación.

Gracias a que el sistema de fumigación se lo implemento con materiales de alta calidad, se logró realizar la distribución de agua de manera que no se presenten fugas de ninguna índole especialmente en las juntas y uniones de las mangueras que van desde la bomba hacia las válvulas, y desde estas a los aspersores.

En esta etapa, se debió añadir a los neoprenos de conexión entre las mangueras y las válvulas abrazaderas de presión metálicas, para de esta forma minimizar pequeños goteos que surgieron en las pruebas iniciales. En cuanto a los aspersores, se debió colocar una gran cantidad de teflón y una mezcla de sello rosca con adhesivo industrial para fijar los aspersores a las mangueras, toda vez que los diámetros entre ambos no eran coincidentes. En las figuras 6.03, 6.04 y 6.05 se observa el funcionamiento de los aspersores, juntas de las válvulas, T de distribución y se evidencia que no existen fugas de líquido hacia el exterior.



Figura 6.03. Funcionamiento de Aspersores.



Figura 6.04. Funcionamiento de Juntas de las Válvulas.



Figura 6.05. Funcionamiento T de Distribución.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

6.1.3. Pruebas de Sensores.

6.1.3.1. Sensores Infrarrojos.

Los sensores usados, como se explicó en el capítulo 3 del presente proyecto, son sensores que miden el ángulo de reflexión de un haz infrarrojo sobre un objeto, pero son sumamente susceptibles a cambios climáticos como variaciones de luz, y vientos extremos. Por lo cual fue necesario estabilizar las condiciones lumínicas para simular de mejor manera las condiciones de un invernadero. En la figura 6.06 se aprecia la carpa plástica de protección que se instaló sobre el campo de pruebas para minimizar el efecto de la luz solar sobre los sensores.



Figura 6.06. Carpa para limitar la Cantidad de Luz Solar en el Campo.

Un ejemplo de esto se presentó cuando se realizaron pruebas del sistema en la mañana con plena luz solar. Los sensores fueron calibrados para estas condiciones, pero en horas de la tarde sin tanta luz solar, los sensores presentaban una configuración totalmente distinta respecto de la mañana. Algunos datos recopilados se muestran en la tabla 6.01.

Tabla 6.01. Valores de Medición de Distancia de los Sensores en la Mañana y en la Tarde.

Distancia [cm]	Voltaje en la Mañana [V]	Voltaje en la Tarde [V]
0	0	0
10	1.23	1.40
20	0.67	0.80
30	0.41	0.53

Otro factor a considerar y que el mismo fabricante menciona en las hojas técnicas de estos sensores, es que cada valor de voltaje representa dos distancias. Esto sucede dado que la curva de transferencia de los sensores Sharp tiene una forma sinusoidal, por lo cual se debe verificar que el voltaje entregado por el sensor corresponde a la distancia requerida.

Por ello, los sensores se deben calibrar adecuadamente para que los valores de distancia que entreguen al microcontrolador sean los indicados y no exista una inconsistencia entre el valor de voltaje y la distancia necesaria. En la figura 6.07 se muestran la curva de transferencia de este tipo de sensor, y se observa el fenómeno que se ha mencionado con anterioridad.

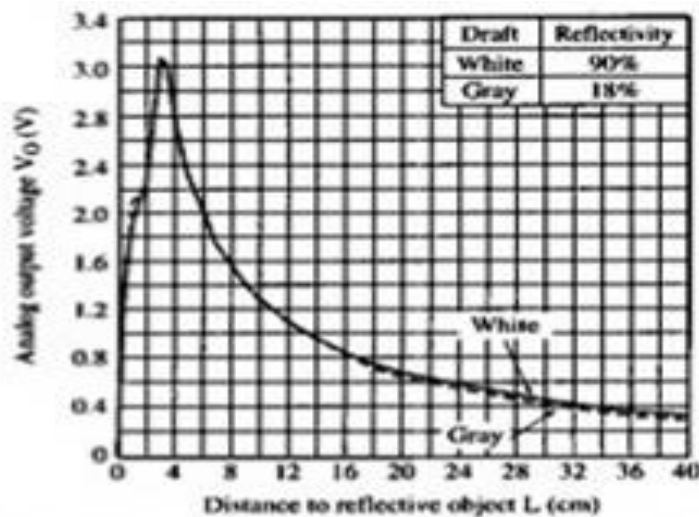


Figura 6.07. Curva de Transferencia de los Sensores Infrarrojos Sharp (Datasheet-Sharp).

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Además de estos factores, los sensores presentaron fallas en la detección de objetos metálicos de tonalidad oscura, ya que el haz infrarrojo rebotaba de manera cambiante sobre la superficie de estos objetos.

Esto hacia que el ángulo de reflexión varíe y no sea un valor constante a la misma distancia del sensor. Por ello, sobre las estructuras metálicas, se colocaron tiras de papel adhesivo para eliminar este problema. Esto se evidencia en las figuras 6.08 y 6.09.



Figura 6.08. Superficie Metálica Sobre la Cual el Haz Infrarrojo Varía el Grado de Reflexión.



Figura 6.09. Papel Adhesivo colocado Sobre la Superficie Metálica.

Luego de estabilizar las condiciones físicas y ambientales de los sensores, y a sabiendas de que el prototipo se lo usara bajo condiciones de alta luminosidad natural, se realizaron pruebas de distancia de este tipo de sensor, para determinar la curva de transferencia real a la que se someterán los sensores. Esto se evidencia en la tabla 6.02.

Tabla 6.02. Valores Reales de Distancia Medidos en la Mañana.

<i>Distancia</i> <i>[cm]</i>	<i>Voltaje</i> <i>[V]</i>
3	3.08
4	2.74
5	2.30
6	1.99
7	1.75

CAPÍTULO 6: PRUEBAS Y RESULTADOS

8	1.55
9	1.4
10	1.23
11	1.14
12	1.08
13	1.00
14	0.94
15	0.90
16	0.83
17	0.79
18	0.75
19	0.71
20	0.67
21	0.64
22	0.61
23	0.58
24	0.56
25	0.54
26	0.52
27	0.50
28	0.46
29	0.44
30	0.41
31	0.4
32	0.39
33	0.37
34	0.36
35	0.36
36	0.34

CAPÍTULO 6: PRUEBAS Y RESULTADOS

37	0.34
38	0.33
39	0.32
40	0.29

Con la tabulación de estos datos, se procedió a realizar una curva del comportamiento real del sensor para verificar que los datos del fabricante se asemejan a los datos reales.

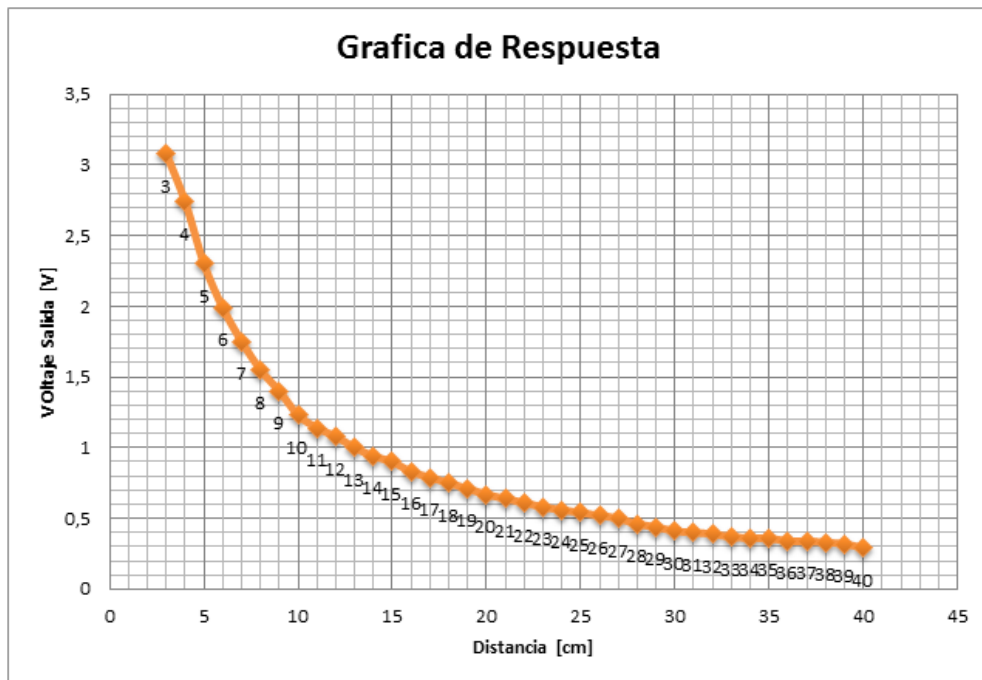


Figura 6.10. Curva de Transferencia Practica de los Sensores de Distancia.

Como podemos evidenciar en la figura 6.10, la curva de transferencia es similar a la curva entregada por el fabricante, y también se puede evidenciar que únicamente se tomaron los dato de voltaje ya estabilizados y no los primero voltajes que corresponden a distancias muy cercanas.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

6.1.3.2. Sensores de Nivel.

Dado que en el mercado no se encontraron sensores de nivel a precio accesible, se optó por diseñar sensores de nivel que aprovechen las propiedades eléctricas del agua, cuya principal característica es ser un buen conductor eléctrico. Por ello se usó el esquema electrónico que se presentó en el capítulo 3 a fin de establecer tres niveles de referencia en el líquido.

Las primeras pruebas se las realizaron comprobando si era posible encender varios diodos led cuando un par de cables se encuentren sumergidos en agua. Como se muestra en las figuras 6.11, 6.12 y 6.13.

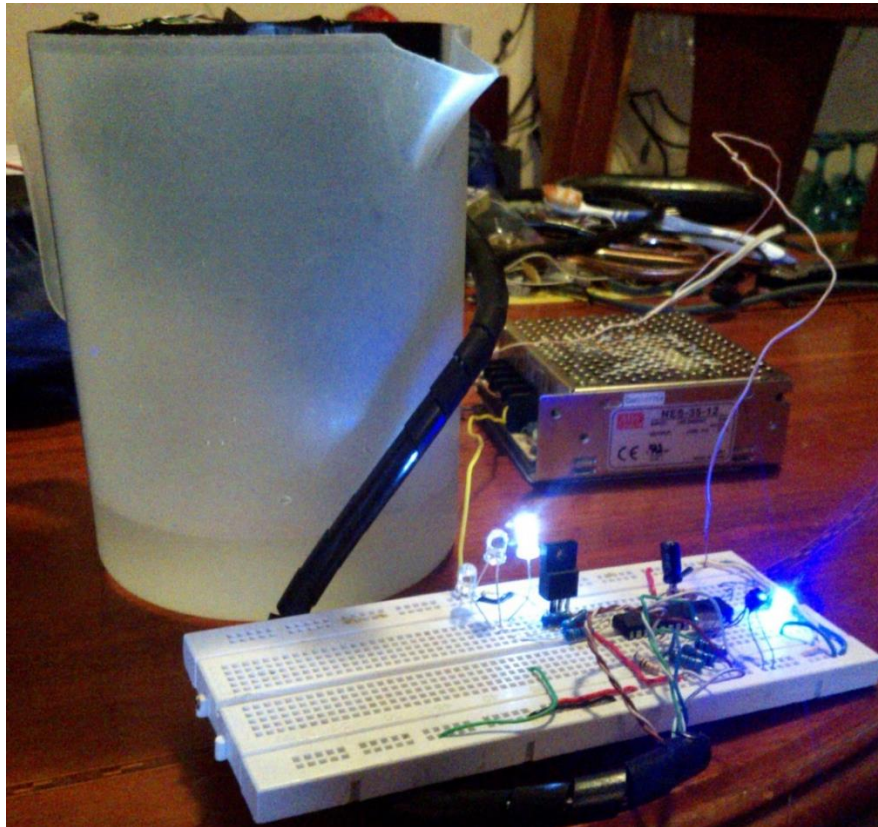


Figura 6.11. Primer Nivel Activado.



Figura 6.12. Segundo Nivel Activado.



Figura 6.13. Tercer Nivel Activado.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Como se puede observar en estas pruebas, se evidencio que si el líquido se encontraba presente entre los dos contactos de cada sensor, este provocaba un corto circuito en dicho ramal lo que provocaba que se presente un uno lógico en la compuerta del microcontrolador, haciendo que el respectivo led se encienda. En consecuencia, este principio se evidencia también en los sensores montados dentro del contenedor de fungicida. Como se aprecia en la figura 5.23.

6.1.4. Pruebas de los Módulos Serie Inalámbricos Xbee Serie 1.

Los módulos Xbee Serie 1, en un principio se debieron configurar en la pantalla X-CTU, que es el programa base, bajo el cual se enlazan ambos dispositivos, y se configura el canal de transmisión que tendrán dentro de los 12 distintos canales que ofrecen estos dispositivos que trabajan en la banda de los 2.4 GHz. En la figura 6.14 se observa la pantalla de configuración principal del X-CTU.

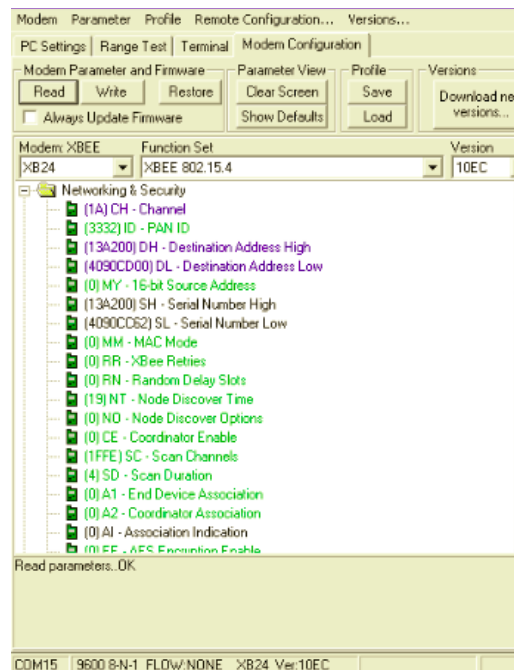


Figura 6.14. Pantalla Principal X-CTU.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Las siguientes pruebas se realizaron para medir el alcance máximo que tienen los módulos Xbee Serie 1. En primera instancia se realizaron pruebas de distancia en campo abierto, es decir que exista una línea de vista entre ambos módulos. Los resultados obtenidos fueron de aproximadamente 40 m, como se muestra en la figura 6.15.

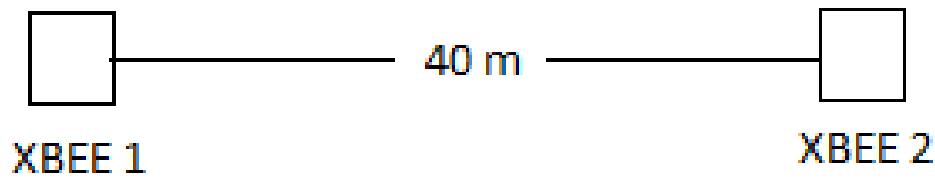


Figura 6.15. Distancia Máxima de Separación en Módulos Xbee.

Por el contrario, cuando se realizaron pruebas dentro de un edificio o una estructura de acero y concreto, los módulos no eran capaces de encontrarse entre sí. Esto se explica debido a que las señales que ambos emitían, eran reflejadas por las paredes de la estructura. Este es el único inconveniente que se presentó con los módulos seriales inalámbricos. En la figura 6.16. se evidencia a pérdida de comunicación de los módulos cuando se encuentran separados por paredes de concreto.

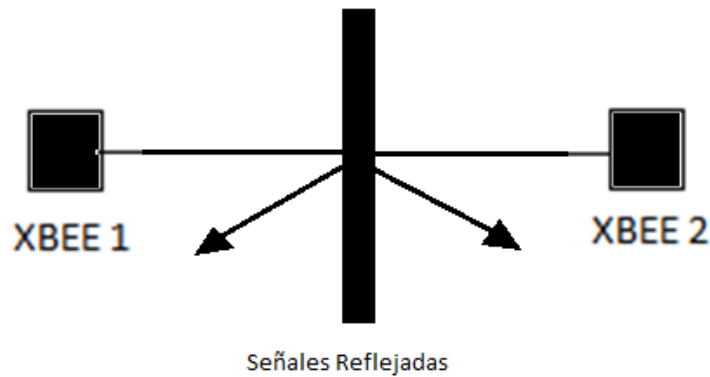


Figura 6.16. Módulos Xbee Serie 1 separados por una Pared.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Dado que para el modo manual se usa una cámara inalámbrica, que se encuentra colocada sobre la plataforma, se adquirió una cámara que no entre en la banda de los 2,4 GHz. {ese a este, aun se presentaba interferencia entre ambas señales, lo cual en el caso de los módulos Xbee Serie 1 era una condición crítica por que se podía perder datos importantes cuando el dispositivo se encuentre en marcha. Por este motivo ambos dispositivos se encuentran ubicados en extremos opuestos de la plataforma para minimizar este inconveniente. Esto se muestra en la figura 6.17.



Figura 6.17. Separación de la Cámara Inalámbrica y el Módulo Xbee Serie 1.

6.1.5. Pruebas de la Tarjeta de Control.

La tarjeta de control constituye una de las etapas más importantes dentro del proyecto, ya que de ella depende el correcto funcionamiento de la plataforma. En ella el controlador recibe órdenes desde el computador, para que la plataforma ejecute una determinada acción o proceso. Básicamente se realizaron pruebas de emisión y recepción de datos a través de los módulos Xbee Serie 1, verificando únicamente por los leds de estado de dichos módulos. Como se muestra en la figura 6.18.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Figura 6.18. Prueba de Envío y Recepción de Datos Desde y Hacia el Controlador.

Como se puede observar en la figura 6.18, los leds de estado del módulo Xbee demuestran que se realiza la transmisión de datos continua entre el controlador y la computadora. Posteriormente a esto, se debió realizar pruebas de conexión de los sistemas al controlador, evidenciando que se activen las válvulas de paso, la bomba de succión y los motores de movimiento.



Figura 6.19. Prueba de Activación de Dispositivos Eléctricos.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

En la figura 6.19 se puede observar que el controlador realiza las acciones básicas para su funcionamiento como son la activación de la bomba de succión, de ambas válvulas de paso, de los motores, comprobando de esta manera que la fuente de alimentación es adecuada para la aplicación. En la siguiente sección del presente capítulo, se evidenciará el correcto funcionamiento de los sensores de nivel y de distancia, ya que allí se realiza las pruebas integrales del sistema.

6.1.6. Pruebas de Interfaz Humano Máquina.

La interfaz humano máquina se desarrolló como se mencionó en el capítulo 4 en específicamente la acción para la que fue programado. Primero verificaremos que se seleccione adecuadamente el modo de operación, así como el envío de datos dentro de la base de datos. En la figura 6.20 se observa la selección del modo automático, con lo cual se activa el botón de arranque del sistema y se verifica mediante consola el envío de la acción tanto a la base de datos como a la plataforma a través del módulo Xbee Serie 1.



Figura 6.20. Selección de Modo Automático.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Como se aprecia en la figura 6.20, al seleccionar el modo automático, no se activan los botones de control del modo manual, más si indica al operador qué modo se ha seleccionado, y activa el botón de comienzo. Además, en consola se puede evidenciar que se han enviado órdenes a la base de datos y a la plataforma por el puerto serial COM5.

Conexión a base de datos jdbc:mysql://localhost/fumigador ..Ok

```
insert into datos values ('27/4/2013','17:7:14','AUTO','2');
```

Datos Enviados giovynet.serial.Parameters@55e10abe

COM5

El dato es: 1

Por el contrario cuando se selecciona el modo manual, se evidencia la aparición de los botones de mando en la interfaz. En este punto, también se activa el evento de teclado que permite controlar la plataforma presionando ciertas teclas de acción como se muestra en la figura 6.21. En la tabla 6.03 se muestran las teclas de acción para el modo manual.



Figura 6.21. Selección de Modo Manual.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Tabla 6.03. Teclas de Acción.

TECLA	ACCION
Flecha Arriba	Adelante
Flecha Abajo	Atrás
Flecha Izquierda	Giro Izquierdo
Flecha Derecha	Giro Derecho
S	Parar
Q	Aspersor Izquierdo
E	Aspersor Derecho
W	Ambos Aspersores
R	Desactivar Aspersores

En la pestaña archivo se encuentra la opción para verificar los datos históricos que se han venido guardando desde que se implementó el control de la plataforma por medio de la presente HMI (Human Machine Interface). Esto se evidencia en la figura 6.22.



fecha	hora	modo	cama
22/1/2013	23:28:22	A	2
22/1/2013	23:28:55	AUTO	2
23/1/2013	17:53:9	AUTO	2
26/1/2013	12:7:56	AUTO	2
27/1/2013	10:8:17	MANUAL	2
27/1/2013	11:43:40	MANUAL	2
27/1/2013	11:45:33	MANUAL	2
27/1/2013	17:10:54	MANUAL	2
27/1/2013	17:13:58	MANUAL	2
27/1/2013	17:16:18	MANUAL	2
27/1/2013	17:23:27	MANUAL	2
27/1/2013	17:25:9	MANUAL	2
27/1/2013	17:27:31	MANUAL	2
27/1/2013	17:28:16	MANUAL	2

Figura 6.22. Reportes de Base de Datos.

6.2. RESULTADOS

La presente sección trata sobre los resultados obtenidos a lo largo del proyecto. Dichos resultados, son la unión de todas las pruebas individuales que se hablaron con anterioridad. Es decir los resultados constituyen las pruebas integrales de los dos modos de funcionamiento que MI-T-NATOR posee.

6.2.1. Modo Manual.

En primera instancia, el modo manual empieza colocando a la plataforma robótica sobre la primera cama, ya que esta ubicación es el punto de partida de MI-T-NATOR. En este modo, la superficie sobre la cual se desliza, se ha numerado dado que en ciertas ocasiones la fumigación debe ser para ciertas plantas, con lo cual se ahorra fungicida. Esto se evidencia en las figuras 6.23, 6.24 y 6.25.



Figura 6.23. Numeración Cama 1.



Figura 6.24. Numeración Cama 2.



Figura 6.25. Numeración Cama 3.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Luego de haber colocado a MI-T-NATOR en el sector 1 de la primera cama, se procede a encender a la unidad, correr la interfaz en el computador y seleccionar el modo de operación manual como se muestra en la figura 6.26. En la figura 6.27 se observa el inicio del recorrido luego de haber pulsado la flecha arriba del teclado.



Figura 6.26. Selección del Modo Manual.



Figura 6.27. Acción Adelante con La Flecha Arriba.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Luego de realizar un corto recorrido, presionamos la flecha abajo para que la plataforma se detenga, y presionamos la letra Q para activar el aspersor izquierdo. Esto se evidencia en las figuras 6.28 y 6.29.



Figura 6.28. Acción Parar con La Flecha Abajo.



Figura 6.29. Acción Encender Aspersor Derecho con la Letra Q.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Luego de esto procedemos a cerrar el aspersor con la tecla R que es la tecla de parada para los aspersores. Continuando con el recorrido hacia adelante, MI-T-NATOR debe detenerse en el final de la cama 1 para realizar el primer giro a la derecha. Esto se lo realiza presionando la flecha izquierda como se muestra en la figura 6.30.



Figura 6.30. Acción Giro a la Izquierda con la Flecha Izquierda.

Posteriormente se debe dirigir al robot hacia la cama 2 mediante la flecha arriba, acción que ya se ha explicado antes. En la figura 6.31 se evidencia el movimiento de MI-T-NATOR a través el costado de la cama 1.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Figura 6.31. Desplazamiento Hacia Adelante entre Cama 1 y Cama 2.

Luego se debe parar a MI-T-NATOR con la flecha abajo y girar nuevamente a la izquierda con la flecha izquierda para ingresar a la siguiente cama. Esto se muestra en la figura 6.32.



Figura 6.32. Ingreso a Cama 2.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Luego de ingresar a la cama 2, la plataforma debe dirigirse hacia la mitad de la cama para fumigar ambos lados de la cama, presionando las teclas mencionadas con anterioridad. Para este fin, se la debe ubicar en el punto deseado, y posteriormente accionar ambos aspersores con la letra W del teclado y detenerlos con la letra R. En las figuras 6.33 y 6.34 evidenciamos lo anteriormente mencionado.



Figura 6.33. Ubicación de la Plataforma en la Mitad de la Cama 2.



Figura 6.34. Activación de los Dos Aspersores con la Letra W.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Posteriormente a esto, la plataforma debe desplazarse hasta el fin de la cama 2, parar y realizar el primer giro a la derecha a través de la flecha derecha. Seguir derecho por el costado de la cama 2 y finalmente ingresar con el segundo giro derecho hacia la última cama. Esto se observa en las figuras 6.35, 6.36 y 6.37.



Figura 6.35. Giro a la Derecha con la Flecha Derecha.



Figura 6.36. Desplazamiento Hacia Adelante entre Cama 2 y Cama 3.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Figura 6.37. Ingreso a Cama 3.

Posteriormente, la plataforma debe desplazarse nuevamente hacia adelante hasta un punto específico en el cual realice la fumigación únicamente del lado derecho, para lo cual se activara el aspersor derecho con la letra E, y se lo detendrá con la letra R como se mostró anteriormente. Esto se muestra en la figura 6.38.



Figura 6.38. Activación Aspersor Derecho con la Letra E.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Finalmente, el robot debe dirigirse hacia adelante para terminar su recorrido al final de la cama, con lo que termina el proceso de fumigación en modo manual, ubicando sitios específicos donde existe más concentración de ácaros. Esto se evidencia en la figura 6.39.



Figura 6.39. Fin de Modo Manual.

6.2.2. Modo Automático.

Una vez que las pruebas del modo manual llegaron a su fin de manera satisfactoria, se llevaron a cabo las pruebas del modo automático, que eran las pruebas más demandantes dada la autonomía que debe tener el robot sobre el campo de pruebas.

Después de haber seleccionado el modo automático en la pantalla del HMI, y de dar clic sobre el botón de inicio, MI-T-NATOR empezara a moverse en línea recta activando su aspersor izquierdo. Esto, toda vez que la plataforma se encuentre ubicada en la posición de inicio. Como se aprecia en la figura 6.40.



Figura 6.40. Fumigación Automática Cama 1.

En la figura 6.40, se presentaron las acciones que se detallan a continuación:

Cama 1:

- PWM: 85% ambas ruedas.
- Activación de la válvula izquierda.
- Función alinea hacia la derecha:

PWM rueda derecha: 65%

PWM rueda izquierda: 97%

- Distancia: 15cm.

Función alinea hacia la izquierda:

PWM rueda derecha: 97%

PWM rueda izquierda: 65%

- Distancia: 10 cm.

Luego, la plataforma llega a la parte final de la primera cama, lo que implica que debe empezar a realizar el primer giro a la izquierda. Lo que se observa en la figuras 6.41, 6.42 y 6.43.



Figura 6.41. Final de Cama1.

Fin de cama1 (deja de detecta sensor delantero izquierdo y disminuye la velocidad)

- PWM ambas ruedas: 65%.
- Desactivación de la válvula izquierda.
- No alinea.



Figura 6.42. Empieza el Giro.

El giro a la izquierda empieza el momento en el que el sensor trasero izquierdo deja de detectar las plantas frente a él.



Figura 6.43. Primer Giro a la Izquierda.

Giro Izquierdo 1:

- PWM rueda izquierda (retro):18%.
- PWM rueda derecha: 78%.
- Distancia máxima de giro:20cm.

Una vez que el sensor izquierdo vuelve a detectar la presencia de las plantas, debe realizar la corrección de dirección para desplazarse de manera rectilínea hasta la siguiente cama. En las figuras 6.44, 6.45 y 6.46 se evidencia lo anteriormente mencionado.



Figura 6.44. Detección de las Plantas en el Giro Izquierdo 1.



Figura 6.45. Corrección de Dirección Hacia Giro 2.



Figura 6.46. Corrección de Dirección Hacia Giro 2.

Adelante hacia giro izquierdo 2:

- Función alinea derecha:

PWM rueda izquierda: 65%.

PWM rueda derecha: 0%.

- Distancia: 22 cm.
- Alinea hacia la izquierda:

PWM rueda izquierda: 0%.

PWM rueda derecha: 65%.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

- Distancia: 20cm.

En determinado momento, el sensor trasero izquierdo dejara de detectar la presencia de las plantas, lo que indica que debe realizar el segundo giro hacia la izquierda.

De este modo ingresara a la cama 2 al momento de que el sensor izquierdo delantero detecte la presencia de las plantas. Esto se muestra en las figuras 6.47, 6.48 y 6.49.



Figura 6.47. Inicio del Segundo Giro a la Izquierda.



Figura 6.48. Fin del Segundo Giro 2 a la Izquierda.



Figura 6.49. Ingreso a la Cama 2.

Giro Izquierdo 2:

- PWM rueda izquierda (retro): 18%
- PWM rueda derecha: 78%
- Distancia máxima de giro: 30cm

Una vez que finalizó el giro izquierda 2, ingresa a la segunda cama y activa ambas válvulas al mismo tiempo que realiza la corrección de dirección para alcanzar el fin de la segunda cama. Estas acciones se observan en la figura 6.40.



Figura 6.50. Fumigación Automática Cama 2.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Cama2

- PWM: 85% ambas ruedas.
- Activación de ambas válvulas.
- Alinea hacia la derecha:

PWM rueda derecha: 65%.

PWM rueda izquierda:97%.

- Distancia: 15cm.
- Alinea hacia la izquierda:

PWM rueda derecha: 97%.

PWM rueda izquierda:65%.

- Distancia: 10cm.

Cuando la plataforma llega al final de la cama, se procede a desactivar ambas válvulas y empezar el primer giro a la derecha para desplazarse hacia la última cama de fumigación. Esto se evidencia en las figuras 6.51 y 6.52.



Figura 6.51. Final de Cama 2.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Fin de cama 2 (deja de detecta sensor delantero derecho y disminuye la velocidad).

- PWM ambas ruedas: 65%.
- Desactivación de ambas válvulas.
- No alinea.



Figura 6.52. Inicio del Primer Giro a la Derecha.

Giro derecho:

- PWM rueda derecha (retro): 18%.
- PWM rueda izquierda: 80%.
- Distancia máxima de giro: 20cm.

Luego, el sensor delantero derecho vuelve a detectar la presencia de las plantas, debe realizar la corrección de dirección para desplazarse de manera rectilínea hasta la siguiente cama. En las figuras 6.53, 6.54 y 6.55 se muestran estas acciones.



Figura 6.53. Detección de Plantas en el Giro Derecho 1.



Figura 6.54. Corrección de Dirección Hacia el Giro 2.



Figura 6.55. Corrección de Dirección Hacia el Giro 2.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Adelante hacia giro derecho 2:

- PWM ambas ruedas: 70%.
- Alinea izquierda:

PWM rueda derecha: 70%.

PWM rueda izquierda: 0%.

- Distancia: 19cm.
- Alinea derecha:

PWM rueda derecha: 0%.

PWM rueda izquierda: 70%.

- Distancia: 20cm.

En el momento en el que el sensor trasero derecho deja de presenciar las plantas, inicia el giro hacia la derecha 2 que constituye el último giro, en el cual se activara la válvula derecha. De esta forma ingresa a la cama 3. Esto se observa en las figuras 6.56 y 6.57.



Figura 6.56. Inicio del Giro 2 a la Derecha.

CAPÍTULO 6: PRUEBAS Y RESULTADOS



Figura 6.57. Fin del Giro 2 a la Derecha.

Giro derecha 2:

- PWM rueda izquierda: 80%.
- PWM rueda derecha (retro): 80%.
- Distancia máxima: 30 cm.

Al finalizar el giro hacia la derecha 2, el robot realizara la alineación a lo largo de la cama para así llegar al fin del trayecto que será cuando el sensor trasero deje de detectar presencia, y desactive tanto los motores como la válvula derecha. Esto se evidencia en las figuras 6.45 y 6.59.



Figura 6.58. Fumigación Automática Cama 3.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Cama 3

- PWM: 85% ambas ruedas.
- Función alinea derecha:

PWM rueda derecha: 65%.

PWM rueda izquierda:97%.

- Distancia: 15cm.
- Función alinea izquierda:

PWM rueda derecha: 97%.

PWM rueda izquierda:65%.

- Distancia: 10cm.
-



Figura 6.59. Fin de Modo Automático.

6.2.2.1. Cualificación de resultados.

Para la cualificación de resultados se realizaron 10 pruebas del modo automático dividiendo el campo de pruebas en 5 sectores como se muestra en la figura 5.60:

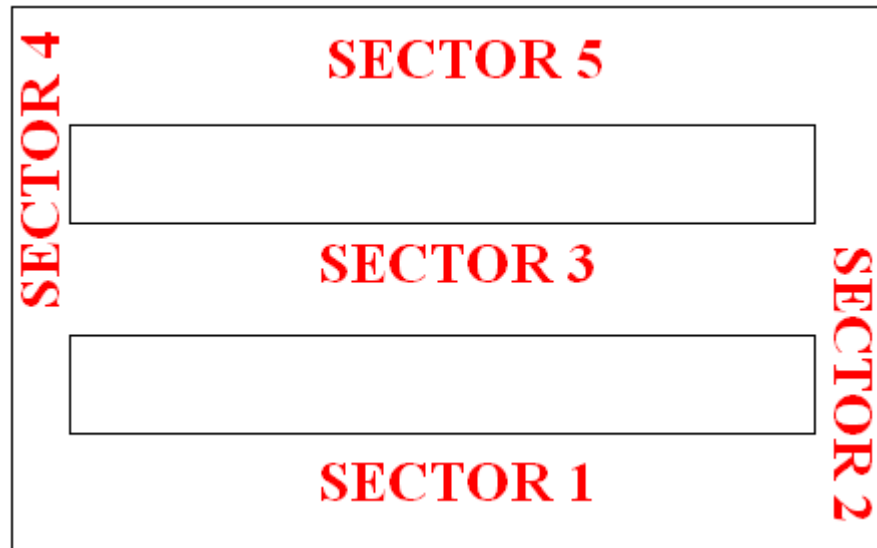


Figura 6.60. Esquema de Trayectoria Dividida en Sectores.

- Para el sector 1 los resultados se muestran en la tabla 6.04.

Tabla 6.04. Resultados Sector 1.

Prueba No	Resultado
1	Sin colisión
2	Sin colisión
3	Sin colisión
4	Sin colisión
5	Sin colisión
6	Sin colisión
7	Sin colisión
8	Sin colisión
9	Sin colisión
10	Sin colisión
Porcentaje de error	0%

- El sector 2 se dividió en 3 partes, lo que se evidencia en la figura 6.61.

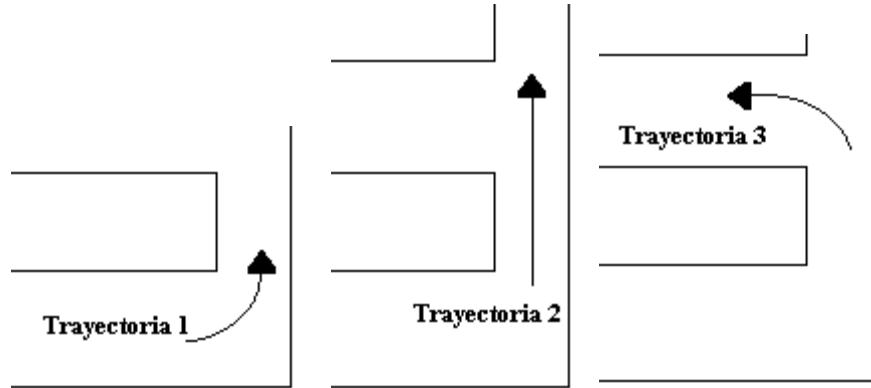


Figura 6.61. Trayectorias del Sector 2.

- Los resultados del sector 2 fueron los que se muestran en la tabla 6.05. En la tabla 6.06 se muestra una comparación de los valores proporcionales de control en los motores.

Tabla 6.05. Resultados Sector 2.

Prueba No	Trayectoria 1	Trayectoria 2	Trayectoria 3
1	Sin colisión	Sin colisión	Sin colisión
2	Sin colisión	Sin colisión	Sin colisión
3	Sin colisión	Sin colisión	Sin colisión
4	Sin colisión	Sin colisión	Sin colisión
5	Sin colisión	Sin colisión	Sin colisión
6	Sin colisión	Sin colisión	Sin colisión
7	Sin colisión	Sin colisión	Sin Colisión
8	Sin colisión	Sin colisión	Sin colisión
9	Sin colisión	Sin colisión	Sin colisión
10	Sin colisión	Sin colisión	Sin colisión
Porcentaje de error	0%	0%	0%

Tabla 6.06. Comparación de Valores PWM Sector 2.

Trayectoria	1	3
PWM calculado rueda izquierda	13	13
%		
PWM implementado rueda izquierda	18	18
%		
Error	5	5

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Rueda izquierda %		
PWM calculado rueda derecha %	78	78
PWM implementado rueda derecha %	78	78
Error rueda derecha %	0	0

- Para el sector 3 se dieron los resultados obtenidos se encuentran tabulados en la tabla 6.07:

Tabla 6.07. Resultados Sector 3.

Prueba No	Resultado
1	Sin colisión
2	Sin colisión
3	Sin colisión
4	Sin colisión
5	Sin colisión
6	Sin colisión
7	Sin colisión
8	Sin colisión
9	Sin colisión
10	Sin colisión
Porcentaje de error	0%

- El sector 4 se dividió en 3 partes como se evidencia en la figura 6.62.

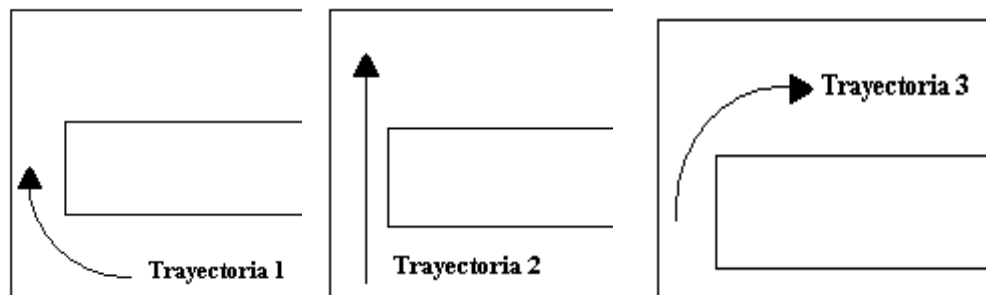


Figura 6.62. Trayectorias del Sector 4.

CAPÍTULO 6: PRUEBAS Y RESULTADOS

- Los resultados del sector 4 fueron los que se muestran en la tabla 6.08. En la tabla 6.09 se muestra una comparación de los valores proporcionales de control en los motores.

Tabla 6.08. Resultados Sector 4.

Prueba No	Trayectoria 1	Trayectoria 2	Trayectoria 3
1	Sin colisión	Sin colisión	Sin colisión
2	Sin colisión	Sin colisión	Sin colisión
3	Sin colisión	Sin colisión	Sin colisión
4	Sin colisión	Colisión	Sin colisión
5	Sin colisión	Sin colisión	Sin colisión
6	Sin colisión	Sin colisión	Sin colisión
7	Sin colisión	Sin colisión	Sin colisión
8	Sin colisión	Sin colisión	Sin colisión
9	Sin colisión	Sin colisión	Sin colisión
10	Sin colisión	Sin colisión	Colisión
Porcentaje de error	0%	10%	10%

Tabla 6.09. Comparación de Valores PWM Sector 4.

Trayectoria	1	3
PWM calculado rueda izquierda %	78	80
PWM implementado rueda izquierda %	80	80
Error Rueda izquierda %	5	0
PWM calculado rueda derecha %	18	80
PWM implementado rueda derecha %	18	80
Error rueda derecha %	0	0

- Para el sector 5 los resultados se encuentran tabulados en la tabla 6.10.

Tabla 6.10. Resultados Sector 5.

Prueba No	Resultado
1	Sin colisión
2	Sin colisión
3	Sin colisión
4	Sin colisión
5	Sin colisión
6	Sin colisión
7	Sin colisión
8	Sin colisión
9	Sin colisión
10	Sin colisión
Porcentaje de error	0%

Finalmente con lo anteriormente expuesto, en la tabla 6.11 se muestra el porcentaje de eficiencia que la plataforma posee en el modo automático.

Tabla 6.11. Cualificación de Pruebas por Sector.

PRUEBA	SECTOR 1	SECTOR 2	SECTOR 3	SECTOR 4	SECTOR 5
1	OK	OK	OK	OK	OK
2	OK	OK	OK	OK	OK
3	OK	OK	OK	OK	OK
4	OK	OK	OK	X	OK
5	OK	OK	OK	OK	OK
6	OK	OK	OK	OK	OK
7	OK	OK	OK	OK	OK
8	OK	OK	OK	OK	OK
9	OK	OK	OK	OK	OK
10	OK	OK	OK	X	OK

CAPÍTULO 6: PRUEBAS Y RESULTADOS

Como se observa en la tabla 6.10, las filas sombreadas son aquellas correspondientes a las pruebas que presentaron fallas en el recorrido. Estas fallas fueron colisiones de la plataforma en los extremos de la trayectoria, debido a que los sensores fallaban en la medición de distancia. Como se puede observar, de 10 pruebas realizadas, 3 de ellas fueron intentos fallidos, con lo cual se concluye que la plataforma tiene un eficacia del 80%, que es un valor adecuado tomando en cuenta que es un prototipo.

CAPÍTULO 7

CONCLUSIONES Y RECOMEDACIONES

7.1. CONCLUSIONES

- El sistema robótico de fumigación después de haber sido sometido a pruebas de campo, se encuentra funcional bajo condiciones ideales, dado que este prototipo se encuentra en su primera fase de desarrollo y se constituye en la base para futuras mejoras.
- Se realizó una investigación técnica – bibliográfica, de donde se extrajo información científica, que fundamenta el presente proyecto.
- Se realizó una investigación de campo que permitió conocer las condiciones bajo las cuales se realiza el proceso de producción florícola. Evidenciando que los químicos que se utilizan para eliminar plagas son altamente tóxicos, y son en gran medida perjudiciales para la salud de los operarios.

CAPÍTULO 7: CONCLUSIONES Y RECOMENDACIONES

- Se diseñó una plataforma móvil utilizando elementos mayormente metálicos para dar soporte a todos los componentes que ella posee. Su configuración es de tres de ruedas en un arreglo diferencial, lo cual permite el giro sobre el propio eje del robot dado el poco espacio en el que se desplaza.
- El diseño de hardware de la plataforma emplea artefactos eléctricos y electrónicos que permiten, a más de la activación de sus componentes, elevar la potencia que genera el controlador, ya que de esto depende el torque para el desplazamiento y la presión de aspersión en la fumigación. Por otro lado la transmisión de datos utiliza módulos serie inalámbricos capaces de comunicarse entre ellos a una distancia de hasta 40 metros, lo cual es idóneo para evitar que el operador este dentro del invernadero en contacto directo con el fungicida.
- Se concluye que la herramienta de desarrollo informático Java, constituye una de las más poderosas plataformas de programación. Ya que en ella se pudo realizar la distribución grafica de controles del robot en la interfaz humano-maquina, y además crear una conexión a una base de datos que realice reportes históricos del proceso de fumigación.
- Pese a que los motores que generan el movimiento de la plataforma poseen características similares, la construcción de ambos es distinta y esto se evidenció en los errores que se presentaron en el desplazamiento del robot. Traduciéndose en un desvío de la trayectoria rectilínea que debe cumplir la plataforma. Lo que quiere decir, que no se puede construir maquinas similares, con características idénticas.
- La distribución de peso en la plataforma es un aspecto crítico, dado que de esto depende la localización del centro de gravedad del robot, y esto afecta directamente a la movilidad del sistema, pero más aún a los giros que realiza.

CAPÍTULO 7: CONCLUSIONES Y RECOMENDACIONES

- Es importante decir que la versatilidad del dispositivo controlador 18f452 permitió satisfacer los requerimientos de memoria del programa, toda vez que un microcontrolador regular no habría superado estas expectativas.
- Los drivers L298 en configuración cascada provocan mayor cantidad de corriente hacia los motores, lo que se traduce en mayor potencia y torque para movilizar la plataforma, logrando mover los aproximadamente 22 Kg que la plataforma posee en carga máxima.
- La técnica de control más adecuada para el manejo de motores de corriente continua de baja potencia, es el control proporcional, dado que no se requiere controlar el arranque de los mismos, más si su velocidad de funcionamiento. Y esto se lo realiza mediante el uso de ancho de pulsos configurados digitalmente en el controlador.

7.2. RECOMENDACIONES

- Los elementos de potencia como son los transistores TIP31C y los driver L298, por la potencia a la que son sometidos, generan una gran cantidad de calor que puede afectar su correcto funcionamiento. Por ello es necesario colocar disipadores metálicos para así reducir los fallos provocados por el recalentamiento de estos dispositivos electrónicos.
- Dado que el método de comunicación y transferencia de energía en el sistema se lo realiza mediante cables conductores, se recomienda la colocación de guías para prevenir contactos con el líquido fungicida y de esta manera también mejor la estética del sistema.

CAPÍTULO 7: CONCLUSIONES Y RECOMENDACIONES

- Debido a la reducida visualización de la cámara implementada en el modo manual, se recomienda montar otra cámara en la plataforma, para una mejor visualización del terreno y manipulación de la plataforma.
- El sistema mecánico constituye la columna vertebral de un robot, cualquiera que sea la configuración, en este caso una plataforma móvil. Por ello es indispensable que en futuros proyectos se realice un análisis exhaustivo del sistema mecánico a fin de minimizar los errores que se presentaron en el presente proyecto.
- Ya que MI-T-NATOR tiene un peso importante dado sus materiales de construcción, se recomienda implementar un arreglo de poleas entre los motores y las llantas a fin de no forzar los ejes, que es un problema constante del diseño actual.
- Se recomienda implementar un giroscopio digital para los giros de la plataforma ya que daría un control más fino al momento de desplazarse entre las distintas camas.
- Para lograr realizar un control de velocidad más robusto, es recomendable implementar encoders en las ruedas, de manera que se pueda realizar un mapeo preciso de la superficie sobre la cual el robot debe desplazarse.
- En un futuro, se puede optar por un control más robusto de la aspersión, por lo cual se recomienda la implementación de un sensor de caudal, para que se pueda realizar un control en lazo cerrado y disminuir aún más el desperdicio de fungicida.

REFERENCIAS BIBLIOGRÁFICAS

- Info Blogspot. (2009). Robótica Agrícola. Recuperado de:
<http://jc-info.blogspot.com/2009/12/robotica-agricola.html>
- Explored. (2009). Robótica quiere automatizar al Ecuador. Recuperado de:
<http://www.explored.com.ec/noticias-ecuador/robotica-quiere-automatizar-al-ecuador-354572.html>
- Ruiz del Solar, Salazar. (2011). Robots Móviles. Recuperado de:
| <http://robotica.li2.uchile.cl/EL63G/capitulo4.pdf>
- Hernández. (2012). Motores de Corriente Continua. Recuperado de :
<http://electronicaadilson.blogspot.com/2012/05/tipos-de-motores.html>
- Blacio (2009). Bombas y Motores. Recuperado de:
<http://www.dspace.espol.edu.ec/bitstream/123456789/6349/1/TNP%20Cap4%20Motores%20y%20Bombas.pdf>
- Carvallo, Vargas (2013). Válvulas de Solenoide. Recuperado de :
<http://profesores.elo.utfsm.cl/~jgb/CARVALLOVARGASc.pdf>
- Ingeniería MCI. (2013). Módulos de Transmisión Inalámbrica. Recuperado de :
<http://www.xbee.cl/>
- Archundia. (2004). El Estadar IEEE 802.15.4. Recuperado de :
http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capitulo4.pdf
- MrGroup. (2009). Sensores de Distancia. Recuperado de:
<http://138.100.100.129/mrgroup/doku.php?id=teaching:sensors>
- (Simbaña, O.V. (2006). Medidor de Líquidos Controlado por un Microcontrolador. Tesis de tecnología no publicada EPN, Quito, Ecuador).

ANEXO A

***CODIGO FUENTE DEL CONTROLADOR
EN LENGUAJE C***

ANEXO A

```

#include <18f452.h>
#define adc=10
#define FUSES NOWDT
#define FUSES HS
  #define FUSES NOPROTECT
#define FUSES NOOSCSN
#define FUSES NOBROWNOUT
#define FUSES NOPUT
#define FUSES NOSTVREN
#define FUSES NODEBUG
#define FUSES NOLVP
#define FUSES NOWRT
#define FUSES NOWRTD
#define FUSES NOWRTB
#define FUSES NOWRTC
#define FUSES NOCPD
#define FUSES NOCPB
#define FUSES NOEBTR
#define FUSES NOEBTRB

#define use delay(clock=20000000)
#define use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,stream=PC1)
char direccion,dato;
int niva=0, nivm=0, nivb=0, nivc=0, gir=0, ad=0, at=0, izq=0, der=0, pwd=0, pwi=0, contd=0,
manual=0,autom=0;
int pwm_c=0,cont=0,ales=0,pwm_s=0,i=0,j=0,ales_c=0,ales_d=0,pwm_f=0;
float calc=0x00,val=0x00,calc1=0x00,calc2=0x00,calc3=0x00,distancia=0x00;
float dist[25],distr[25],x=0,y=0;
unsigned int32 tempo=0;

//interrupcion puerto serial
#define int_RDA
void RDA_isr(void)
{
  disable_interrupts(GLOBAL);
  if(kbhit()){
    direccion=getc();
    dato=direccion;
    putc(direccion);
  }
  enable_interrupts(GLOBAL);
}
//funcion para verificar el nivel de liquido

```

ANEXO A

```

void nivel(){
    /// nivel critico////////////////////
    if(!(input(PIN_E0))&&(!input(PIN_E1))&&(!input(PIN_E2))){

        nivc=1;
        nivb=0;
        nivm=0;
        niva=0;
    }

    ///// nivel bajo////////////////////
    if((input(PIN_E0))&&(!input(PIN_E1))&&(!input(PIN_E2))){

        printf("3");
        nivc=0;
        nivb=1;
        nivm=0;
        niva=0;

    }

    ///// error sensor nivel bajo////////////////////
    if(!(input(PIN_E0))&&(input(PIN_E1))&&(!input(PIN_E2))){
        printf("errb");
    }

    ///// nivel medio //////////////////////
    if((input(PIN_E0))&&(input(PIN_E1))&&(!input(PIN_E2))){
        printf("2");
        nivc=0;
        nivb=0;
        nivm=1;
        niva=0;

    }

    /// error sensor nivel medio bajo ///
    if((!input(PIN_E0))&&(!input(PIN_E1))&&(input(PIN_E2))){
        printf("errmb");

    }

    ///// error sensor nivel medio////////
    if((input(PIN_E0))&&(!input(PIN_E1))&&(input(PIN_E2))){
        printf("errm");
    }

    ///// error sensor nivel bajo///
    if(!(input(PIN_E0))&&(input(PIN_E1))&&(input(PIN_E2))){
        printf("errb");
    }

```

ANEXO A

```

}
///// nivel alto //
if((input(PIN_E0))&&(input(PIN_E1))&&(input(PIN_E2))){
printf("1");
  nivc=0;
  nivb=0;
  nivm=0;
  niva=1;
  }
}

//filtro datos de sensores
void seleccion(){
for(i=1;i<20;i++){
x=abs(dist[i]-dist[i-1]);
if(x<=3){
distr[j]=dist[i];
j++;
}
}
for(i=0;i<j;i++){
y=y+distr[i];
}
y=(y/j);
}
void toma_adi(){
dato=0x00;
///// sensor infrarojo adelante izquierda
set_adc_channel(0);
for(i=0;i<25;i++){
dist[i]=0;
distr[i]=0;
}
y=0;
x=0;
i=0;
j=0;
do{
dist[i]=read_adc();
i++;
delay_us(100);
}while(i<20);
seleccion();
calc=y;
for(i=0;i<25;i++){
dist[i]=0;
distr[i]=0;
}
}

```

ANEXO A

```

    }
    y=0;
    x=0;
    i=0;
    j=0;

}
void toma_add(){
dato=0x00;
///// sensor infrarojo adelante izquierda
set_adc_channel(2);
for(i=0;i<25;i++){
dist[i]=0;
distr[i]=0;
}
y=0;
x=0;
i=0;
j=0;
do{
dist[i]=read_adc();
i++;
delay_us(1600);
}while(i<20);
seleccion();
calc2=y;
for(i=0;i<25;i++){
dist[i]=0;
distr[i]=0;
}
y=0;
x=0;
i=0;
j=0;
}
//Conversor analogo digital
void toma_adc(){
dato=0x00;
calc=0;
calc1=0;
calc2=0;
calc3=0;
///// sensor infrarojo adelante izquierda
delay_us(50);
set_adc_channel(0);
delay_us(50);

```


ANEXO A

```

    calc=read_adc();
    //// sensor infrarojo atras izquierda
    delay_us(50);
    set_adc_channel(1);
    delay_us(50);
    calc1=read_adc();
    //// sensor infrarojo adelante derecha
    delay_us(100);
    delay_us(50);
    set_adc_channel(2);
    delay_us(50);
    calc2=read_adc();
    //// sensor infrarojo atras derecha
    delay_us(50);
    set_adc_channel(3);
    delay_us(50);
    calc3=read_adc();
    delay_us(50);
    }
    ////pwm motor izquierdo ////
    void setpwm1(){
        nivel();
        //////////////////////////////////////
        ////BAJO CRITICO
        if((nivb==1||nivc==1)&&(pwi==1)&&(pwm_c!=1)){
            //nivel bajo o critico alinea
            set_pwm1_duty(320);
        }
        if((nivb==1||nivc==1)&&(pwd==1)&&(pwm_c!=1)){
            //nivel bajo o critico alinea
            set_pwm1_duty(280);
        }
        if((nivb==1||nivc==1)&&(pwm_c==1)){
            //modo automatico nivel bajo o critico disminuye vel al llegar al borde
            set_pwm1_duty(260);
        }
        if((nivb==1||nivc==1)&&(pwd!=1)&&(pwi!=1)&&(pwm_c!=1)){
            // velocidad normal nivel bajo critico
            set_pwm1_duty(300);
            gir=0;
        }
        ////ALTO MEDIO
        if((niva==1||nivm==1)&&(pwi==1)&&(pwm_c!=1)){
            // nivel medio alto alinea
            set_pwm1_duty(388);
        }
    }

```

ANEXO A

```

    if((niva==1||nivm==1)&&(pwd==1)&&(pwm_c!=1)){
        // nivel medio alto alinea
        set_pwm1_duty(260);
    }
    if((nivm==1||niva==1)&&(pwm_s==1)&&(pwi!=1)&&(pwd!=1)){
        //modo automatico nivel medio o alto disminuye vel al llegar al borde
        set_pwm1_duty(312);
    }
if((nivm==1||niva==1)&&(pwd!=1)&&(pwi!=1)&&(pwm_c!=1)&&(pwm_s!=1)){
    /// velocidad normal nivel medio alto
    set_pwm1_duty(340);
    gir=1;

}
if((nivm==1||niva==1)&&(pwm_f==1)&&(pwi!=1)&&(pwd!=1)){
    set_pwm1_duty(276);

}

}
}
/////pwm motor derecho
void setpwm2(){
    nivel();
    ///////////NIVEL BAJO CRITICO///
    //////////////////////////////////////
    ///////////nivel bajo critico
    if((nivb==1||nivc==1)&&(pwi==1)&&(pwm_c!=1)){
        set_pwm2_duty(280);
    }
    if((nivb==1||nivc==1)&&(pwd==1)&&(pwm_c!=1)){
        set_pwm2_duty(320);
    }
    if((nivb==1||nivc==1)&&pwm_c==1){
        //nivel bajo o critico motor 2 giro auto filo
        set_pwm2_duty(260);
    }

if((nivb==1||nivc==1)&&(pwd!=1)&&(pwi!=1)&&(pwm_c!=1)&&(pwm_s!=1)){
    set_pwm2_duty(300);
    gir=0;

}

}
/////NIVEL MEDIO ALTO
//nivel medio alto
if((niva==1||nivm==1)&&(pwi==1)&&(pwm_c!=1)){
    set_pwm2_duty(260);
}
}

```

ANEXO A

```

if((niva==1||nivm==1)&&(pwd==1)&&(pwm_c!=1)){
    set_pwm2_duty(388);//355
}
if((nivm==1||niva==1)&&(pwm_c==1)){
//modo automatico nivel medio o alto disminuye vel al llegar al borde
    set_pwm2_duty(280);
}
if((nivm==1||niva==1)&&(pwm_s==1)&&(pwi!=1)&&(pwd!=1)){
//modo automatico nivel medio o alto disminuye vel al llegar al borde
    set_pwm2_duty(312);//312
    // set_pwm2_duty(300);
}
if((nivm==1||niva==1)&&(pwd!=1)&&(pwi!=1)&&(pwm_c!=1)&&(pwm_s!=1)){

    set_pwm2_duty(340);
    gir=1;
// printf("seteadopw2");
}
if((nivm==1||niva==1)&&(pwm_f==1)&&(pwi!=1)&&(pwd!=1)){
    set_pwm2_duty(276);
}
}
//derecha correcion trayetoria recta
void derecha_pwmc(){
    nivel();
    if(gir==0){
        setpwm1();
        output_low(PIN_B3);
        output_high(PIN_B2);
        setpwm2();
        output_high(PIN_B7);
        output_low(PIN_B6);
    }
    if(gir==1){
        setpwm1();
        output_low(PIN_B3);
        output_high(PIN_B2);

        setpwm2();

        output_high(PIN_B7);
        output_low(PIN_B6);
    }
}
//correccion derecha trayetoria recta

```

ANEXO A

```

void izquierda_pwmc(){
    nivel();
    if(gir==0){
        setpwm2();
        output_high(PIN_B7);
        output_low(PIN_B6);
        setpwm1();
        output_low(PIN_B3);
        output_high(PIN_B2);
    }

    if(gir==1){
        setpwm2();
        output_high(PIN_B7);
        output_low(PIN_B6);
        setpwm1();
        output_low(PIN_B3);
        output_high(PIN_B2);
    }
}

void derecha_pwm(){
    nivel();
    if(gir==0){
        setpwm1();
        output_low(PIN_B3);
        output_high(PIN_B2);
        setpwm2();
        output_low(PIN_B7);
        output_low(PIN_B6);
    }
    if(gir==1){
        setpwm1();
        output_low(PIN_B3);
        output_high(PIN_B2);
        setpwm2();
        output_low(PIN_B7);
        output_high(PIN_B6);
    }
}

void izquierda_pwm(){
    nivel();
    if(gir==0){
        setpwm1();
        output_high(PIN_B2);
        output_low(PIN_B3);
    }
}

```

ANEXO A

```
    output_low(PIN_B6);
    output_low(PIN_B7);
    }
if(gir==1){
    setpwm2();
    output_low(PIN_B6);
    output_high(PIN_B7);
    // set_pwm1_duty(380);
    setpwm1();
    output_high(PIN_B3);
    output_low(PIN_B2);
    // delay_ms(5);
    }
}
void retro(){
    ad=0;
    at=1;
    izq=0;
    der=0;
    nivel();
    setpwm1();
    output_high(PIN_B3);
    output_low(PIN_B2);
    setpwm2();
    output_high(PIN_B6);
    output_low(PIN_B7);
    }
void derecha(){
    ad=0;
    at=0;
    izq=0;
    der=1;
    nivel();
if(gir==0){
    setpwm1();
    output_low(PIN_B3);
    output_high(PIN_B2);
    output_low(PIN_B6);
    output_low(PIN_B7);
    }
if(gir==1){
    set_pwm2_duty(328);
    output_low(PIN_B7);
    output_high(PIN_B6);

    set_pwm1_duty(324);
```

ANEXO A

```

        output_high(PIN_B2);
        output_low(PIN_B3);
    }
}

void adelante(){
    ad=1;
    at=0;
    izq=0;
    der=0;
    nivel();
    setpwm2();
    output_high(PIN_B7);
    output_low(PIN_B6);
    setpwm1();
    output_high(PIN_B2);
    output_low(PIN_B3);
}

void adelante_fin(){
    ad=1;
    at=0;
    izq=0;
    der=0;
    nivel();

    setpwm2();
    set_pwm1_duty(320);
    output_high(PIN_B7);
    output_low(PIN_B6);
    set_pwm1_duty(324);
    output_high(PIN_B2);
    output_low(PIN_B3);
}

void izquierda(){
    ad=0;
    at=0;
    izq=1;
    der=0;
    nivel();
    if(gir==0){
        setpwm2();
        output_high(PIN_B7);
        output_low(PIN_B6);
        setpwm1();
        output_low(PIN_B3);
    }
}

```

ANEXO A

```

        output_low(PIN_B2);
    }
    if(gir==1){
//  set_pwm2_duty(360);
    setpwm2();
    output_low(PIN_B6);
    output_high(PIN_B7);
//  set_pwm1_duty(380);
    setpwm1();
    output_high(PIN_B3);
    output_low(PIN_B2);
//  delay_ms(5);
    }
    }
    void izquierda_eje(){
    ad=0;
    at=0;
    izq=1;
    der=0;
    setpwm2();
    output_high(PIN_B7);
    output_low(PIN_B6);
    set_pwm1_duty(72);
    output_high(PIN_B3);
    output_low(PIN_B2);
    }
    void derecha_eje(){
    ad=0;
    at=0;
    izq=0;
    der=1;
    set_pwm1_duty(328);
    output_low(PIN_B3);
    output_high(PIN_B2);
    set_pwm2_duty(72);
    output_high(PIN_B6);
    output_low(PIN_B7);
    }
    void parar(){
    ad=0;
    at=0;
    izq=0;
    der=0;
    output_high(PIN_B2);
    output_high(PIN_B3);

```

```

    output_high(PIN_B6);
    output_high(PIN_B7);
    // set_pwm2_duty(0);
    // set_pwm1_duty(0);
    }
//Funcion Alinea modo manual
void alinea_manual(){
    toma_adc();
    if((ad==1)&&(calc>200)){//225
        pwi=1;

        do{
            if(dato=='i'){
                parar();
                ad=0;
                at=0;
                izq=0;
                der=0;
            }
            toma_adc();
            if(contd==0){
                derecha_pwm();
            }
            contd++;
        }while((calc>195)&&(calc<200));
        pwi=0;
        contd=0;
        adelante();
    }

    if((ad==1)&&(calc2>200)){
        pwd=1;
        do{
            if(dato=='i') {
                parar();

            }
            toma_adc();
            if(contd==0){
                izquierda_pwm();
            }
            contd++;
        }while((calc2>195)&&(calc<200));
        pwd=0;
        contd=0;
        adelante();
    }

```


ANEXO A

```

if((izq==1)&&(calc>192)){
    parar();
    izq=0;
}
if((der==1)&&(calc2>192)){
    parar();
    der=0;
}
}
}
//Funcion alineaa moo automatico
void alineaa(){
    toma_adc();
    if((ales==0)&&(ales_c==0)&&(ales_d==0)){
        if((ad==1)&&(calc>135)){//225
            pwi=1;

            do{
                if(dato=='i'){
                    parar();
                    ad=0;
                    at=0;
                    izq=0;
                    der=0;
                }
                toma_adc();
                if(contd==0){
                    derecha_pwm();
                }
                contd++;
            }while((calc>130)&&(calc2<260));

            pwi=0;
            contd=0;
            adelante();
        }

        if((ad==1)&&(calc2>230)){
            pwd=1;
            do{
                if(dato=='i') {
                    parar();
                }
                toma_adc();

                if(contd==0){

```

```

        izquierda_pwm();
    }
    contd++;
}while((calc2>229)&&(calc<140));
pwd=0;
contd=0;
adelante();
if((izq==1)&&(calc>192)){
    parar();
    izq=0;
}
if((der==1)&&(calc2>192)){
    parar();
    der=0;
}
}
}
if(ales==1){
// nivel();
pwm_c=0;
if((ad==1)&&(calc>190)){
// pwi=1;

do{
    if(dato=='i'){
        parar();
        ad=0;
        at=0;
        izq=0;
        der=0;
    }

    toma_adc();
    if(contd==0){
        derecha_pwm();
    }
    contd++;

}while((calc>140));//178
contd=0;
pwm_c=1;
adelante();
}
if((ad==1)&&(calc2>220)){
    pwm_c=1;
    do{

```

ANEXO A

```

    if(dato=='i') {
        parar();
    }
    toma_adc();

    if(contd==0){
        izquierda_pwm();
    }
    contd++;
    }while((calc2>217));//215
contd=0;
pwm_c=1;
adelante();
    }
}

if(ales_c==1){
if((calc>135)){
    pwi=1;
    do{
        if(dato=='i'){
            parar();
            ad=0;
            at=0;
            izq=0;
            der=0;
        }
        toma_adc();
        if(contd==0){
            derecha_pwm();
        }
        contd++;

    }while((calc>130)&&(calc2<260));
    pwi=0;
    contd=0;
    adelante_fin();
    }
if((calc2>235)){
    pwd=1;
    do{
        if(dato=='i') {
            parar();
        }
        toma_adc();
        if(contd==0){
            izquierda_pwm();

```

```

        }
        contd++;
    }while((calc2>234)&&(calc<135));
pwd=0;
contd=0;
adelante_fin();
    }
}

if(ales_d==1){
    pwm_s=1;
    pwm_f=0;
    if((calc2>225)){

do{
    if(dato=='i'){
        parar();
        ad=0;
        at=0;
        izq=0;
        der=0;

    }
    toma_adc();
    if(contd==0){
        izquierda_pwm();
    }
    contd++;
}while((calc2>224));
contd=0;
pwm_s=0;
pwm_f=1;
adelante();
    }

if((calc>192)){

    pwm_s=1;
    pwm_f=0;
    do{
        if(dato=='i') {
            parar();
        }
        toma_adc();
    if(contd==0){
        derecha_pwm();

```

ANEXO A

```

        }
        contd++;
    }while((calc>191));
contd=0;
pwm_s=0;
pwm_f=1;
adelante();
    }
}

}
//Activacion aspersor izquierdo
void asp_i(){
    output_high(PIN_B0);
    output_high(PIN_B1);
    output_low(PIN_B4);
}
//Activacion aspersor derecho
void asp_d(){
    output_high(PIN_B0);
    output_high(PIN_B4);
    output_low(PIN_B1);
}
// Activacion ambos aspersores
void asp_a(){
    output_high(PIN_B0);
    output_high(PIN_B4);
    output_high(PIN_B1);
}
//Desactivacion aspersores
void asp_s(){
    output_low(PIN_B0);
    output_low(PIN_B4);
    output_low(PIN_B1);
}
//Funcion principal
void main()
{
    setup_adc_ports(AN0_AN1_AN2_AN3_AN4);
    setup_adc(ADC_CLOCK_DIV_2);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SS_DISABLED);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DIV_BY_1,99,1);
    setup_wdt(WDT_OFF);

```

ANEXO A

```
setup_ccp1(CCP_PWM);
setup_ccp2(CCP_PWM);
enable_interrupts(INT_RDA);
enable_interrupts(GLOBAL);
set_tris_b(0x00);
output_b(0x00);
delay_ms(500);
nivel();
printf("MITENATOR!!!!!!!!");
while(true){
  //activacion modo manual
  if((dato=='m')){
    manual=1;
    autom=0;
  }
  if (dato=='n'){
    nivel();
    dato=0x00;
  }

  if(manual==1){
    if(dato=='u')
    {
      pwm_s=1;
      adelante();
    }

    if(dato=='k')
    {
      derecha_eje();
    }
    if(dato=='h')
    {
      izquierda_eje();
    }
    if(dato=='j')
    {
      retro();
    }
    if(dato=='i')
    {
      parar();
    }
    alineamiento_manual();
    if(dato=='r')
    {
```

ANEXO A

```

        asp_d();
    }
    if(dato=='l')
    {
        asp_i();
    }
    if(dato=='b')
    {
        asp_a();
    }
    if(dato=='a')
    {
        asp_s();
    }
    autom=0;
}

//Modo automatico
if(dato=='x'){
    manual=0;
    autom=1;
}
if(autom==1){
    manual=0;
    if(dato=='t'){
        nivel();
        //adelante
        asp_i();
        delay_ms(300);
        adelante();

        cont=0;
    do{
        if(dato=='i'){
            parar();
            delay_ms(5000);
        }
        alinea();
        toma_adc();
    }while((calc>3));
    asp_s();
    //Fin de cama 1
    //adelante lento
    if((calc2>220)){
        izquierda();
    }
}

```

ANEXO A

```

do{
toma_adc();
}while(calc2>215);
}
parar();
delay_ms(200);
adelante();
pwm_c=1;
cont=0;
do{
if(dato=='i'){
parar();

}
toma_adc();
cont++;
parar();
adelante();
}while((calc1>11));
cont=0;
pwm_s=1;
delay_ms(400);
parar();
cont=0;
delay_ms(200);
//Izquierda
izquierda_eje();
delay_ms(100);
do{
if(dato=='i'){
parar();
}
toma_adi();
delay_ms(10);
}while(calc<=193);
parar();
cont=0;
delay_ms(200);
pwm_s=0;
pwm_c=1;
//Adelante borde
adelante();
do{
if(dato=='i'){
parar();
}
}

```


ANEXO A

```

    ales=1;
    alinea();
    toma_adc();
    }while((calc>10));
    do{
        if(dato=='i'){
            parar();
        }
        ales=1;
        alinea();
        toma_adc();
    }while(calc1>14);
    ales=0;
    parar();
    delay_ms(200);
//Izquierda
    pwm_c=1;
    izquierda_eje();
    delay_ms(850);

do{
    if(dato=='i'){
        parar();
    }
    toma_adc();
    }while(calc<=111);
    delay_ms(50);
    do{
        if(dato=='i'){
            parar();
        }
        toma_adc();
    }while(calc<=111);
//Adelante
    derecha();
do{
    toma_adc();
    }while(calc>210);
    delay_ms(100);
    pwm_c=0;
    ales=0;
    pwi=0;
    pwd=0;
    cont=0;
    parar();

```

ANEXO A

```

delay_ms(300);

pwm_f=1;
adelante();
ad=1;
tempo=0;
do{
  if(dato=='i'){
    parar();
  }
  if((calc1>100)||((calc3>100))){
    asp_a();
  }
  alinea();
  toma_adc();
  tempo++;
}while(tempo<80);
tempo=0;
do{
  if(dato=='i'){
    parar();
  }
  alinea();
  toma_adc();
}while(((calc>3)&&(calc2>3)));
asp_s();
parar();
delay_ms(200);

pwm_f=0;
//fin de cama 2
parar();
delay_ms(700);//250
pwm_c=1;
//adelante
adelante();
cont=0;
do{
  if(dato=='i'){
    parar();
  }
  alinea();
  toma_adc();
  parar();
  adelante();
  cont++;
}

```

```
    }while((calc1>50)&&calc3>50);
cont=0;
pwm_c=0;
delay_ms(220);
parar();
cont=0;
delay_ms(250);
//derecha
pwm_s=1;
derecha_eje();
delay_ms(850);
do{
if(dato=='i'){
    parar();
    }
toma_adc();
printf(" calc2 es %f ",calc2);
    }while(calc2<=193);
    delay_ms(100);
do{
if(dato=='i'){
    parar();
    }
toma_adc();
printf(" calc2 es %f ",calc2);
    }while(calc2<=193);
    parar();
    cont=0;
    delay_ms(200);
    pwm_s=0;
    pwm_c=0;
    pwm_f=1;
//Adelante borde
adelante();
tempo=0;
do{
    if(dato=='i'){
    parar();
    }
    ales_d=1;
    alineaa();
    toma_adc();
    tempo++;
    }while((calc2>10)||tempo<20);
    delay_ms(50);
    tempo=0;
```

ANEXO A

```

    do{
    if(dato=='i'){
    parar();
    }
    ales_d=1;
    alineaa();
    toma_adc();
    }while((calc2>10));
    parar();
    delay_ms(750);
    adelante();
    do{
    if(dato=='i'){
    parar();
    }
    ales_d=1;
    alineaa();
    toma_adc();

    }while(calc3>20);
    ales_d=0;
//Derecha
pwm_f=0;
pwm_c=0;
pwm_s=1;
parar();
cont=0;
delay_ms(700);
derecha();
delay_ms(400);
do{
    if(dato=='i'){
    parar();
    }
    toma_adc();
    printf("calc2 %f,"calc2);
    }while(calc2<=125);
    delay_ms(10);
    do{
    if(dato=='i'){
    parar();
    }
    toma_adc();

    printf("calc2 %f,"calc2);

```

```
    }while(calc2<=126);//110
//adelante
pwm_c=0;
pwi=0;
pwd=0;
cont=0;
parar();
tempo=0;
delay_ms(400);
pwm_s=0;
  ales_d=0;
adelante_fin();
  ad=1;
pwm_s=1;
pwm_c=0;
  ales_d=0;
ales_c=1;
do{
  if(dato=='i'){
    parar();
  }
  if(calc1>100){
    asp_d();
  }

  alineo();
  toma_adc();
  tempo++;
}while(tempo<35);
adelante();
ales_c=0;
tempo=0;
pwm_s=0;
do{
  if(dato=='i'){
    parar();
  }
  if(calc1>100){
    asp_d();
  }
  alineo();
  toma_adc();
  tempo++;
}while(calc2>50||tempo<30);
parar();
```

ANEXO A

```
    asp_s();
    delay_ms(300);
    tempo=0;
    parar();
    }
    }
    enable_interrupts(GLOBAL);
clear_interrupt(int_rda);
    }
    }
```

ANEXO B

***CODIGO FUENTE LA INTERFAZ
JAVA***

ANEXO B

```
package hardworker01;

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author adria_000
 */

import giovynet.serial.Baud;
import giovynet.serial.Com;
import giovynet.serial.Parameters;
import java.awt.KeyEventDispatcher;
import java.awt.KeyboardFocusManager;
import java.awt.event.KeyEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Calendar;
import javax.swing.JOptionPane;

public class Principal extends javax.swing.JFrame {

    private class Teclado implements KeyEventDispatcher {

        @Override
        public boolean dispatchKeyEvent(KeyEvent e) {

            if (e.getKeyCode() == KeyEvent.VK_UP) {
                System.out.println("adelante");

                try {

                    // TODO add your handling code here:

                    Parameters settings = new Parameters();
                    settings.setPort("COM5");

                    settings.setBaudRate(Baud._9600);

                    settings.setMinDelayWrite(10);

                    Com com1 = new Com(settings);

                    com1.sendSingleData('u');

                    String ul=com1.receiveSingleString();

                    recibe(ul);

                    com1.close();

                } catch (Exception ex) {
```



```
        ex.printStackTrace();
    }

} else if (e.getKeyCode() == KeyEvent.VK_S) {
    System.out.println("paro");

    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();

        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('i');

        String ul=com1.receiveSingleString();

        recibe(ul);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }

} else if (e.getKeyCode() == KeyEvent.VK_DOWN) {
    System.out.println("atras");

    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();

        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('j');

        String ul=com1.receiveSingleString();

        recibe(ul);

        com1.close();

    } catch (Exception ex) {
```

```
        ex.printStackTrace();
    }

} else if (e.getKeyCode() == KeyEvent.VK_RIGHT) {
    System.out.println("derecha");

    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();
        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('k');

        String ul=com1.receiveSingleString();

        recibe(ul);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }

} else if (e.getKeyCode() == KeyEvent.VK_LEFT) {
    System.out.println("izquierda");

    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();

        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('h');

        String ul=com1.receiveSingleString();

        recibe(ul);

        com1.close();

    } catch (Exception ex) {
```

```
        ex.printStackTrace();
    }
} else if (e.getKeyCode() == KeyEvent.VK_W) {
    System.out.println("ambos");

    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();

        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('b');

        String ul=com1.receiveSingleString();

        recibe(ul);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }

} else if (e.getKeyCode() == KeyEvent.VK_Q) {
    System.out.println("aspizq");

    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();

        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('l');

        String ul=com1.receiveSingleString();

        recibe(ul);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
```

```
    }

} else if (e.getKeyCode() == KeyEvent.VK_E) {
    System.out.println("aspder");

    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();

        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('r');

        String ul=com1.receiveSingleString();

        recibe(ul);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }

} else if (e.getKeyCode() == KeyEvent.VK_R) {
    System.out.println("paroasp");

    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();

        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('a');

        String ul=com1.receiveSingleString();

        recibe(ul);

        com1.close();

    } catch (Exception ex) {
```

ANEXO B

```

        ex.printStackTrace();
    }

    }

    return false;
}

/**
 * Creates new form Principal
 */
public Principal() {

    initComponents();

    pman.setVisible(false);
    comenzar.setVisible(false);

    //nival.setVisible(false);
    nivme.setVisible(false);
    nivba.setVisible(false);

    //na.setVisible(false);
    nm.setVisible(false);
    nb.setVisible(false);

}

/**
 * This method is called from within the constructor to
 initialize the form.
 * WARNING: Do NOT modify this code. The content of this method
 is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jButton9 = new javax.swing.JButton();
    buttonGroup1 = new javax.swing.ButtonGroup();
    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    auto = new javax.swing.JRadioButton();
    manual = new javax.swing.JRadioButton();
    selmo = new javax.swing.JButton();
    es = new javax.swing.JLabel();
    comenzar = new javax.swing.JButton();
    jPanel2 = new javax.swing.JPanel();
    logo = new javax.swing.JLabel();

```

ANEXO B

```

jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jPanel3 = new javax.swing.JPanel();
jLabel5 = new javax.swing.JLabel();
pman = new javax.swing.JPanel();
Izquierda = new javax.swing.JButton();
Parar = new javax.swing.JButton();
Derecha = new javax.swing.JButton();
adelante = new javax.swing.JButton();
Atras = new javax.swing.JButton();
aspiz = new javax.swing.JButton();
aspder = new javax.swing.JButton();
aspam = new javax.swing.JButton();
partot = new javax.swing.JButton();
jPanel5 = new javax.swing.JPanel();
sif = new javax.swing.JLabel();
pem = new javax.swing.JButton();
jLabel10 = new javax.swing.JLabel();
nival = new javax.swing.JLabel();
nivme = new javax.swing.JLabel();
nivba = new javax.swing.JLabel();
na = new javax.swing.JLabel();
nm = new javax.swing.JLabel();
nb = new javax.swing.JLabel();
jPanel4 = new javax.swing.JPanel();
jLabel6 = new javax.swing.JLabel();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
jMenuItem2 = new javax.swing.JMenuItem();

jButton9.setText("Ambos");
jButton9.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jButton9ActionPerformed(evt);
    }
});

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
;

setTitle("HMI Fumigador");

jPanell1.setBorder(javax.swing.BorderFactory.createTitledBorder(java
x.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBor
der.RAISED), "Selección de Modo",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, null, new
java.awt.Color(0, 51, 204)));

jLabel11.setText("Modo:");

buttonGroup1.add(auto);
auto.setText("AUTOMATICO");

buttonGroup1.add(manual);
manual.setText("MANUAL");

```

ANEXO B

```

        selmo.setText("SELECCIONAR");
        selmo.addActionListener(new java.awt.event.ActionListener()
        {
            public void actionPerformed(java.awt.event.ActionEvent
            evt) {
                selmoActionPerformed(evt);
            }
        });

        es.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

        comenzar.setText("COMENZAR");
        comenzar.addActionListener(new
        java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
            evt) {
                comenzarActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel1Layout = new
        javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
        .LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            jPanel1Layout.createSequentialGroup()

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
            .Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing
                    .GroupLayout.Alignment.LEADING)
                        .addGroup(jPanel1Layout.createSequentialGroup()
                            .addGroup(jPanel1Layout.createParallelGroup(javax
                            .swing.GroupLayout.Alignment.LEADING)
                                .addGroup(jPanel1Layout.createSequentialGroup()
                                    .addGroup(jPanel1Layout.createSequentialGroup()
                                    .addGap(44, 44, 44)
                                    .addComponent(auto)
                                    .addGap(18, 18, 18)
                                    .addComponent(manual))
                                .addGroup(jPanel1Layout.createSequentialGroup()
                                    .addGroup(jPanel1Layout.createSequentialGroup()
                                    .addComponent(jLabel1)
                                    .addGap(18, 18, 18)
                                    .addComponent(es,
                                    javax.swing.GroupLayout.PREFERRED_SIZE, 178,
                                    javax.swing.GroupLayout.PREFERRED_SIZE)))
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
                            , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
                            .Alignment.TRAILING, false)
                                .addComponent(comenzar,
                                javax.swing.GroupLayout.DEFAULT_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                .addComponent(selmo,
                                javax.swing.GroupLayout.DEFAULT_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                            .addGap(53, 53, 53)
                        );
            jPanel1Layout.setVerticalGroup(

```

ANEXO B

```

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup())

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
    .addComponent(auto)
    .addComponent(manual)
    .addComponent(selmo))
    .addGap(15, 15, 15)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
    .addComponent(jLabel1)
    .addComponent(es,
javax.swing.GroupLayout.PREFERRED_SIZE, 46,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(comenzar))
    .addGap(32, 32, 32)
);

jPanel2.setBorder(javax.swing.BorderFactory.createBevelBorder(javax
.swing.border.BevelBorder.RAISED));

    logo.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/hardworker01/espe.jp
g"))); // NOI18N

    jLabel2.setFont(new java.awt.Font("Tahoma", 1, 24)); //
NOI18N
    jLabel2.setForeground(new java.awt.Color(0, 153, 0));
    jLabel2.setText("DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA");

    jLabel3.setFont(new java.awt.Font("Tahoma", 1, 18)); //
NOI18N
    jLabel3.setForeground(new java.awt.Color(51, 51, 51));
    jLabel3.setText("PROTOTIPO DE ROBOT FUMIGADOR PARA CAMPOS
DE ROSAS");

    jLabel4.setFont(new java.awt.Font("Tahoma", 0, 14)); //
NOI18N
    jLabel4.setForeground(new java.awt.Color(51, 51, 51));
    jLabel4.setText("IVÁN LEÓN / ADRIÁN VÁSQUEZ");

    javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(logo)

```


ANEXO B

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(28, 28, 28)
        .addComponent(jLabel3)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(177, 177, 177)
        .addComponent(jLabel4)
    .addGroup(jPanel2Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)
    .addComponent(jLabel2)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    );
jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(jLabel3)
        .addGap(18, 18, 18)
        .addComponent(jLabel4)
        .addGap(0, 0, Short.MAX_VALUE))
        .addComponent(logo,
javax.swing.GroupLayout.DEFAULT_SIZE, 114, Short.MAX_VALUE))
    );

jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(java
x.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBor
der.RAISED), "Esquema de Trayectoria"));

    jLabel5.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/hardworker01/campopl
at.jpg"))); // NOI18N

    javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addGap(26, 26, 26)
        .addComponent(jLabel5,
javax.swing.GroupLayout.PREFERRED_SIZE, 404,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

ANEXO B

```

        .addContainerGap(38, Short.MAX_VALUE))
    );
    jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup())
            .addGap(70, 70, 70)
            .addComponent(jLabel5)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

pman.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.s
wing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder
.RAISED), "Controles Modo Manual",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, null, new
java.awt.Color(0, 0, 204)));

    Izquierda.setText("Izquierda");
    Izquierda.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            IzquierdaActionPerformed(evt);
        }
    });

    Parar.setText("Parar");
    Parar.addActionListener(new java.awt.event.ActionListener()
{
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            PararActionPerformed(evt);
        }
    });

    Derecha.setText("Derecha");
    Derecha.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            DerechaActionPerformed(evt);
        }
    });

    adelante.setText("Adelante");
    adelante.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            adelanteActionPerformed(evt);
        }
    });

    Atras.setText("Atras");
    Atras.addActionListener(new java.awt.event.ActionListener()
{

```

ANEXO B

```

        public void actionPerformed(java.awt.event.ActionEvent
    evt) {
            AtrasActionPerformed(evt);
        }
    });

    aspiz.setText("Aspersor Izq");
    aspiz.addActionListener(new java.awt.event.ActionListener()
    {
        public void actionPerformed(java.awt.event.ActionEvent
    evt) {
            aspizActionPerformed(evt);
        }
    });

    aspder.setText("Aspersor Der");
    aspder.addActionListener(new
    java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
    evt) {
            aspderActionPerformed(evt);
        }
    });

    aspam.setText("Ambos");
    aspam.addActionListener(new java.awt.event.ActionListener()
    {
        public void actionPerformed(java.awt.event.ActionEvent
    evt) {
            aspamActionPerformed(evt);
        }
    });

    partot.setText("Parada Total");
    partot.addActionListener(new
    java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
    evt) {
            partotActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout pmanLayout = new
    javax.swing.GroupLayout(pman);
    pman.setLayout(pmanLayout);
    pmanLayout.setHorizontalGroup(

    pmanLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
    ADING)
        .addGroup(pmanLayout.createSequentialGroup()
            .addGroup(pmanLayout.createParallelGroup()
                .addComponent(Izquierda)
                .addGap(18, 18, 18)
            )
        )
        .addGroup(pmanLayout.createParallelGroup(javax.swing.GroupLayout.Al
    ignment.LEADING)
            .addGroup(pmanLayout.createSequentialGroup()
                .addGroup(pmanLayout.createParallelGroup()
                    .addComponent(Derecha)
                    .addGap(18, 18, 18)
                )
            )
        )
        .addGroup(pmanLayout.createParallelGroup(javax.swing.GroupLayout.Al
    ignment.LEADING, false)
            .addGroup(pmanLayout.createSequentialGroup()
                .addComponent(AspersorIzq)
                .addGap(18, 18, 18)
                .addComponent(AspersorDer)
                .addGap(18, 18, 18)
            )
        )
    );

```

ANEXO B

```

        .addComponent (Atras,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent (Parar,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap (18, 18, 18)
        .addComponent (Derecha)
        .addComponent (adelante,
javax.swing.GroupLayout.PREFERRED_SIZE, 79,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap (114, 114, 114)

.addGroup (pmanLayout.createParallelGroup (javax.swing.GroupLayout.Al
ignment.LEADING)
        .addComponent (aspder,
javax.swing.GroupLayout.DEFAULT_SIZE, 113, Short.MAX_VALUE)
        .addComponent (aspam,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent (aspiz,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGap (18, 18, 18)
        .addComponent (partot,
javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap (27, 27, 27))
);
pmanLayout.setVerticalGroup (

pmanLayout.createParallelGroup (javax.swing.GroupLayout.Alignm
ENT.LEADING)
        .addGroup (pmanLayout.createSequentialGroup ()
        .addContainerGap ()

.addGroup (pmanLayout.createParallelGroup (javax.swing.GroupLayout.Al
ignment.BASELINE)
        .addComponent (adelante)
        .addComponent (aspiz))

.addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)

.addGroup (pmanLayout.createParallelGroup (javax.swing.GroupLayout.Al
ignment.BASELINE)
        .addComponent (Parar)
        .addComponent (Derecha)
        .addComponent (Izquierda)
        .addComponent (aspam)
        .addComponent (partot))

.addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)

.addGroup (pmanLayout.createParallelGroup (javax.swing.GroupLayout.Al
ignment.BASELINE)
        .addComponent (Atras)
        .addComponent (aspder))

```

ANEXO B

```

.addContainerGap (javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    );

jPanel5.setBorder (javax.swing.BorderFactory.createTitledBorder (java
x.swing.BorderFactory.createBevelBorder (javax.swing.border.BevelBor
der.RAISED), "Estado Robot"));

    pem.setBackground (new java.awt.Color (255, 0, 0));
    pem.setFont (new java.awt.Font ("Tahoma", 1, 12)); // NOI18N
    pem.setText ("PARADA DE EMERGENCIA");
    pem.addActionListener (new java.awt.event.ActionListener () {
        evt) {
            public void actionPerformed (java.awt.event.ActionEvent
                pemActionPerformed (evt);
            }
        });

    jLabel10.setText ("Nivel de agua Inicial:");

    nival.setBackground (new java.awt.Color (255, 0, 0));
    nival.setIcon (new
javax.swing.ImageIcon (getClass ().getResource ("/hardworker01/verde.p
ng"))); // NOI18N

    nivme.setIcon (new
javax.swing.ImageIcon (getClass ().getResource ("/hardworker01/amarill
o.png"))); // NOI18N

    nivba.setIcon (new
javax.swing.ImageIcon (getClass ().getResource ("/hardworker01/rojo.pn
g"))); // NOI18N

    na.setFont (new java.awt.Font ("Tahoma", 0, 20)); // NOI18N
    na.setText ("Nivel Alto");

    nm.setFont (new java.awt.Font ("Tahoma", 0, 20)); // NOI18N
    nm.setText ("Nivel Medio");

    nb.setFont (new java.awt.Font ("Tahoma", 0, 20)); // NOI18N
    nb.setText ("Nivel Bajo");

    javax.swing.GroupLayout jPanel5Layout = new
javax.swing.GroupLayout (jPanel5);
    jPanel5.setLayout (jPanel5Layout);
    jPanel5Layout.setHorizontalGroup (

jPanel5Layout.createParallelGroup (javax.swing.GroupLayout.Alignment
.LEADING)
    .addGroup (jPanel5Layout.createSequentialGroup ()

.addContainerGap (javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addGroup (jPanel5Layout.createParallelGroup (javax.swing.GroupLayout
.Alignment.LEADING)
    .addGroup (jPanel5Layout.createSequentialGroup ()
        .addComponent (jLabel10)

```

ANEXO B

```

.addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
        .addComponent (sif,
javax.swing.GroupLayout.PREFERRED_SIZE, 157,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent (pem,
javax.swing.GroupLayout.PREFERRED_SIZE, 198,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup (jPanel5Layout.createSequentialGroup ())

.addGroup (jPanel5Layout.createParallelGroup (javax.swing.GroupLayout
.Alignment.TRAILING, false)
        .addComponent (nivba,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent (nivme,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent (nival,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGap (51, 51, 51)

.addGroup (jPanel5Layout.createParallelGroup (javax.swing.GroupLayout
.Alignment.LEADING)
        .addComponent (nm)
        .addComponent (nb)
        .addComponent (na))))
);
jPanel5Layout.setVerticalGroup (

jPanel5Layout.createParallelGroup (javax.swing.GroupLayout.Alignment
.LEADING)
        .addGroup (jPanel5Layout.createSequentialGroup ())

.addGroup (jPanel5Layout.createParallelGroup (javax.swing.GroupLayout
.Alignment.TRAILING)
        .addGroup (jPanel5Layout.createSequentialGroup ())
        .addComponent (pem,
javax.swing.GroupLayout.PREFERRED_SIZE, 97,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED
)

.addGroup (jPanel5Layout.createParallelGroup (javax.swing.GroupLayout
.Alignment.TRAILING)
        .addComponent (sif,
javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (jLabel10))
        .addGap (18, 18, 18)
        .addComponent (nival,
javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent (na)
        .addGap (18, 18, 18)

```

ANEXO B

```

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
    .addComponent(nivme,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(nm,
javax.swing.GroupLayout.Alignment.TRAILING))
.addGap(18, 18, 18)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.TRAILING)
    .addComponent(nivba,
javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(nb))
.addContainerGap(59, Short.MAX_VALUE)
);

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(java
x.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBor
der.RAISED), "MI-T-NATOR"));

    jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/hardworker01/robo.pn
g"))); // NOI18N

    javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);
    jPanel4.setLayout(jPanel4Layout);
    jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel4Layout.createSequentialGroup())

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    .addComponent(jLabel6)
    .addGap(39, 39, 39)
);
jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup())
    .addContainerGap()
    .addComponent(jLabel6,
javax.swing.GroupLayout.PREFERRED_SIZE, 341, Short.MAX_VALUE)
    .addContainerGap())
);

jMenu1.setText("Archivo");

 jMenuItem1.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.a
wt.event.KeyEvent.VK_R, java.awt.event.InputEvent.ALT_MASK));
jMenuItem1.setText("Reportes");

```


ANEXO B

```

        .addComponent(jPanel4,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(jPanel5,
javafx.swing.GroupLayout.DEFAULT_SIZE, 0, Short.MAX_VALUE)
        .addComponent(pman,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
        .addContainerGap()
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javafx.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javafx.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addGap(14, 14, 14)
            .addComponent(jPanel2,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED)
)

.addGroup(layout.createParallelGroup(javafx.swing.GroupLayout.Alignm
ent.TRAILING, false)
        .addComponent(jPanel4,
javafx.swing.GroupLayout.Alignment.LEADING,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jPanel5,
javafx.swing.GroupLayout.Alignment.LEADING,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jPanel3,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)

.addGroup(layout.createParallelGroup(javafx.swing.GroupLayout.Alignm
ent.LEADING, false)
        .addComponent(pman,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jPanell1,
javafx.swing.GroupLayout.DEFAULT_SIZE, 0, Short.MAX_VALUE))

.addContainerGap(javafx.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

    pack();
} // </editor-fold>

private void adelanteActionPerformed(java.awt.event.ActionEvent
evt) {

```

ANEXO B

```

// TODO add your handling code here:

try {
    // TODO add your handling code here:

    Parameters settings = new Parameters();
    settings.setPort("COM5");

    settings.setBaudRate(Baud._9600);

    settings.setMinDelayWrite(10);

    Com com1 = new Com(settings);

    com1.sendSingleData('u');

    String u1=com1.receiveSingleString();

    recibe(u1);

    com1.close();

} catch (Exception ex) {
    ex.printStackTrace();
}

}

private void
IzquierdaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();
        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('h');

        String u1=com1.receiveSingleString();

        recibe(u1);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }

}

```

ANEXO B

```
private void AtrasActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();
        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('j');

        String u1=com1.receiveSingleString();

        recibe(u1);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }

}

private void DerechaActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();
        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('k');

        String u1=com1.receiveSingleString();

        recibe(u1);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }

}
```

ANEXO B

```
    }

    private void PararActionPerformed(java.awt.event.ActionEvent
    evt) {
        // TODO add your handling code here:
        try {
            // TODO add your handling code here:

            Parameters settings = new Parameters();
            settings.setPort("COM5");

            settings.setBaudRate(Baud._9600);

            settings.setMinDelayWrite(10);

            Com com1 = new Com(settings);

            com1.sendSingleData('i');

            String u1=com1.receiveSingleString();

            recibe(u1);

            com1.close();

        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    private void aspizActionPerformed(java.awt.event.ActionEvent
    evt) {
        // TODO add your handling code here:
        try {
            // TODO add your handling code here:

            Parameters settings = new Parameters();
            settings.setPort("COM5");

            settings.setBaudRate(Baud._9600);

            settings.setMinDelayWrite(10);

            Com com1 = new Com(settings);

            com1.sendSingleData('l');

            String u1=com1.receiveSingleString();

            recibe(u1);

            com1.close();

        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

ANEXO B

```

private void aspderActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();
        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('r');

        String u1=com1.receiveSingleString();

        recibe(u1);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

private void aspamActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:

    try {
        // TODO add your handling code here:

        Parameters settings = new Parameters();
        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('b');

        String u1=com1.receiveSingleString();

        recibe(u1);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

private void jButton9ActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:

```

ANEXO B

```

    }

    private void partotActionPerformed(java.awt.event.ActionEvent
    evt) {
        // TODO add your handling code here:
        try {
            // TODO add your handling code here:
            Parameters settings = new Parameters();
            settings.setPort("COM5");

            settings.setBaudRate(Baud._9600);

            settings.setMinDelayWrite(10);

            Com com1 = new Com(settings);

            com1.sendSingleData('a');

            String u1=com1.receiveSingleString();

            recibe(u1);

            com1.close();

        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    private void selmoActionPerformed(java.awt.event.ActionEvent
    evt) {
        // TODO add your handling code here:

        if (auto.isSelected()) {
            manual.setVisible(false);

            es.setText("AUTO");

            String p1 = "AUTO";

            conectar(p1);

            comenzar.setVisible(true);

            try {

                Parameters settings = new Parameters();
                settings.setPort("COM5");

                settings.setBaudRate(Baud._9600);

                settings.setMinDelayWrite(10);

                Com com1 = new Com(settings);

                com1.sendSingleData('x');

                System.out.println("Datos Enviados "+settings);

                String u1=com1.receiveSingleString();

```

ANEXO B

```

        recibe (u1);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }

}

if (manual.isSelected()) {
    auto.setVisible(false);

    pman.setVisible(true);

    es.setText("MANUAL");

    String p2 = "MANUAL";

    conectar(p2);

    //sentencias de teclado
    KeyboardFocusManager manager =
KeyboardFocusManager.getCurrentKeyboardFocusManager();
    manager.addKeyEventDispatcher(new Teclado());

    try {

        Parameters settings = new Parameters();
        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('m');

        String u1=com1.receiveSingleString();

        recibe (u1);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }

}

private void pemActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:

```

ANEXO B

```

    try {

        Parameters settings = new Parameters();
        settings.setPort("COM5");

        settings.setBaudRate(Baud._9600);

        settings.setMinDelayWrite(10);

        Com com1 = new Com(settings);

        com1.sendSingleData('e');

        String u1=com1.receiveSingleString();

        recibe(u1);

        com1.close();

    } catch (Exception ex) {
        ex.printStackTrace();
    }

}

private void
 jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int confirmado = JOptionPane.showConfirmDialog(this,
    "¿Desea Salir?");

    if (JOptionPane.OK_OPTION == confirmado) {
        System.exit(0);

    } else {
        try {
            // TODO add your handling code here:
            Parameters settings = new Parameters();
            settings.setPort("COM5");

            settings.setBaudRate(Baud._9600);

            settings.setMinDelayWrite(10);

            Com com1 = new Com(settings);

            com1.sendSingleData('e');

            com1.close();

        } catch (Exception ex) {
            ex.printStackTrace();
        }

        System.out.println("Terminado");
    }

}

private void
 jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

```


ANEXO B

```

        String seleccion = JOptionPane.showInputDialog(this,
        "Ingrese Contraseña:", JOptionPane.QUESTION_MESSAGE);
        if (seleccion.equals("adrian")) {

            System.out.println("Correcto");
            datos i = new datos();
            i.setVisible(true);
        } else {
            JOptionPane.showMessageDialog(this, "Clave Erronea");
        }

    }

    private void comenzarActionPerformed(java.awt.event.ActionEvent
    evt) {
        // TODO add your handling code here:
        try {
            // TODO add your handling code here:
            Parameters settings = new Parameters();
            settings.setPort("COM5");

            settings.setBaudRate(Baud._9600);

            settings.setMinDelayWrite(10);

            Com com1 = new Com(settings);

            com1.sendSingleData('t');

            String u1=com1.receiveSingleString();

            recibe(u1);

            com1.close();

        } catch (Exception ex) {
            ex.printStackTrace();
        }

    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and
        feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available,
        stay with the default look and feel.
         * For details see
        http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf
        .html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

```

ANEXO B

```

javax.swing.UIManager.setLookAndFeel(info.getClassName());
        break;
    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Principal.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Principal.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Principal.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Principal.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Principal().setVisible(true);
    }
});
}
// Variables declaration - do not modify
private javax.swing.JButton Atras;
private javax.swing.JButton Derecha;
private javax.swing.JButton Izquierda;
private javax.swing.JButton Parar;
private javax.swing.JButton adelante;
private javax.swing.JButton aspam;
private javax.swing.JButton aspder;
private javax.swing.JButton aspiz;
private javax.swing.JRadioButton auto;
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JButton comenzar;
private javax.swing.JLabel es;
private javax.swing.JButton jButton9;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;

```

ANEXO B

```

private javax.swing.JLabel logo;
private javax.swing.JRadioButton manual;
private javax.swing.JLabel na;
private javax.swing.JLabel nb;
private javax.swing.JLabel nival;
private javax.swing.JLabel nivba;
private javax.swing.JLabel nivme;
private javax.swing.JLabel nm;
private javax.swing.JButton partot;
private javax.swing.JButton pem;
private javax.swing.JPanel pman;
private javax.swing.JButton selmo;
private javax.swing.JLabel sif;
// End of variables declaration

public void conectar(String p) {
    String bd = "fumigador";
    String login = "root";
    String password = "Isaac2004";
    String url = "jdbc:mysql://localhost/" + bd;
    Connection conn = null;
    try {

        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(url, login,
password);
        DriverManager.setLogStream(System.out);

        if (conn != null) {
            System.out.println("Conexión a base de datos " +
url + " .. Ok");

            Statement stmt = conn.createStatement();
            //String hola = ("hola");
            Calendar Cal = Calendar.getInstance();

            String fec = Cal.get(Calendar.DATE) + "/" +
Cal.get(Calendar.MONTH) + "/" + Cal.get(Calendar.YEAR);

            /*int
fec=Cal.get(Calendar.DAY_OF_MONTH+Calendar.MONTH+Calendar.YEAR);

            String fecha=Integer.toString(fec);*/

            String hora = Cal.get(Calendar.HOUR_OF_DAY) + ":" +
Cal.get(Calendar.MINUTE) + ":" + Cal.get(Calendar.SECOND);

            String l = Integer.toString(2);

            String sentencia = "insert into datos values ('" +
fec + "',''" + hora + "',''" + p + "',''" + l + "')";
            System.out.println(sentencia);
            stmt.executeUpdate(sentencia);

            conn.close();
        }
    } catch (SQLException ex) {

```

ANEXO B

```
        System.out.println("Hubo un problema al intentar
conectarse con la base de datos " + url);
        System.out.println(ex.getCause());
    } catch (ClassNotFoundException ex) {
        System.out.println(ex);
    }
}

public void recibe(String a){

    System.out.println("El dato es: "+a);

    if(a.equals("1"))
    {
        nival.setVisible(true);
        na.setVisible(true);

        nivme.setVisible(false);
        nm.setVisible(false);
        nivba.setVisible(false);
        nb.setVisible(false);
    }
    if(a.equals("2"))
    {
        nivme.setVisible(true);
        nm.setVisible(true);

        nival.setVisible(false);
        na.setVisible(false);
        nivba.setVisible(false);
        nb.setVisible(false);
    }
    if(a.equals("3"))
    {
        nivba.setVisible(true);
        nb.setVisible(true);

        nivme.setVisible(false);
        nm.setVisible(false);
        nival.setVisible(false);
        na.setVisible(false);
    }
}

}
```

ANEXO C

HOJAS TÉCNICAS DE LOS DISPOSITIVOS UTILIZADOS



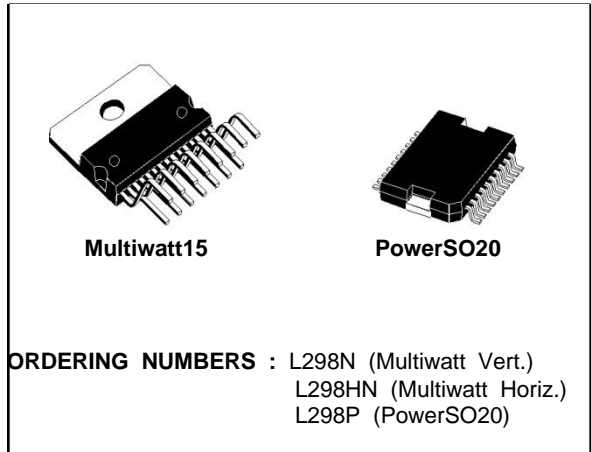
L298

DUAL FULL-BRIDGE DRIVER

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

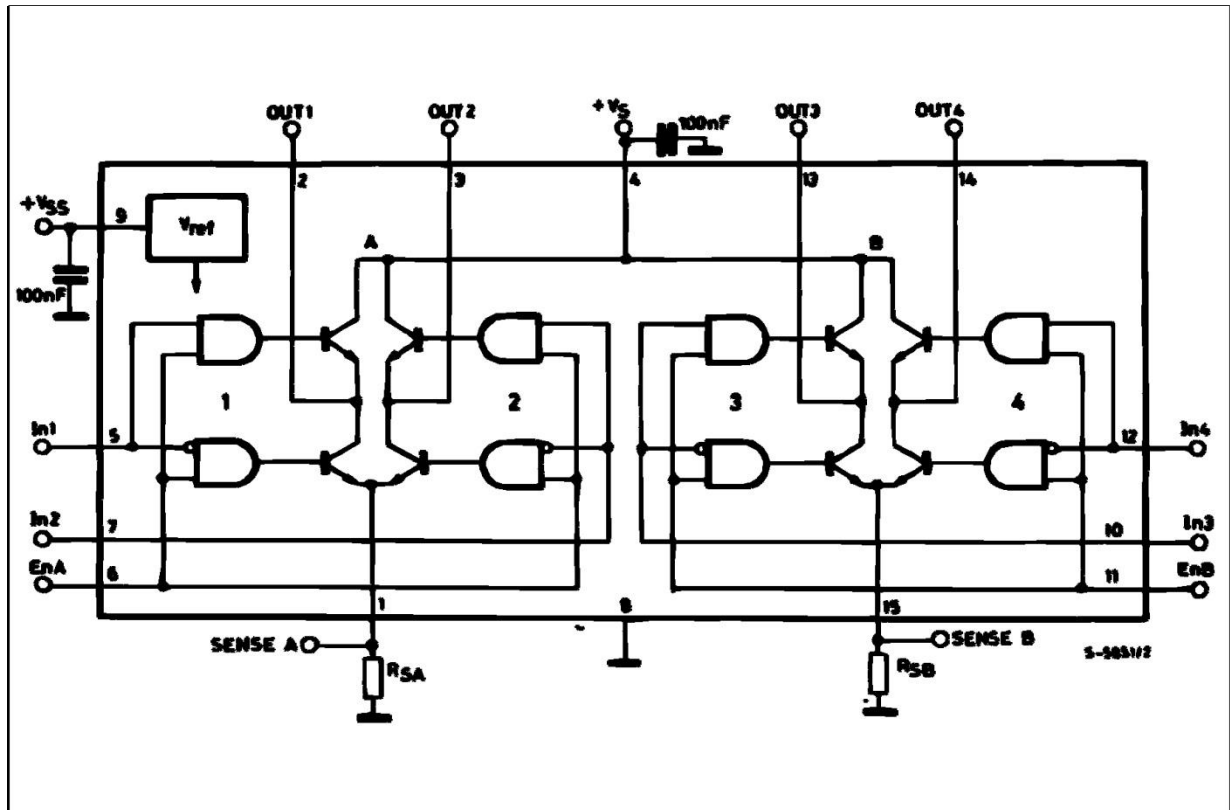
DESCRIPTION

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 de-signed 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 in-put signals. The emitters of the lower transistors of each bridge are connected together and the corre-sponding 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



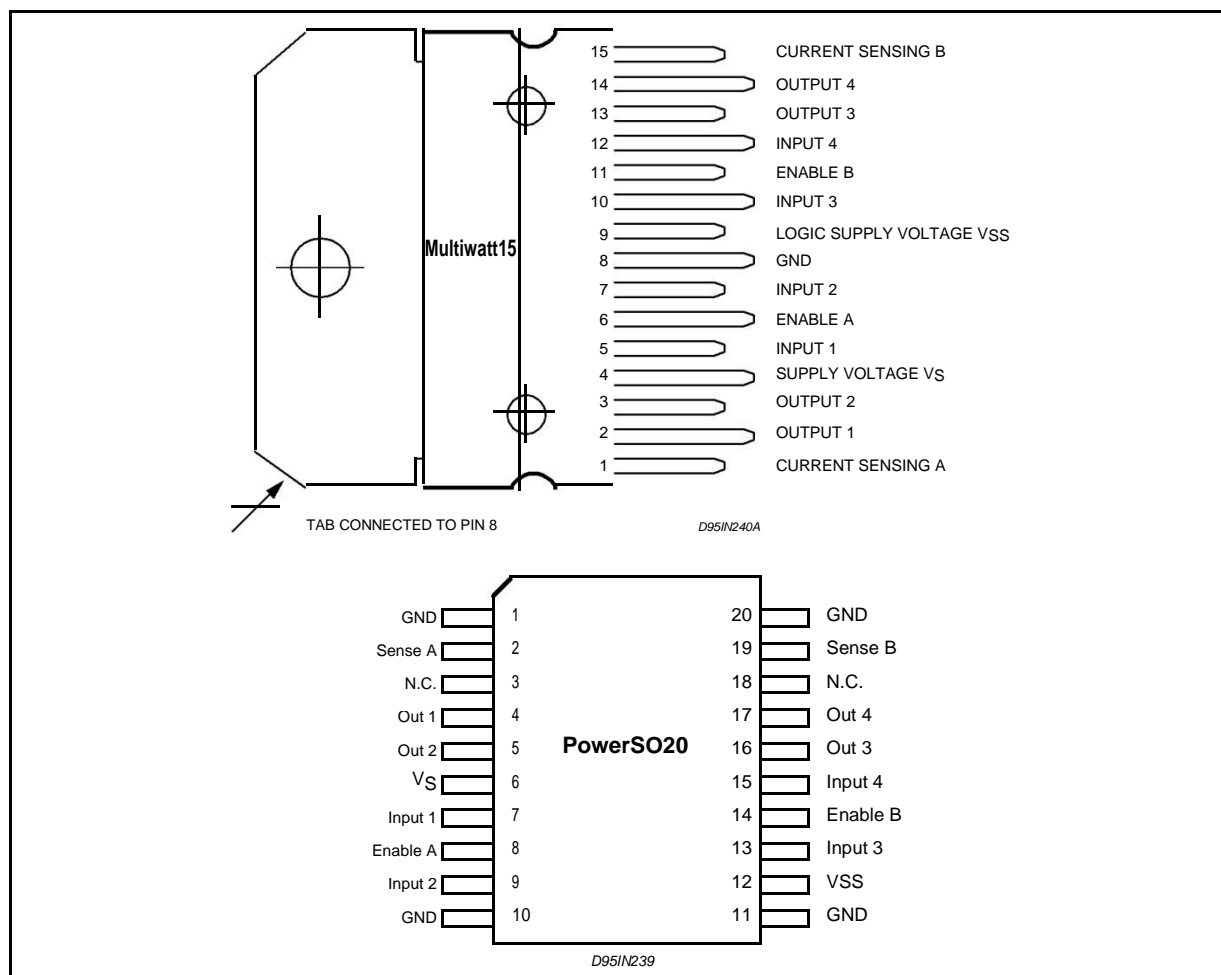
ANEXO C

L298

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V _S	Power Supply	50	V
V _{SS}	Logic Supply Voltage	7	V
V _I , V _{en}	Input and Enable Voltage	-0.3 to 7	V
I _O	Peak Output Current (each Channel)		
	- Non Repetitive (t = 100μs)	3	A
	- Repetitive (80% on -20% off; t _{on} = 10ms)	2.5	A
	-DC Operation	2	A
V _{sens}	Sensing Voltage	-1 to 2.3	V
P _{tot}	Total Power Dissipation (T _{case} = 75°C)	25	W
T _{op}	Junction Operating Temperature	-25 to 130	°C
T _{stg} , T _j	Storage and Junction Temperature	-40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
R _{th j-case}	Thermal Resistance Junction-case	Max. -	3	°C/W
R _{th j-amb}	Thermal Resistance Junction-ambient	Max. 13 (*)	35	°C/W

(*) Mounted on aluminum substrate



PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	Vs	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS ($V_s = 42V$; $V_{SS} = 5V$, $T_j = 25^\circ C$; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_s	Supply Voltage (pin 4)	Operative Condition	$V_{IH} + 2.5$		46	V
V_{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	$V_{en} = H$; $I_L = 0$ $V_i = L$ $V_i = H$		13 50	22 70	mA mA
		$V_{en} = L$ $V_i = X$			4	mA
I_{SS}	Quiescent Current from VSS (pin 9)	$V_{en} = H$; $I_L = 0$ $V_i = L$ $V_i = H$		24 7	36 12	mA mA
		$V_{en} = L$ $V_i = X$			6	mA
V_{iL}	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V_{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V_{SS}	V
I_{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	$V_i = L$			-10	μA
I_{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	$V_i = H \leq V_{SS} - 0.6V$		30	100	μA
$V_{en} = L$	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
$V_{en} = H$	Enable High Voltage (pins 6, 11)		2.3		V_{SS}	V
$I_{en} = L$	Low Voltage Enable Current (pins 6, 11)	$V_{en} = L$			-10	μA
$I_{en} = H$	High Voltage Enable Current (pins 6, 11)	$V_{en} = H \leq V_{SS} - 0.6V$		30	100	μA
$V_{CEsat} (H)$	Source Saturation Voltage	$I_L = 1A$ $I_L = 2A$	0.95	1.35 2	1.7 2.7	V V
$V_{CEsat} (L)$	Sink Saturation Voltage	$I_L = 1A$ (5) $I_L = 2A$ (5)	0.85	1.2 1.7	1.6 2.3	V V
V_{CEsat}	Total Drop	$I_L = 1A$ (5) $I_L = 2A$ (5)	1.80		3.2 4.9	V V
V_{sens}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

ANEXO C

L298

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$T_1 (V_i)$	Source Current Turn-off Delay	$0.5 V_i$ to $0.9 I_L$ (2); (4)		1.5		μs
$T_2 (V_i)$	Source Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (2); (4)		0.2		μs
$T_3 (V_i)$	Source Current Turn-on Delay	$0.5 V_i$ to $0.1 I_L$ (2); (4)		2		μs
$T_4 (V_i)$	Source Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (2); (4)		0.7		μs
$T_5 (V_i)$	Sink Current Turn-off Delay	$0.5 V_i$ to $0.9 I_L$ (3); (4)		0.7		μs
$T_6 (V_i)$	Sink Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (3); (4)		0.25		μs
$T_7 (V_i)$	Sink Current Turn-on Delay	$0.5 V_i$ to $0.9 I_L$ (3); (4)		1.6		μs
$T_8 (V_i)$	Sink Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (3); (4)		0.2		μs
$f_c (V_i)$	Commutation Frequency	$I_L = 2\text{A}$		25	40	KHz
$T_1 (V_{en})$	Source Current Turn-off Delay	$0.5 V_{en}$ to $0.9 I_L$ (2); (4)		3		μs
$T_2 (V_{en})$	Source Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (2); (4)		1		μs
$T_3 (V_{en})$	Source Current Turn-on Delay	$0.5 V_{en}$ to $0.1 I_L$ (2); (4)		0.3		μs
$T_4 (V_{en})$	Source Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (2); (4)		0.4		μs
$T_5 (V_{en})$	Sink Current Turn-off Delay	$0.5 V_{en}$ to $0.9 I_L$ (3); (4)		2.2		μs
$T_6 (V_{en})$	Sink Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (3); (4)		0.35		μs
$T_7 (V_{en})$	Sink Current Turn-on Delay	$0.5 V_{en}$ to $0.9 I_L$ (3); (4)		0.25		μs
$T_8 (V_{en})$	Sink Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (3); (4)		0.1		μs

1) Sensing voltage can be -1 V for $t \leq 50\ \mu\text{sec}$; in steady state $V_{\text{sens min}} \geq -0.5\text{ V}$.

2) See fig. 2.

3) See fig. 4.

4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

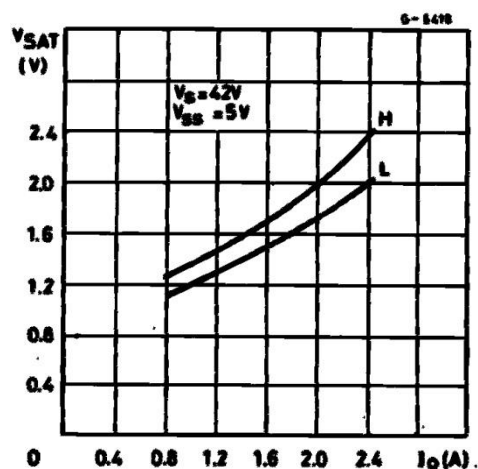
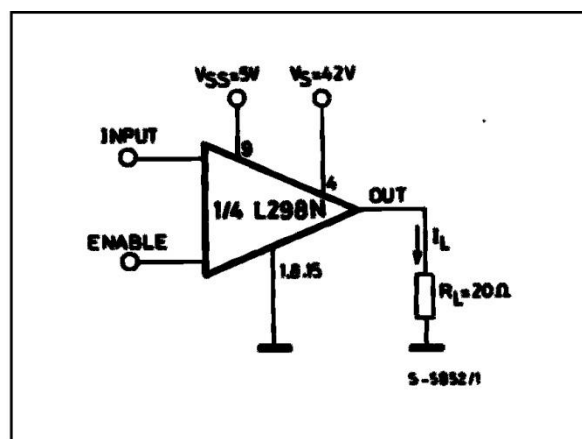


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

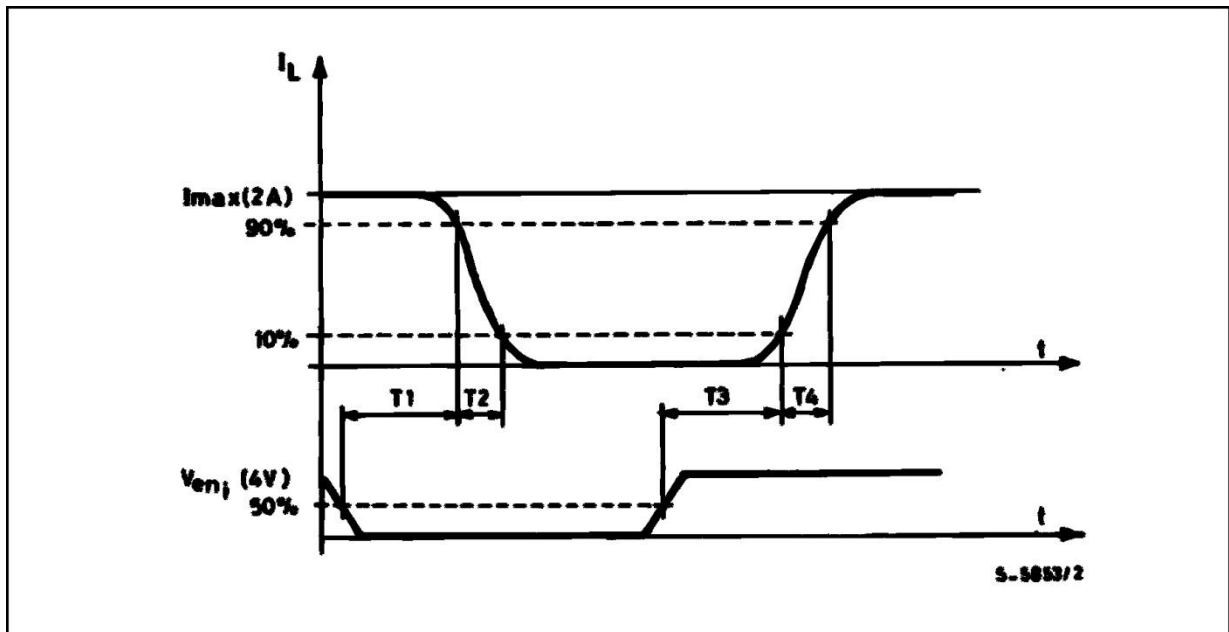
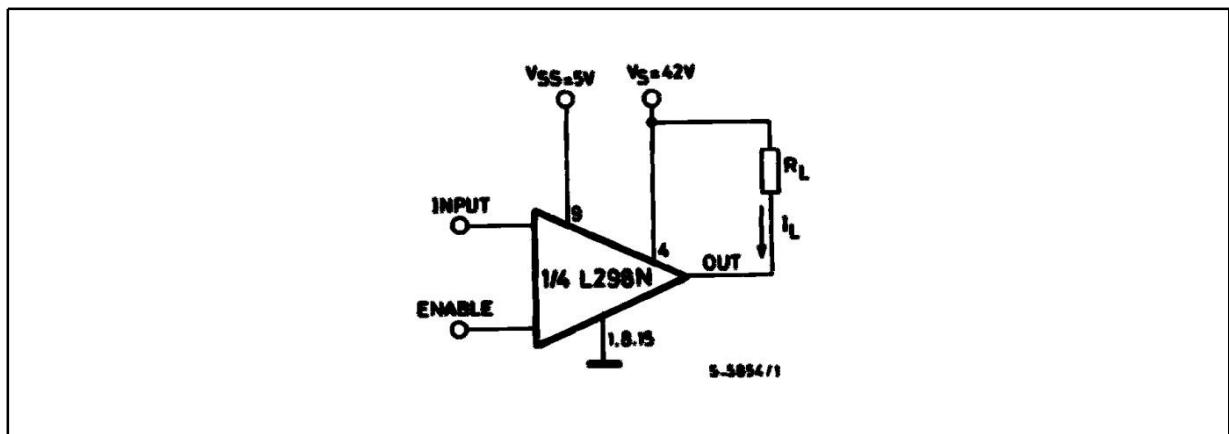


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = L

L298

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

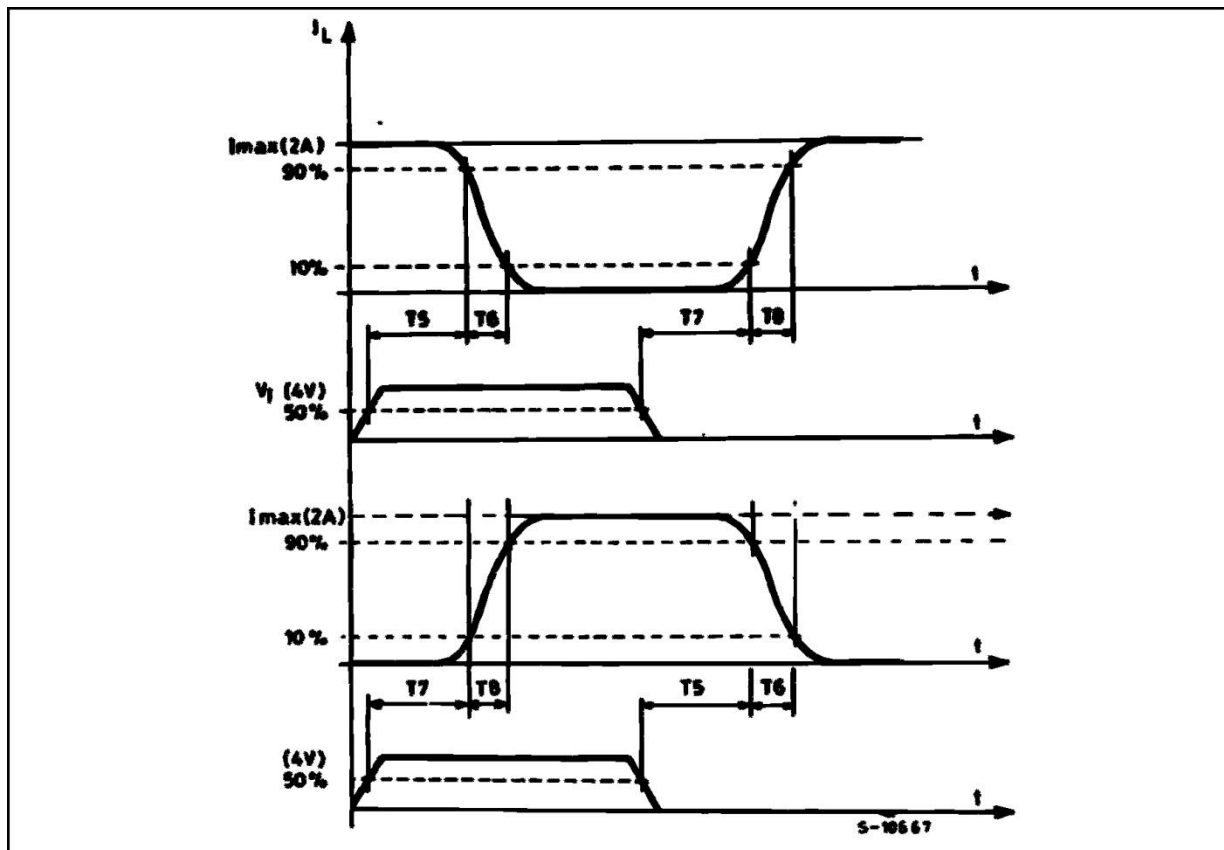


Figure 6 : Bidirectional DC Motor Control.

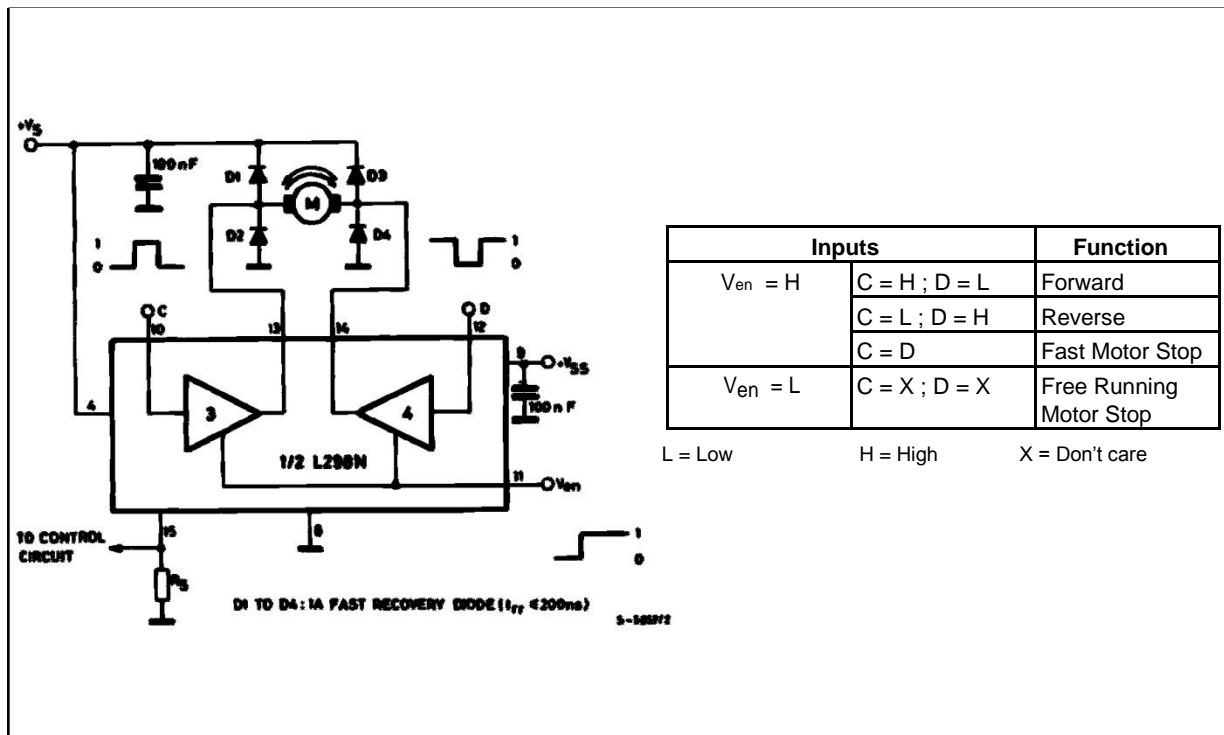
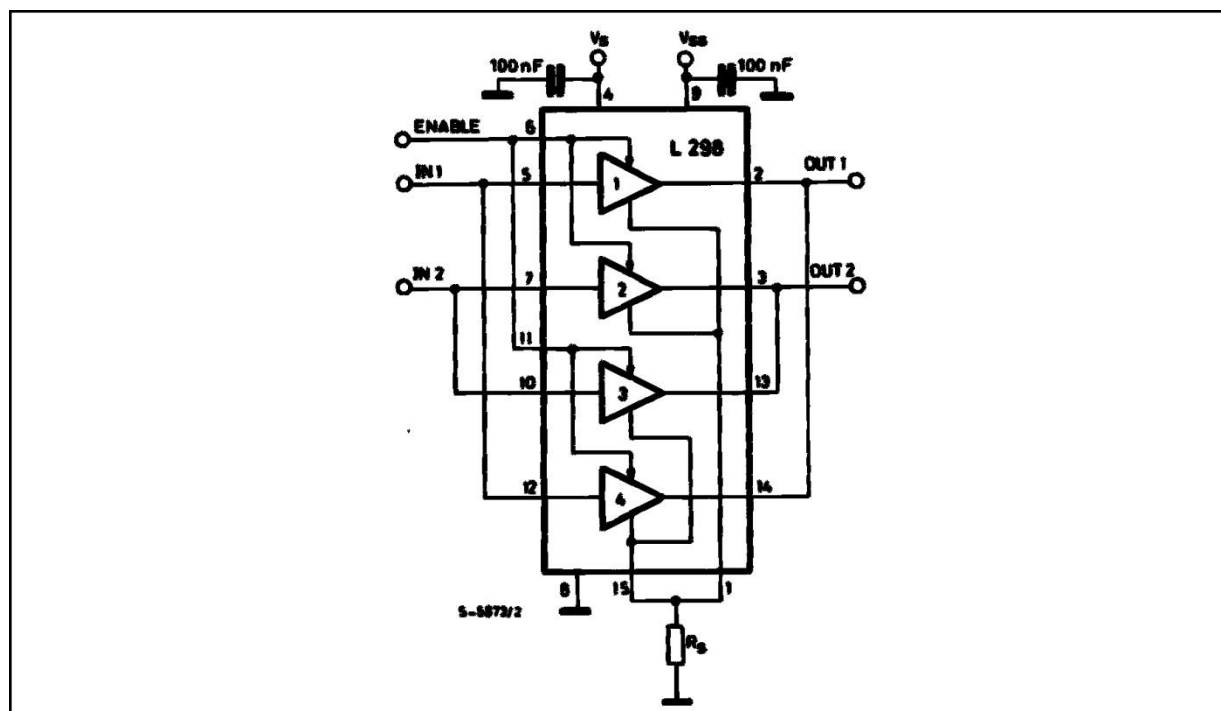


Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB} .) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are $In1$; $In2$; EnA and $In3$; $In4$; EnB . The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both V_s and V_{ss} , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of V_s that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Sup-ply Voltage and before to Turn it OFF, the Enable in-put must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes $D1$ to $D4$ is made by four fast recovery elements ($trr \leq 200$ nsec) that must be chosen of a V_F as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Schottky diodes would be preferred.

ANEXO C

L298

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

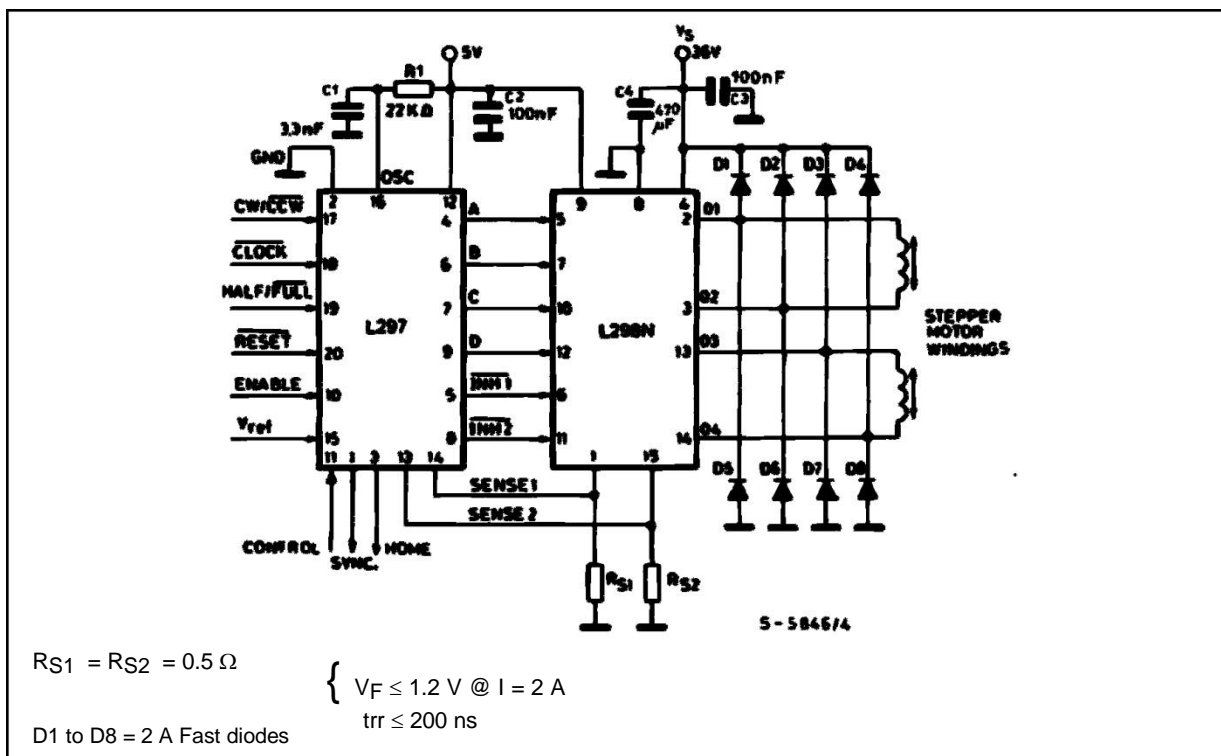


Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

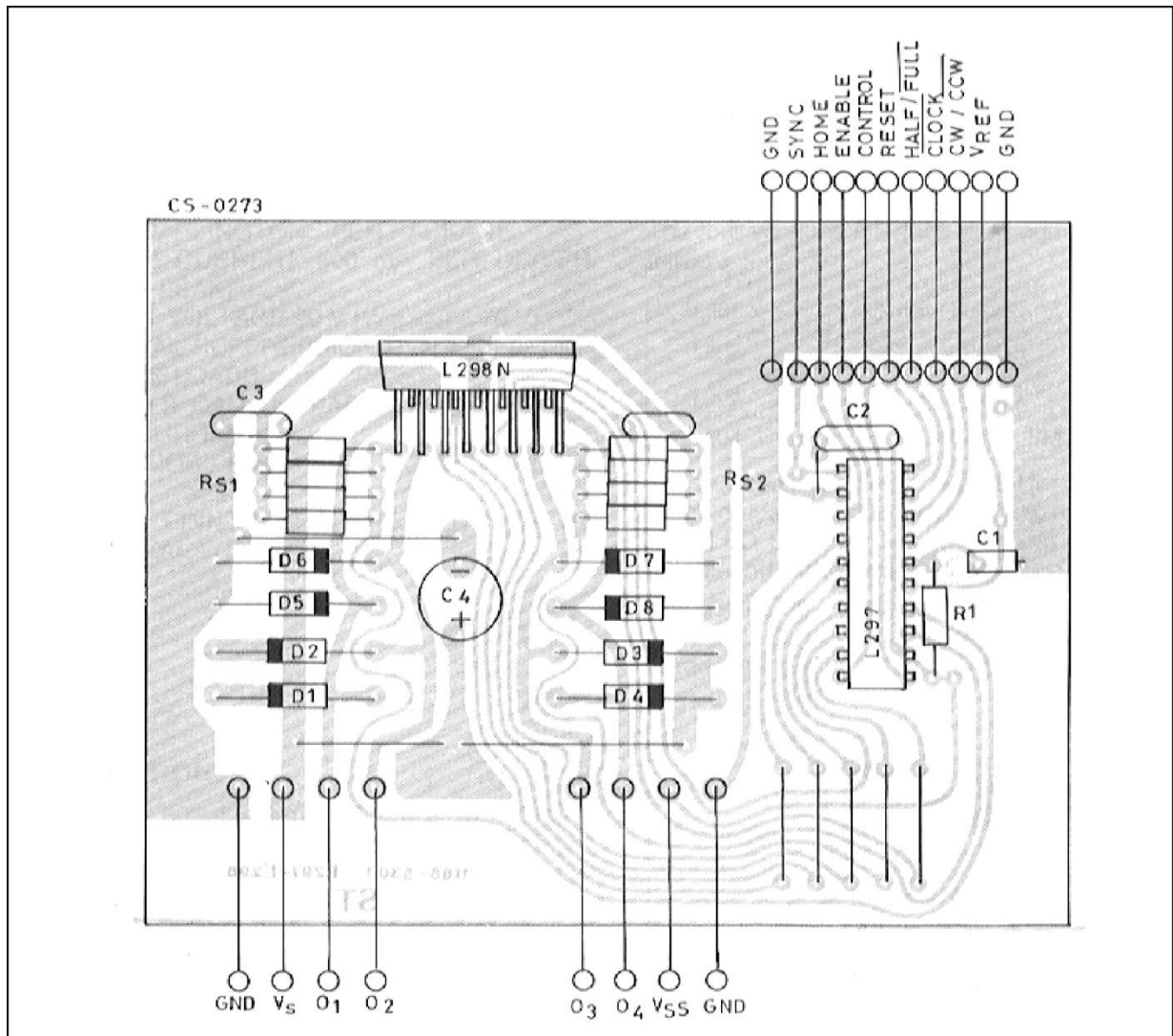
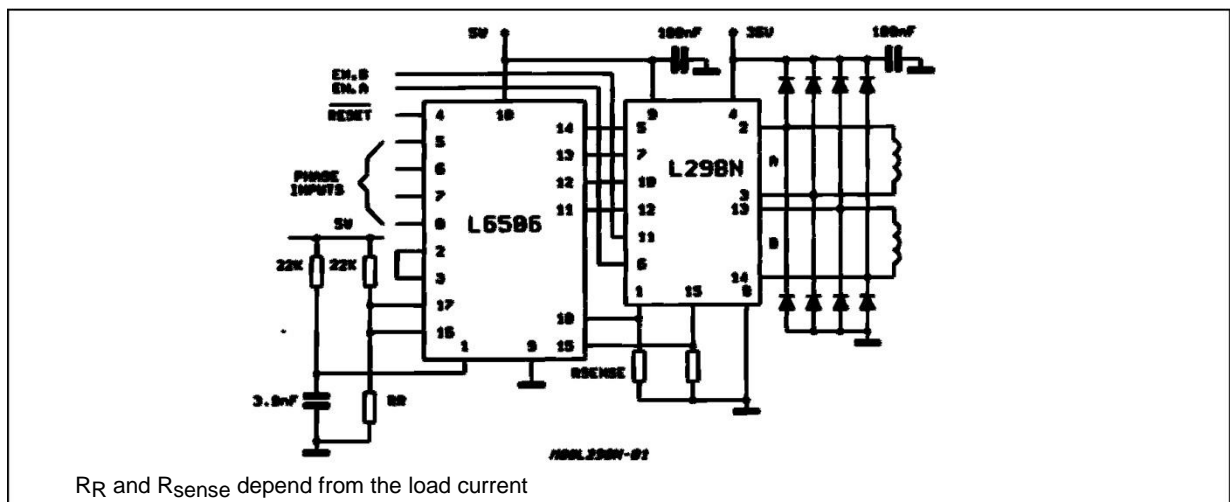


Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.

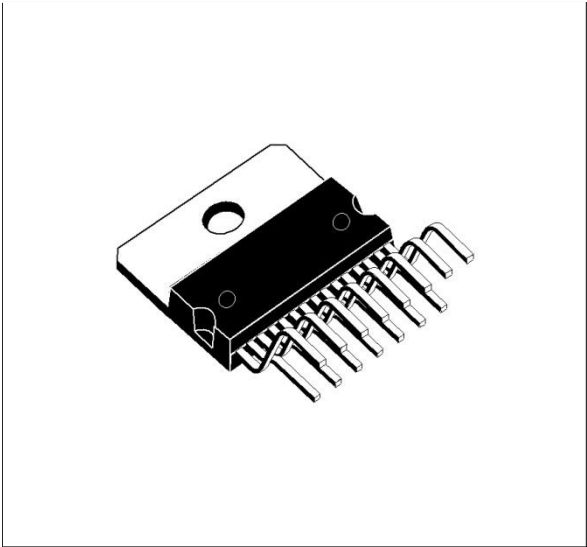


ANEXO C

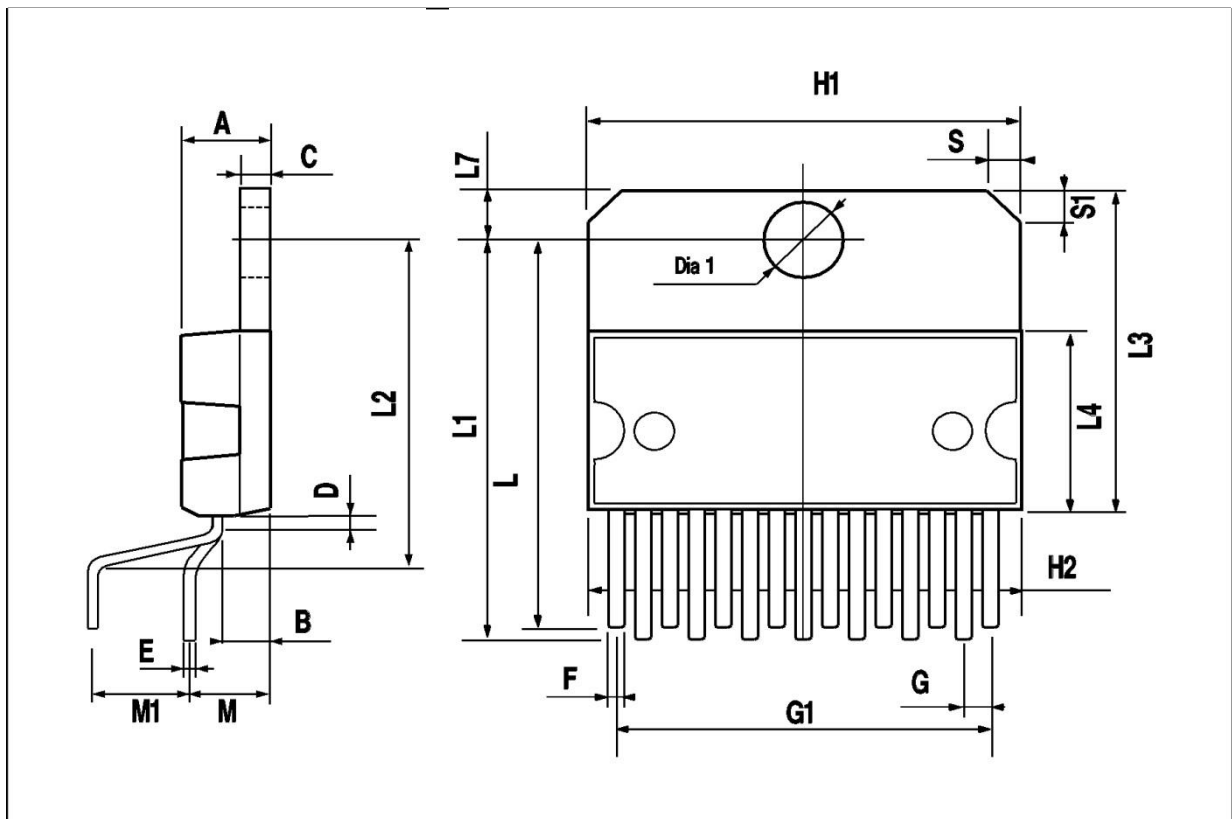
L298

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



Multiwatt15 V

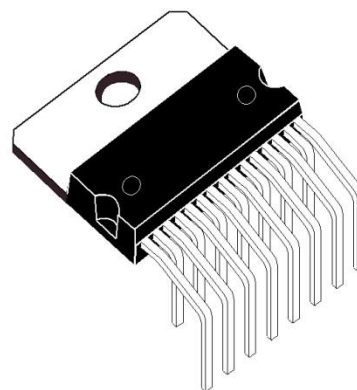


ANEXO C

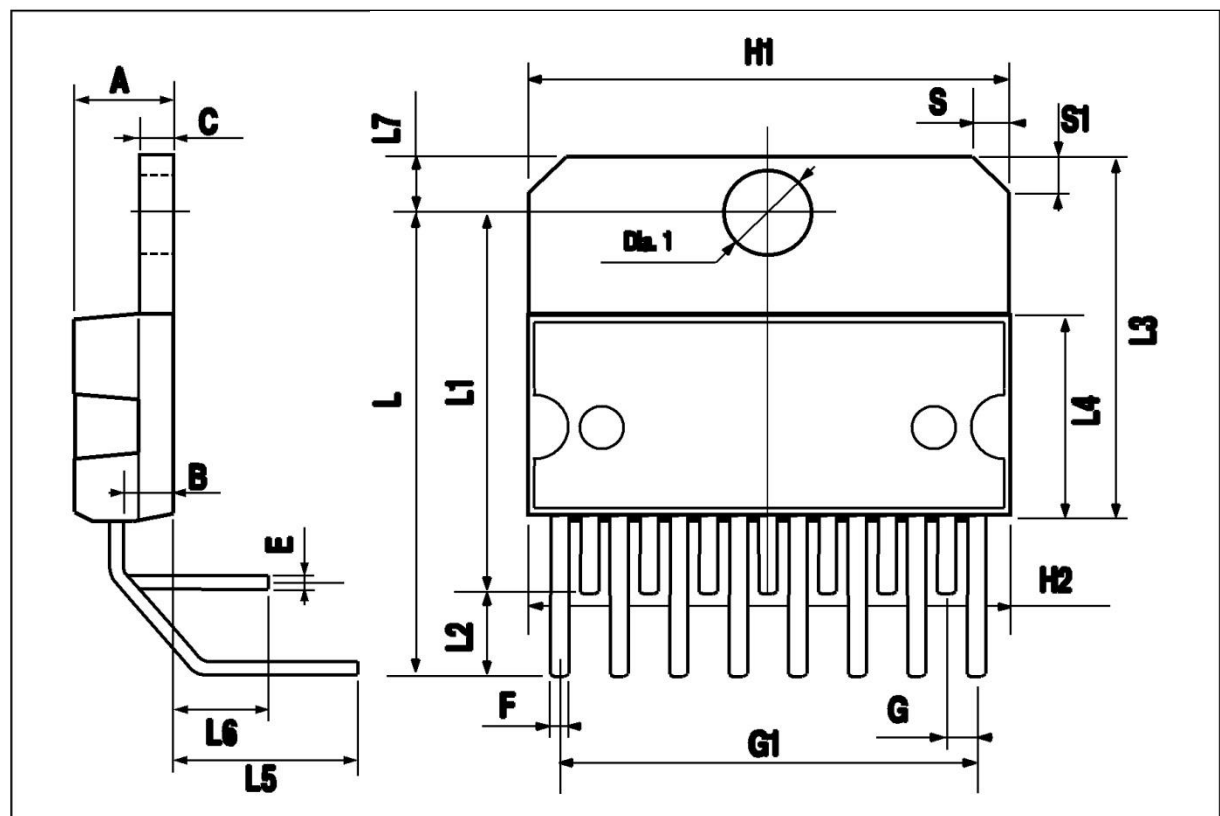
L298

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



Multiwatt15 H



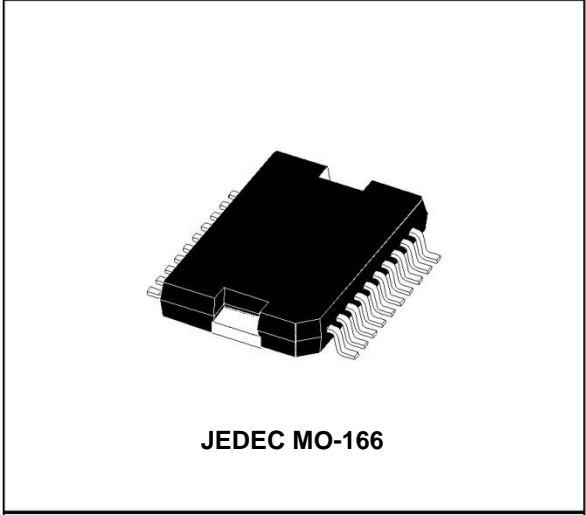
ANEXO C

L298

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

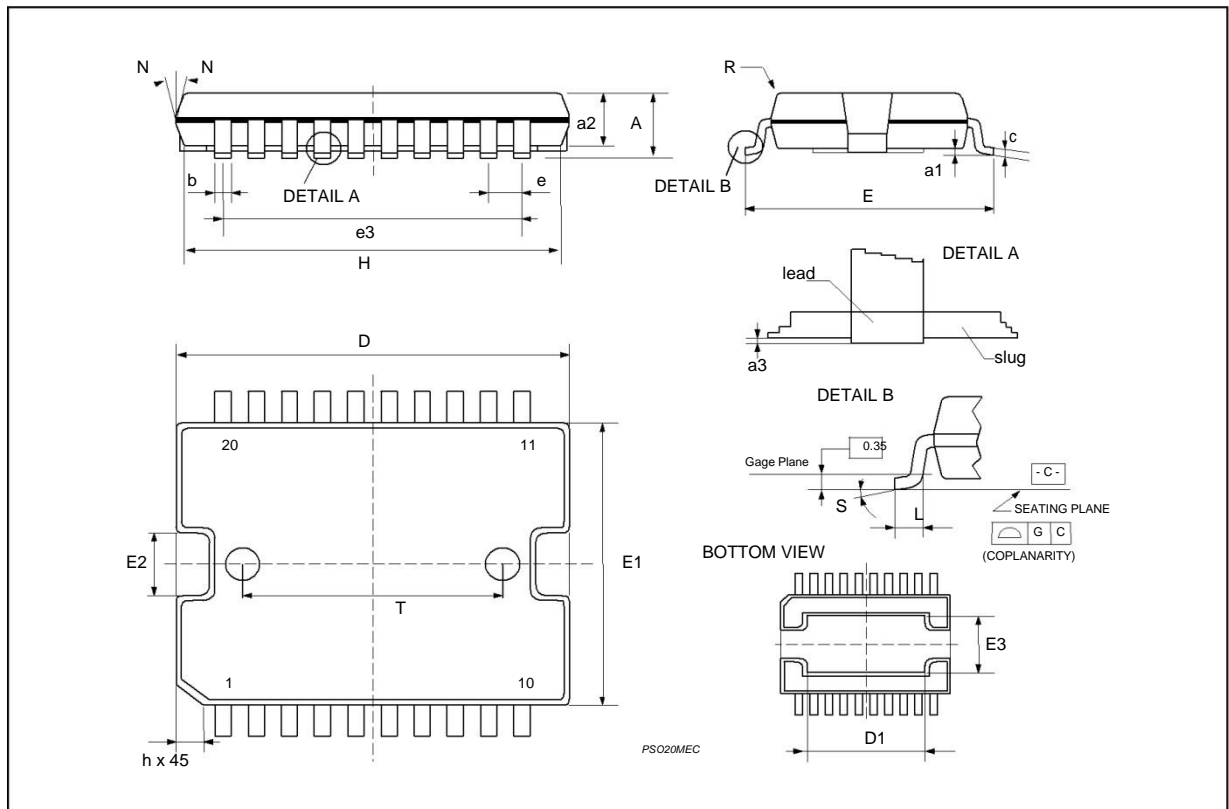
(1) "D and F" do not include mold flash or protrusions.
 - Mold flash or protrusions shall not exceed 0.15 mm (0.006").
 - Critical dimensions: "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA



JEDEC MO-166

PowerSO20



ANEXO C

TIP31C

TYPE NO.		IC (A) MAX	PD (W)	BV CBO (V) MIN	BV CEO (V) MIN	hFE		@ IC (A)	VCE(SAT) (V) MAX	@ IC (A)	fT (MHz) MIN
NPN	PNP					MIN	MAX				
SE9300	SE9400	10	70	60	60	1,000	--	4.0	2.0	4.0	1.0
SE9301	SE9401	10	70	80	80	1,000	--	4.0	2.0	4.0	1.0
SE9302	SE9402	10	70	100	100	1,000	--	4.0	2.0	4.0	1.0
TIP29	TIP30	1.0	30	40	40	15	75	1.0	0.7	1.0	3.0
TIP29A	TIP30A	1.0	30	60	60	15	75	1.0	0.7	1.0	3.0
TIP29B	TIP30B	1.0	30	80	80	15	75	1.0	0.7	1.0	3.0
TIP29C	TIP30C	1.0	30	100	100	15	75	1.0	0.7	1.0	3.0
TIP31	TIP32	3.0	40	40	40	10	50	3.0	1.2	3.0	3.0
TIP31A	TIP32A	3.0	40	60	60	10	50	3.0	1.2	3.0	3.0
TIP31B	TIP32B	3.0	40	80	80	10	50	3.0	1.2	3.0	3.0
TIP31C	TIP32C	3.0	40	100	100	10	50	3.0	1.2	3.0	3.0
TIP41	TIP42	6.0	65	40	40	15	75	3.0	1.5	6.0	3.0
TIP41A	TIP42A	6.0	65	60	60	15	75	3.0	1.5	6.0	3.0
TIP41B	TIP42B	6.0	65	80	80	15	75	3.0	1.5	6.0	3.0
TIP41C	TIP42C	6.0	65	100	100	15	75	3.0	1.5	6.0	3.0
TIP47		1.0	40	350	250	30	150	0.3	1.0	1.0	10
TIP48		1.0	40	400	300	30	150	0.3	1.0	1.0	10
TIP49		1.0	40	450	350	30	150	0.3	1.0	1.0	10
TIP50		1.0	40	500	400	30	150	0.3	1.0	1.0	10
TIP100	TIP105	8.0	80	60	60	1,000	20,000	3.0	2.0	3.0	4.0
TIP101	TIP106	8.0	80	80	80	1,000	20,000	3.0	2.0	3.0	4.0
TIP102	TIP107	8.0	80	100	100	1,000	20,000	3.0	2.0	3.0	4.0
TIP110	TIP115	2.0	50	60	60	500	--	2.0	2.5	2.0	25
TIP111	TIP116	2.0	50	80	80	500	--	2.0	2.5	2.0	25
TIP112	TIP117	2.0	50	100	100	500	--	2.0	2.5	2.0	25
TIP120	TIP125	5.0	65	60	60	1,000	--	3.0	2.0	3.0	4.0
TIP121	TIP126	5.0	65	80	80	1,000	--	3.0	2.0	3.0	4.0
TIP122	TIP127	5.0	65	100	100	1,000	--	3.0	2.0	3.0	4.0
TIP130	TIP135	8.0	70	60	60	1,000	15,000	4.0	2.0	4.0	--
TIP131	TIP136	8.0	70	80	80	1,000	15,000	4.0	2.0	4.0	--
TIP132	TIP137	8.0	70	100	100	1,000	15,000	4.0	2.0	4.0	--

Shaded areas indicate Darlington.
Available in TO-220FP Full Pak upon request.

ANEXO C

Electroválvula rosca 1/4 pulgada, 110Vac, 220 VAC, 12 VDC, 24 VDC

Ideal para pequeños proyectos con agua.

**Specifications:**

Model	2P025-06	2P025-08	2P025-06A	2P025-08A
Fluid	Air, Water, Oil, Gas			
Operation	Direct Drive Type			
Type	Normal Close Type			
Aperture of flow rate(mm)	2.5			
Cv value	0.23			
Port size	NPT1/8"/1/4"			
Operation fluid viscosity	20 CST Below			
Operation pressure	0 ~ 0.7MPa			
Max.Proof pressure	1.05MPa			
Operation temperature range	-5 ~ +80C			
Voltage range	±10%			
Conventional Voltage	12V 24V 110V 220V			
Material of body	Engineering Plastic			
Material of oil seal	NBR, EPDM or VITON			

REFERENCE

SPEC.No. ED-05G061
ISSUE March 1, 2005

SHARP

OPTO-ELECTRONIC DEVICES DIVISION
ELECTRONIC COMPONENTS GROUP
SHARP CORPORATION

SPECIFICATION

Sensor Sharp ~~GP2D120XJ00F~~ SPECIFICATION FOR

Analog Output Type Distance Measuring Sensor

MODEL No.

GP2D120XJ00F

Specified for

Enclosed please find copies of the Specifications which consists of 10 pages including cover.
After confirmation of the contents, please be sure to send back copies of the Specifications
with approving signature on each.

CUSTOMER'S APPROVAL

PRESENTED

DATE

DATE

BY

BY

H. O

H. Ogura,
Department General Manager of
Engineering Dept. III
Opto-Electronic Devices Div.
ELECTOM Group
SHARP CORPORATION

REFERENCE

Product name : Analog output type distance measuring Sensor

Model No. : GP2D120XJ00F

1. These specification sheets include materials protected under copyright of Sharp Corporation ("Sharp"). Please do not reproduce or cause anyone to reproduce them without Sharp's consent.
2. When using this product, please observe the absolute maximum ratings and the instructions for use outlined in these specification sheets, as well as the precautions mentioned below. Sharp assumes no responsibility for any damage resulting from use of the product which does not comply with the absolute maximum ratings and the instructions included in these specification sheets, and the precautions mentioned below.

(Precautions)

- (1) This product is designed for use in the following application areas ;

{
 • Computers • OA equipment • Telecommunication equipment (Terminal)
 • Measuring equipment • Tooling machines • Audio visual equipment
 • Home appliances
 }

If the use of the product in the above application areas is for equipment listed in paragraphs (2) or (3), please be sure to observe the precautions given in those respective paragraphs.

- (2) Appropriate measures, such as fail-safe design and redundant design considering the safety design of the overall system and equipment, should be taken to ensure reliability and safety when this product is used for equipment which demands high reliability and safety in function and precision, such as ;

{
 • Transportation control and safety equipment (aircraft, train, automobile etc.)
 • Traffic signals • Gas leakage sensor breakers • Rescue and security equipment
 • Other safety equipment
 }

- (3) Please do not use this product for equipment which require extremely high reliability and safety in function and precision, such as ;

{
 • Space equipment • Telecommunication equipment (for trunk lines)
 • Nuclear power control equipment • Medical equipment
 }

- (4) Please contact and consult with a Sharp sales representative if there are any questions regarding interpretation of the above three paragraphs.

3. Please contact and consult with a Sharp sales representative for any questions about this product.

REFERENCE

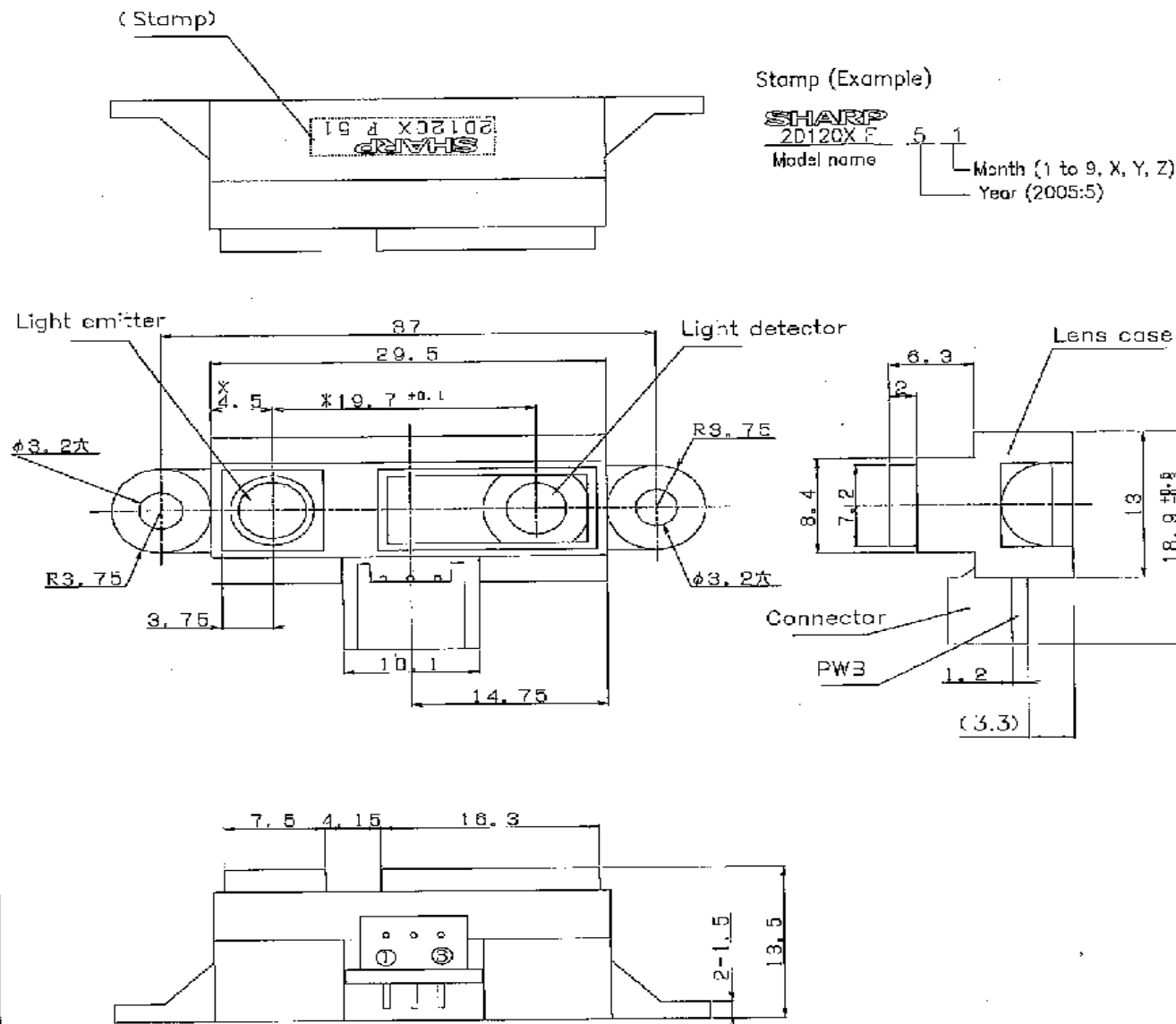
1. Application
This specification is applied for the outline and the characteristics of the analog output distance measuring sensor ;
Model No. GP2D120XJ00F.
2. Outline
Refer to the attached drawing No. SOE005102.
3. Ratings and characteristics
Refer to the attached sheet, page 4, 5.
4. Reliability
Refer to the attached sheet, Page 6.
5. Outgoing inspection
Refer to the attached sheet, Page 6.
6. Supplements
 - 6-1 GP2D120XJ00F Example of output distance characteristics
Refer to the attached sheet, page 8.
 - 6-2 GP2D120XJ00F Example of output characteristics with inverse number of distance
Refer to the attached sheet, page 9.
 - 6-3 This product shall not contain the following materials.
Also, the following materials shall not be used in the production process for this product.
Materials for ODS : CFCs, Halon, Carbon tetrachloride 1.1.1-Trichloroethane (Methyl chloroform)
 - 6-4 Product mass : Approx. 3.6g (TYP)
 - 6-5 This product does not contain the chemical materials regulated by RoHS.
(Except for the NOT regulated by RoHS)
7. Notes
Refer to the attached sheet, page 7.

REFERENCE

ED-05G061 GP2D120XJ00F 3/9

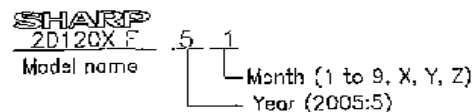
March 1, 2005

2. Outline



(Stamp)

Stamp (Example)



Connector signal

Signal name
① V _o
② GND
③ V _{cc}

Connector :
J.S.T.TRADING COMPANY,LTD.
S3B-PH

Materials

Lens: Acrylic acid resin
(Visible light cut-off resin)
Case: Carbonic ABS
(Conductive resin)
PCB: Paper phenol

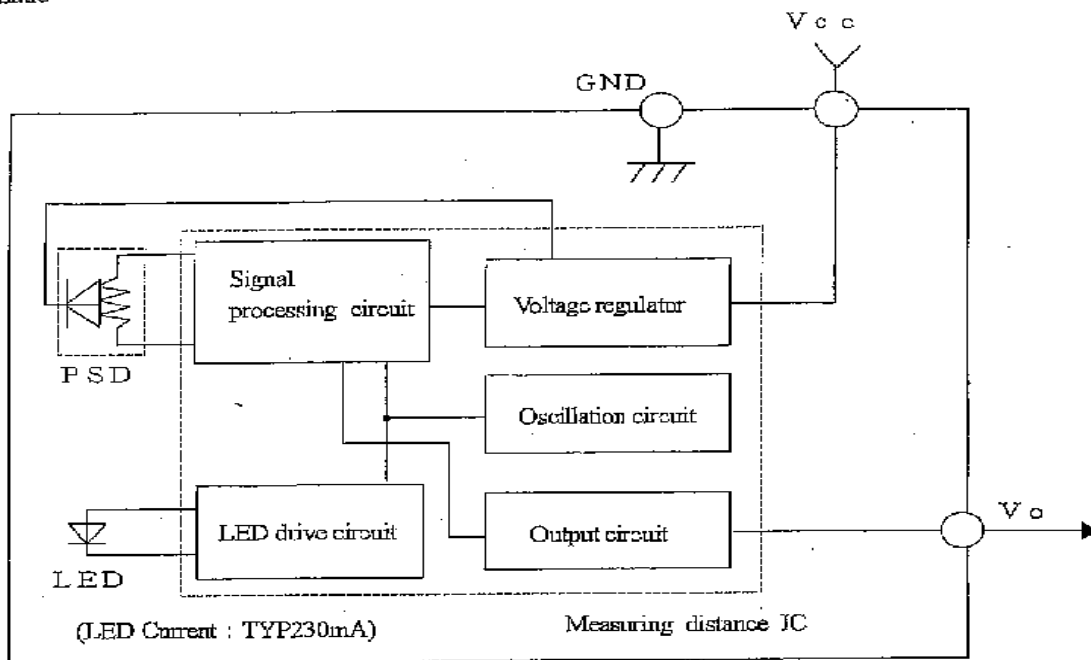
Unit: mm Scale: 2/1

Note 1: * dimensions shall be reference lens center.
Note 2: Unspecified tolerance shall be ± 0.3 mm.
Note 3: The dimensions in parenthesis are shown for reference

Name	GP2D120XJ00F Outline Dimensions
Drawing No.	SOE005102

REFERENCE

3-1 Schematic



3-2 Absolute maximum ratings

(Ta=25°C, Vcc=5V)

Parameter	Symbol	Ratings	Unit	Remark
Supply voltage	Vcc	-0.3 to +7	V	-
Output terminal voltage	Vo	-0.3 to Vcc+0.3	V	-
Operating temperature	Topr	-10 to +60	°C	-
Storage temperature	Tstg	-40 to +70	°C	-

Operating supply voltage

Symbol	Rating	Unit	Remark
Vcc	4.5 to 5.5	V	-

REFERENCE

ED-05G061 GP2D120XJ00F
March 1, 2005

5/9

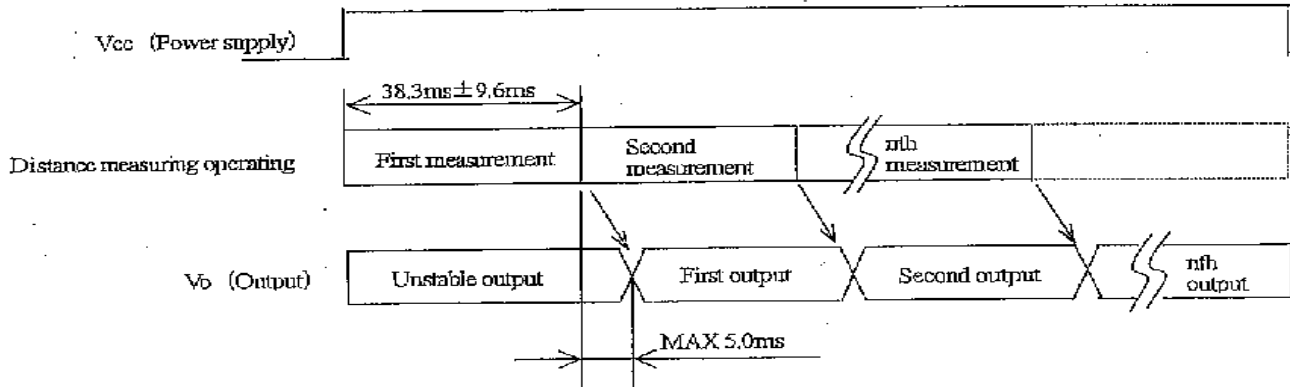
3-3 Electro-optical Characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Measuring distance range	ΔL	(Note 1)	4	-	30	cm
Output terminal voltage	V_o	L=30cm (Note 1)	0.25	0.4	0.55	V
Output voltage difference	ΔV_o	Output change at L change (30cm \rightarrow 4cm) (Note 1)	1.95	2.25	2.55	V
Average supply current	I_{oc}	L=30cm (Note 1)	-	33	50	mA

※ L: Distance to reflective object

(Note 1) Using reflective object: White paper (Made by Kodak Co., Ltd. gray cards R-27 white face, reflective ratio; 90%)

3-4 Timing chart



REFERENCE

4. Reliability

The reliability of products shall be satisfied with items listed below.

Confidence level : 90%

LTPD : 20 or 30

No.	Test Items	Test Conditions	Failure Judgement Criteria	Samples (n)
				Defective (c)
1	Temperature cycling	1 cycle -40°C to +70°C (30min.) (30min.) 25 cycle test	Initial $\times 0.8 > V_0$ $V_0 > \text{Initial} \times 1.2$ (Note 1)	n=11, c=0
2	High temp. and high humidity storage	+40°C, 90%RH, 500h		n=11, c=0
3	High temp. storage	+70°C, 500h		n=11, c=0
4	Low temp. storage	-40°C, 500h		n=11, c=0
5	Operation life (High temp.)	+60°C, Vcc=5V, 500h		n=11, c=0
6	Mechanical shock	1000m/s ² , 6.0ms 3times/±X, ±Y, ±Z direction		n=8, c=0
7	Variable frequency vibration	10 to 55 to 10Hz/1min. 2h/X, Y, Z direction overall amplitude : 1.5mm		n=8, c=0

(Note 1) Test conditions are according to 3-3 Electro-optical characteristics.

(Note 2) After test, measurement shall be measured after leaving under the normal temperature and the normal humidity for two hours. But no dew point.

5. Outgoing inspection

(1) Inspection lot

Inspection shall be carried out per each delivery lot.

(2) Inspection method

A single sampling plan, normal inspection level II based on ISO 2859 is applied.
The AQL according to the inspection items are shown below.

Defect	Inspection item	AQL (%)
Major defect	Electro-optical characteristics defect (in para. 3-3)	0.4
Minor defect	Defect on appearance and dimension ※ Crack, chip, scratch, stain	1.0

※ Crack, chip, scratch, stain

One which affects the characteristics of para. 3-3 shall be defect.

REFERENCE

7. Notes

[Advice for the optics]

7-1 Lens of this device shall be kept cleanly. There are cases that dust, water or oil and so on deteriorate the characteristics of this device. Please consider in actual application.

7-2 In case that protection is set in front of the emitter and detector portion, the protection cover which has the most efficient transmittance at the emitting wavelength range of LED for this product ($\lambda=850\text{nm}\pm 70\text{nm}$), shall be recommended to use. The face and back of protection cover should be mirror polishing. Also, as there are cases that the characteristics may not be satisfied with according to the distance between the protection cover and this product or the thickness of the protection cover, please use this product after confirming the operation sufficiently in actual application.

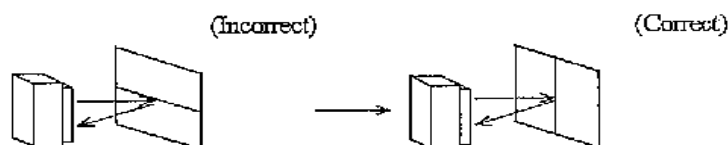
[Advice for the characteristics]

7-3 In case that there is an object near to light exits of the sensor between the sensor and the detected object, please use this device after confirming sufficiently what the characteristics of this sensor do not change by the object.

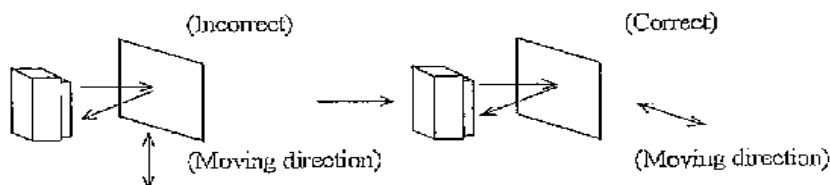
7-4 When the detector surface receive direct light from the sun, tungsten lamp and so on, there are cases that it can not measure the distance exactly. Please consider the design that the detector does not receive direct light from such light source.

7.5 Distance between sensor and mirror reflector can not sometimes measure exactly.
In case of changing the mounting angle of this product, it may measure the distance exactly.

7.6 In case that reflective object has boundary line clearly, there is cases that distance can not measure exactly. At that time, if direction of boundary line and the line between emitter center and detector center parallels, it is possible to decrease deviation of measuring distance.



7-7 In order to decrease measuring error by moving direction of object, we recommend to mount the sensor like below drawing.



7-8 In order to stabilize power supply line, we recommend to connect a by-pass capacitor of $10\mu\text{F}$ or more between Vcc and GND near this product.

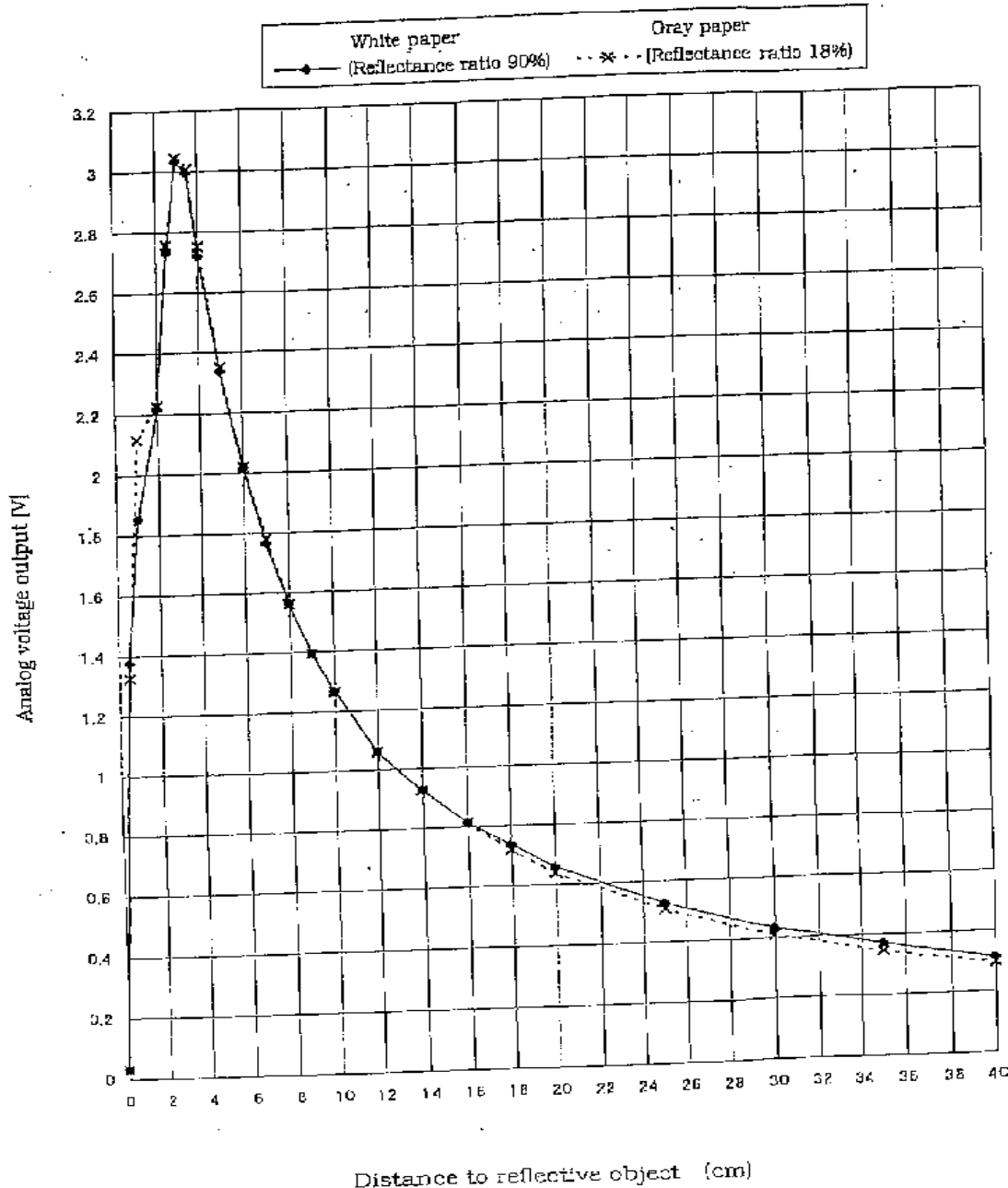
[Notes on handling]

7-9 Please don't do washing. Washing may deteriorate the characteristics of optical system and so on. Please confirm resistance to chemicals under the actual usage since this product has not been designed with the consideration.

7-10 There are some possibilities that the sensor inside the case package with lens may be exposed to the excessive mechanical stress. Please be careful not to cause any excessive pressure on the case package with lens and also on the PCB at the assembly and inserting of the set.

REFERENCE

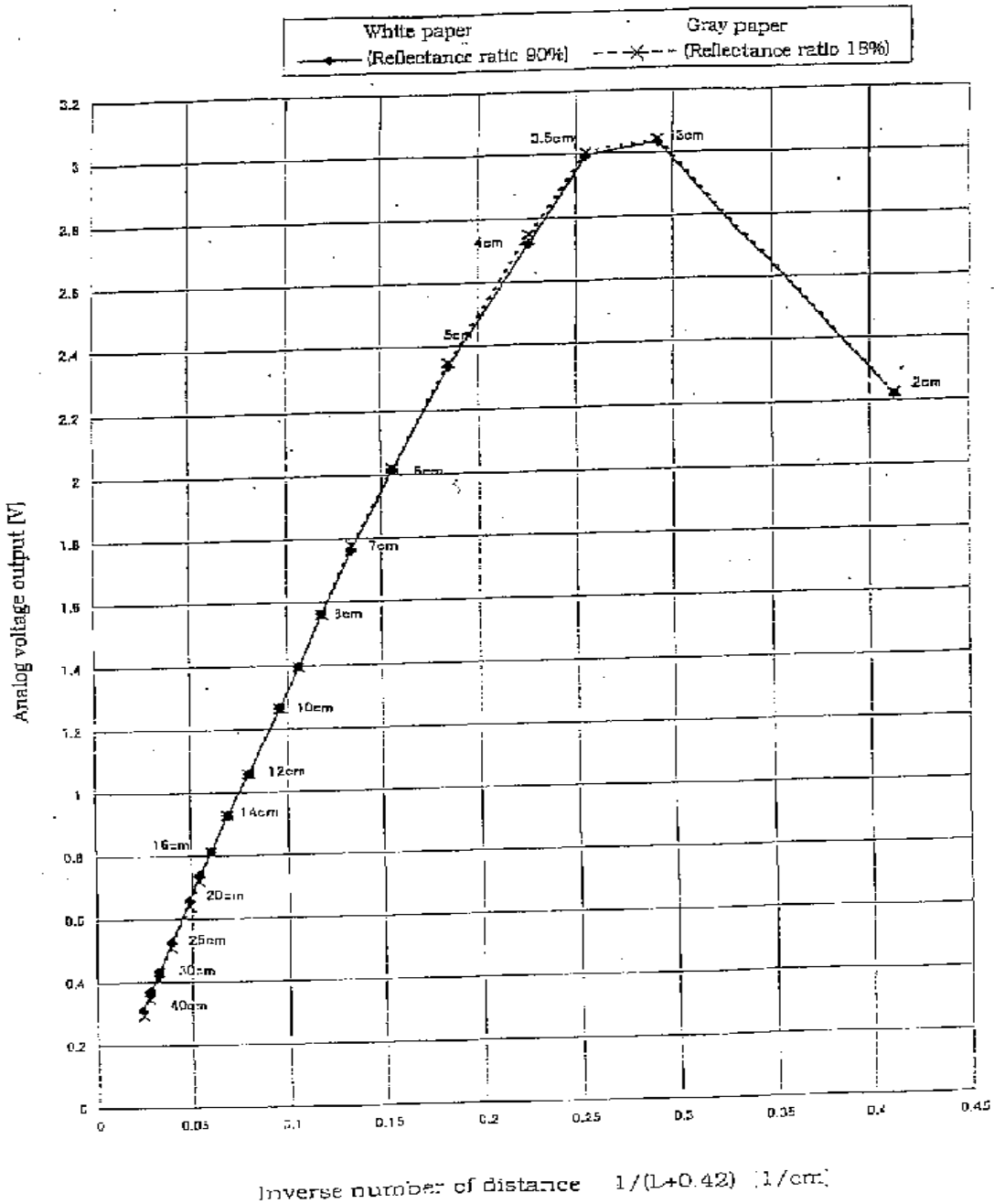
6-1 GP2D120XJ00F Example of output distance characteristics



REFERENCE

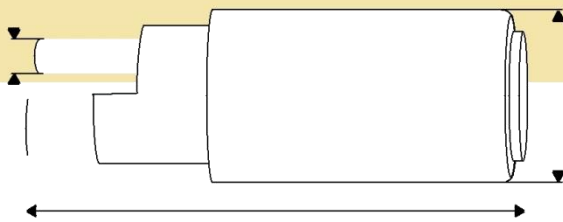
ED 05G061 GP2D120XJ00F
9/9
March 1, 2005

6-2 GP2D120XJ00F Example of output distance characteristics with inverse number of distance





Bombas de Combustible Internas al Tanque



Tubo de salida con punta lisa

Referencia	Presión de prueba Bar Lbs/pul ²	Caudal/litros (medidos en 30 s en la presión de prueba) Mínimo	Corriente máxima A	Voltaje V	A Diámetro externo (mm)	B Largo total (mm)	C Diámetro del tubo de salida (mm)	Conexión eléctrica de la bomba
0 580 453 053	3,1...3,5 bar 44...50 lbs	1	9	12	43	140	8	Terminal plano
0 580 453 081	3,6...4,0 bar 51...57 lbs	0,9	13	12	43	140	8	Tornillo con rosca M4/M5
0 580 453 417	0,9...1,1 bar 12,7...15,6 lbs	0,66	6	12	38	111,4	8,8	Terminal plano
0 580 453 507	0,9...1,1 bar 12,7...15,6 lbs	0,7	6	12	43	125	8	Terminal plano
0 580 453 509	0,68...0,76 bar 9,6...10,7 lbs	0,58	4,5	12	43	125	8	Terminal plano
0 580 453 514	0,9...1,1 bar 12,7...15,6 lbs	0,7	6	12	43	125	8	Terminal plano
0 580 453 900	2,7...3,0 bar 38,3...42,6 lbs	0,62	4,5	12	43	148	8	Terminal plano
0 580 453 915	2,7...3,0 bar 38,3...42,6 lbs	0,62	5,2	12	43	163	9	Terminal plano
0 580 453 966	2,7...3,0 bar 38,3...42,6 lbs	0,7	6	12	43	146,5	9,5	Terminal plano
0 580 464 985	0,9...1,1 bar 12,7...15,6 lbs	0,78	4	12	52	151	8	Tornillo con rosca M4/M5
0 580 464 085	2,2...2,5 bar 31...35,5 lbs	0,79	4,5	12	52	175	8	Tornillo con rosca M4/M5
0 580 453 470	2,7...3,0 bar 38,3...42,6 lbs	1	7,3	12	38	111,4	8,8	Terminal plano
0 580 453 477	2,7...3,0 bar 38,3...42,6 lbs	0,87	5,8	12	38	111,4	8,8	Terminal plano
0 580 453 483	0,9...1,1 bar 12,7...15,6 lbs	0,83	3,5	12	38	111,4	8,8	Terminal plano
0 580 453 606	3,6...4,0 bar 51...57 lbs	0,95	8	12	38	111,4	8,8	Terminal plano
0 580 453 607	0,9...1,1 bar 12,7...15,6 lbs	0,83	3,5	12	38	111,4	8,8	Terminal plano
0 580 453 985	3,5...3,8 bar 50...54 lbs	0,7	7	12	43	146,5	9,5	Terminal plano



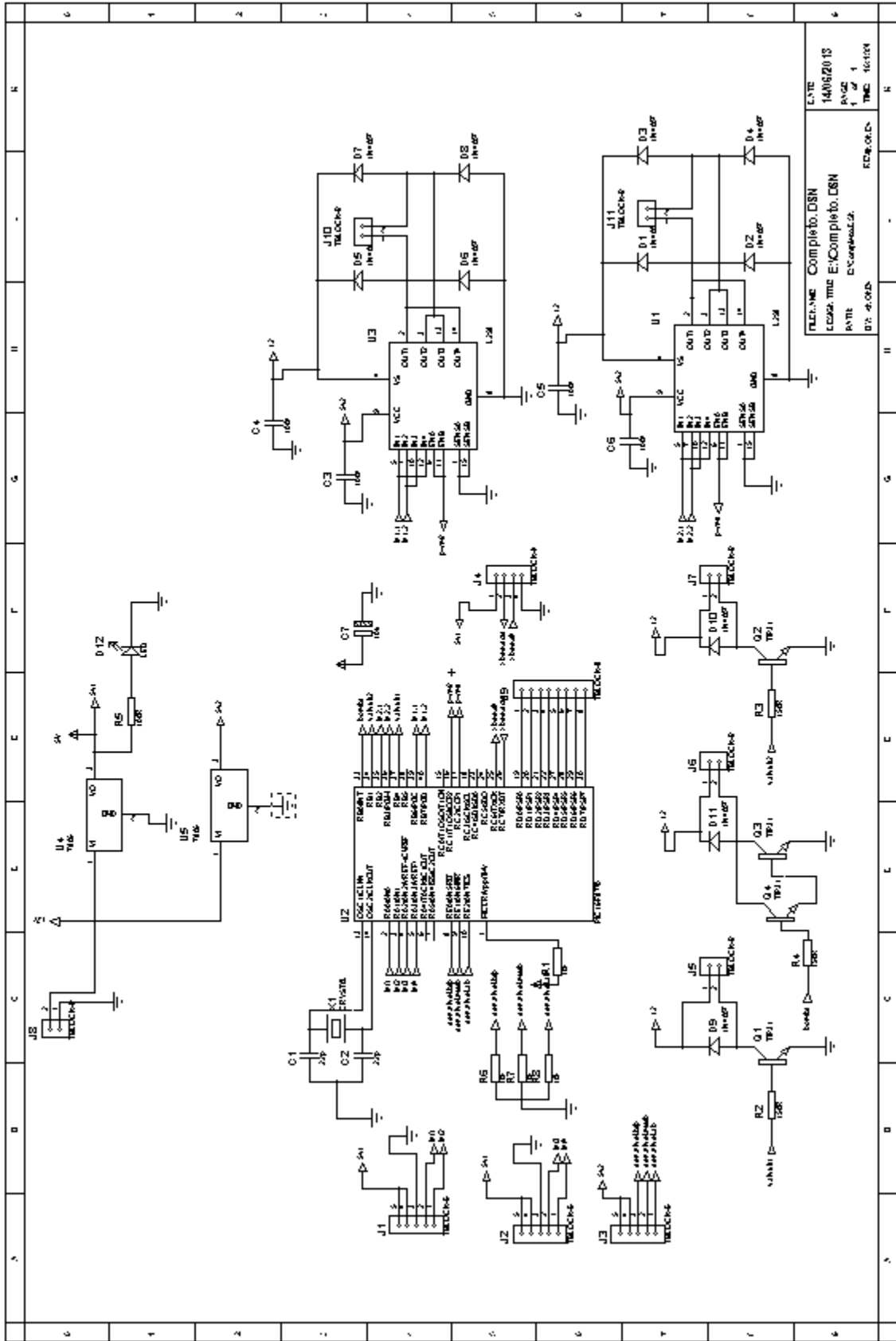
Tubo de salida con punta estriada

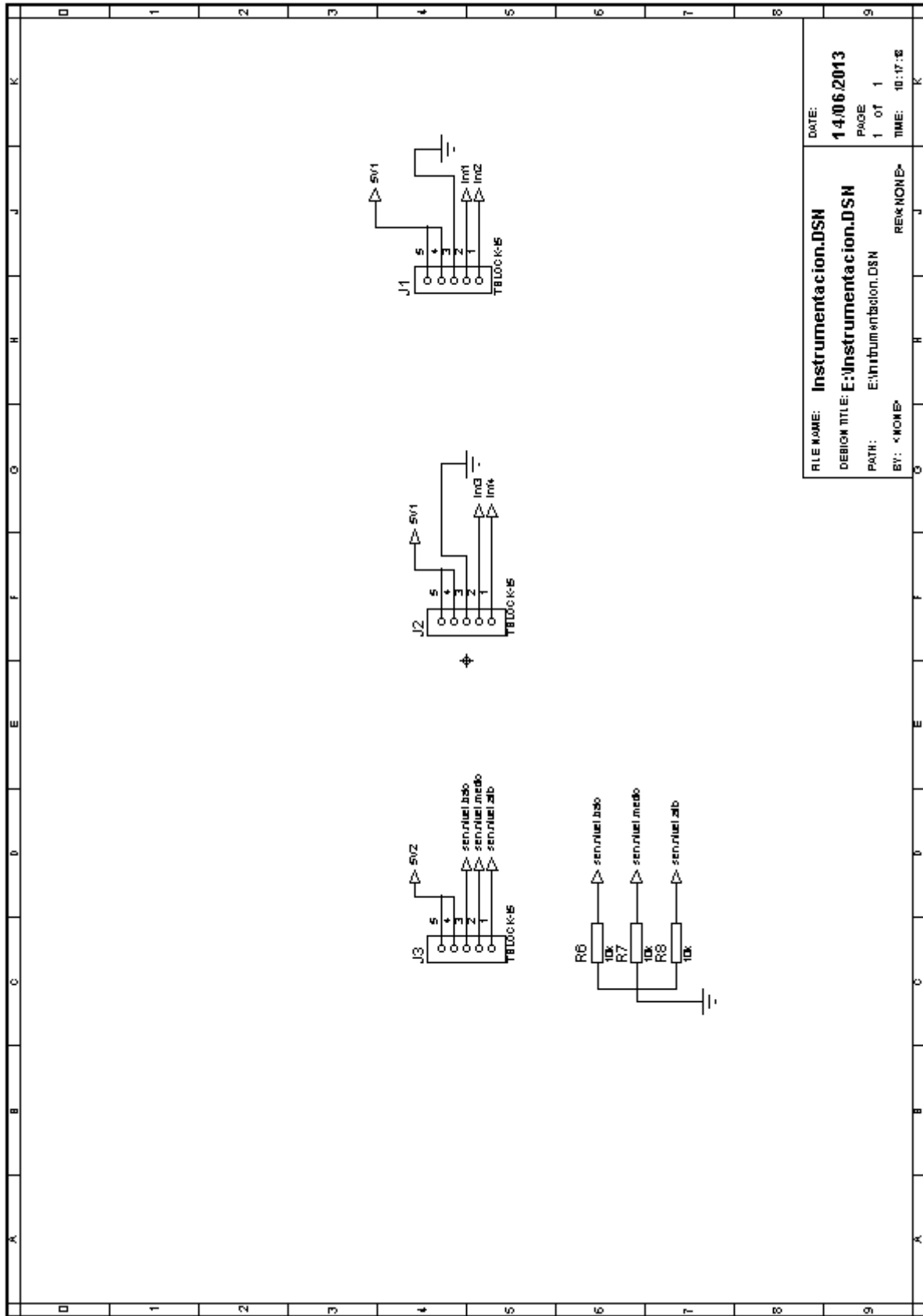
Referencia	Presión de prueba. Bar Lbs/pul ²	Caudal/litros (medidos en 30 s en la presión de prueba) Mínimo	Corriente máxima A	Voltaje V	A Diámetro externo (mm)	B Largo total (mm)	C Diámetro del tubo de salida (mm)	Conexión eléctrica de la bomba
0 580 453 976	2,7...3,0 bar 38,3...42,6 lbs	0,83	6	12	43	146,5	9,5	Terminal plano
0 580 453 443	3,6...4,0 bar 51...57 lbs	0,91	7,5	12	38	114,9	9,2	Terminal plano
0 580 453 471	2,7...3,0 bar 38,3...42,6 lbs	0,87	5,8	12	38	111,4	9,2	Terminal plano
0 580 453 481	3,6...4,0 bar 51...57 lbs	0,83	8	12	38	111,4	8,8	Terminal plano
0 580 453 482	0,9...1,1 bar 12,7...15,6 lbs	0,83	3,5	12	38	111,4	9,2	Terminal plano
0 580 453 490	2,7...3,0 bar 38,3...42,6 lbs	0,87	5	12	38	111,4	9,2	Terminal plano

Los datos son válidos para bombas nuevas y combustible a una temperatura de 23 °C, aproximadamente. Bombas usadas pueden presentar valores de caudal hasta 20% a menos.

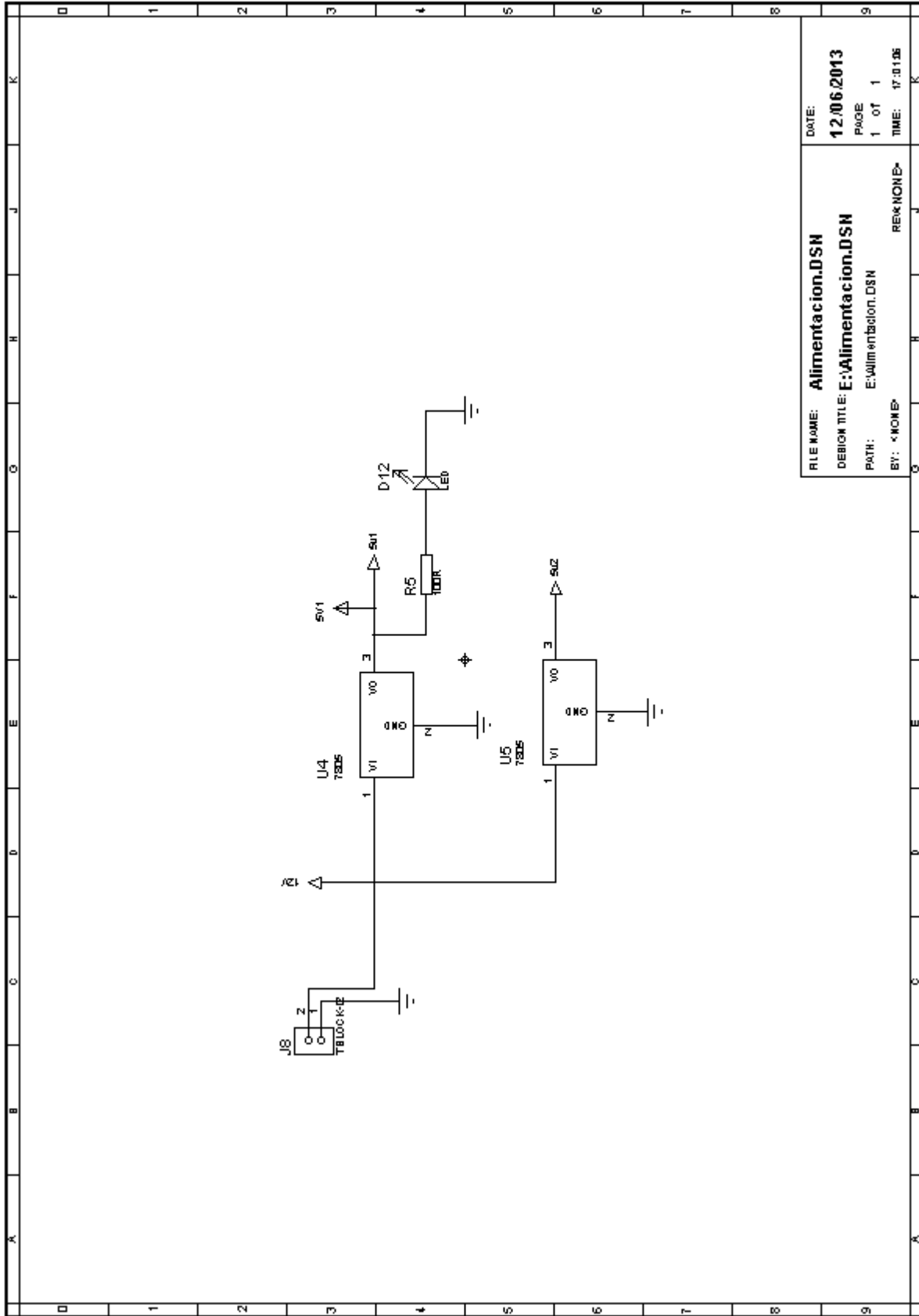
ANEXO D

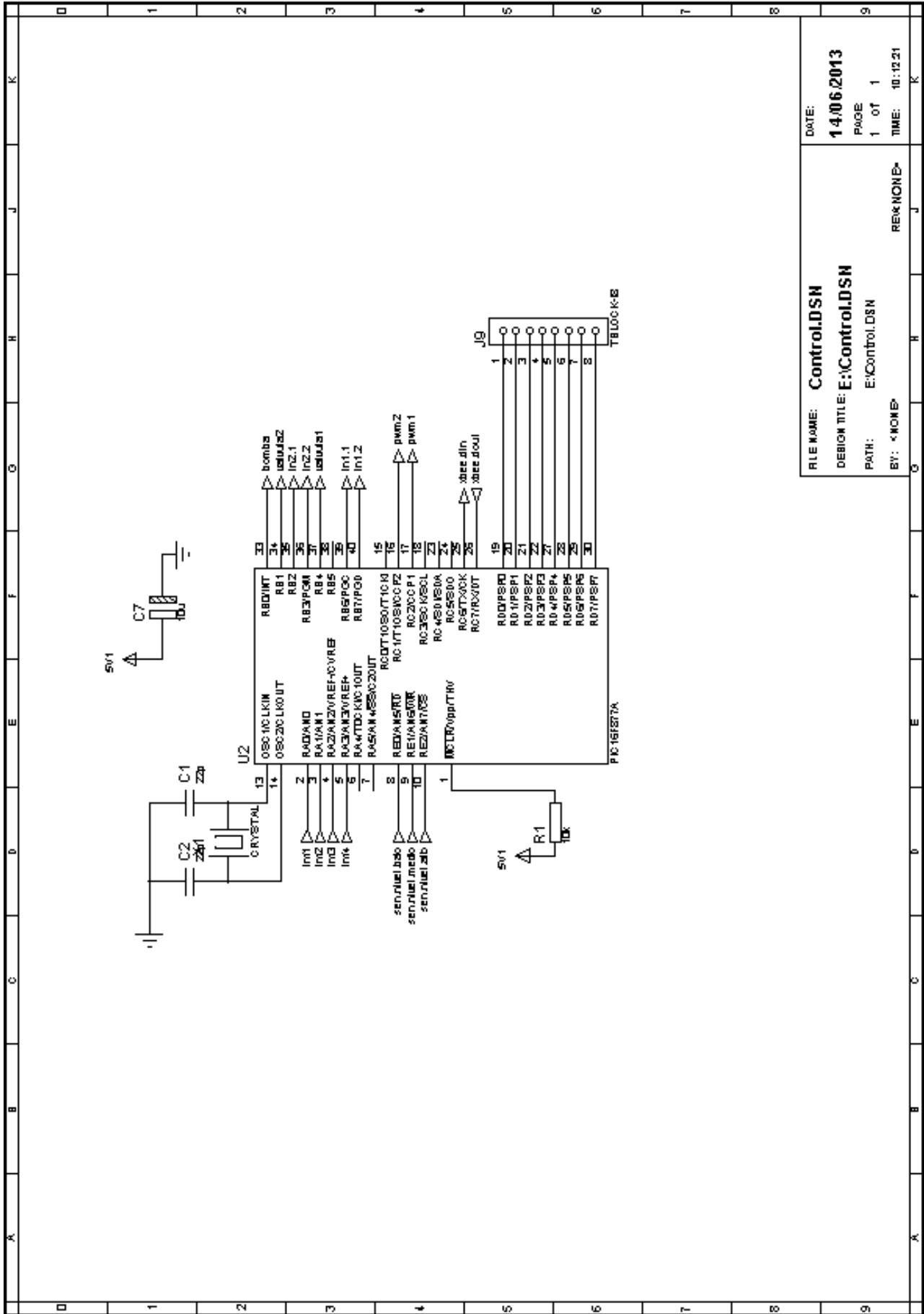
DIAGRAMAS ELECTRÓNICOS





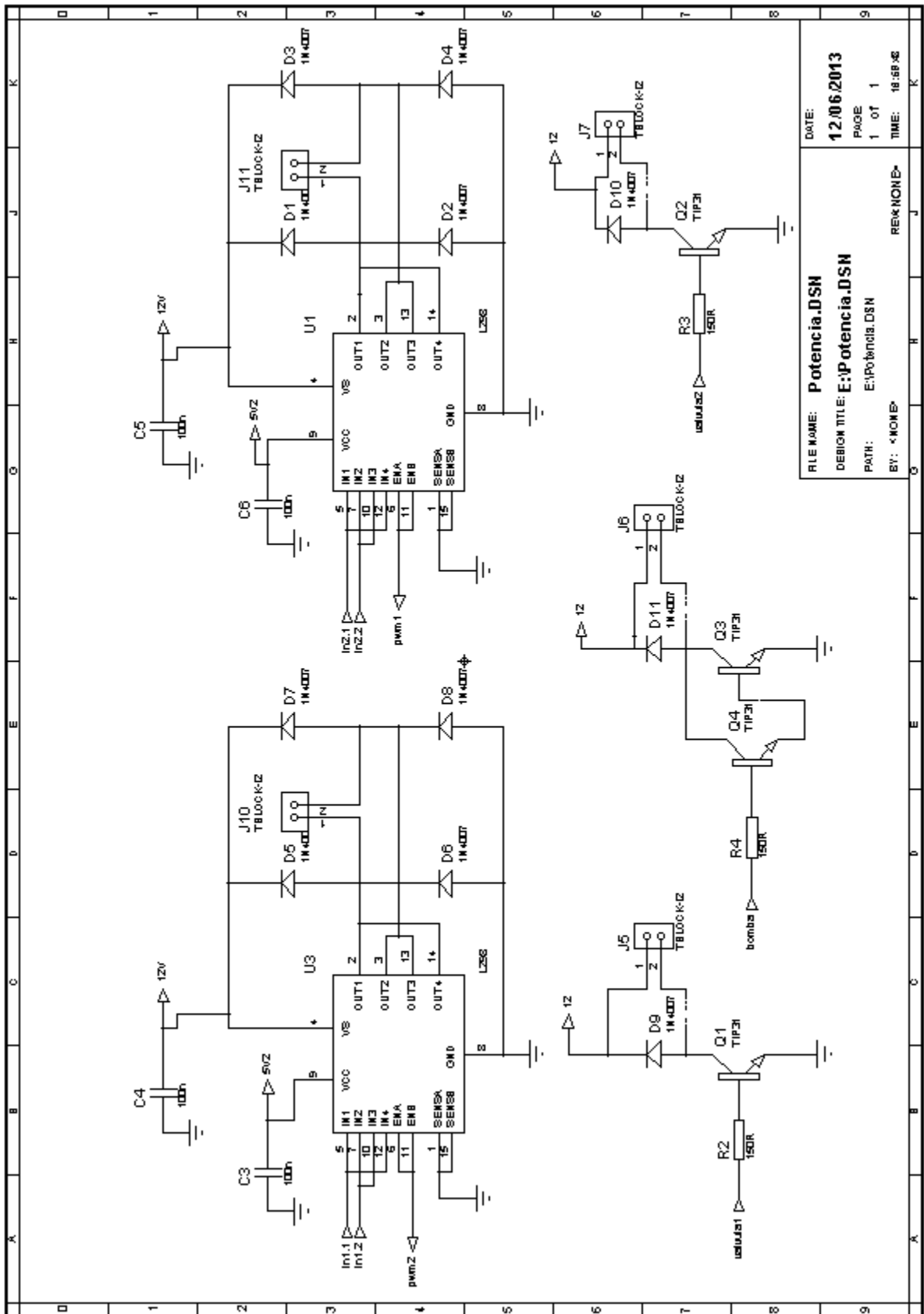
FILE NAME: Instrumentacion.DSN
 DATE: 14/06/2013
 DESIGN TITLE: E:Instrumentacion.DSN
 PAGE: 1 OF 1
 PATH: E:\Instrumentacion.DSN
 BY: <NONE> REV: NONE
 TIME: 10:17:38





FILE NAME:	Control.DSN
DESIGN TITLE:	E:\Control.DSN
PATH:	E:\Control.DSN
BY:	*NONE*
DATE:	14/06/2013
PAGE:	1 OF 1
TIME:	10:12:21

REV: NONE



FILE NAME:	Potencia.DSN
DESIGN TITLE:	E:Potencia.DSN
PATH:	E:\Potencia.DSN
BY:	<NONE>
DATE:	12/06/2013
PAGE	1
OF	1
TIME:	18:58:48

ÍNDICE DE FIGURAS

Figura 1.01. Químico de toxicología IV.	9
Figura 1.02. Químico categoría IV disuelto en agua.	10
Figura 1.03. Bomba distribuidora.	11
Figura 1.04. Manguera de transporte.	11
Figura 1.05. Aguilón de aspersión.	12
Figura 1.06. Pitón del aguilón.	13
Figura 1.07. Altura de una planta de rosas adulta.	13
Figura 1.08. Separación entre camas.	14
Figura 1.09. Longitud de camas de rosas.	15
Figura 1.10. Fumigación manual de rosas.	16
Figura 1.11. Área física para las pruebas del proyecto en metros.	16
Figura 2.01. Arreglo diferencial de 3 y 4 ruedas (Ruiz del Solar, Salazar, 2011).	20
Figura 2.02. Arreglo síncrono de 3 ruedas (Ruiz del Solar, Salazar, 2011).	21
Figura 2.03. Arreglo Triciclo (Ruiz del Solar, Salazar, 2011).	22
Figura 2.04. Arreglo Carro (Ruiz del Solar, Salazar, 2011).	23
Figura 2.05. Arreglo Omnidireccional (Ruiz del Solar, Salazar, 2011).	23
Figura 2.06. Ruedas de Tracción.	28
Figura 2.07. Plano Ruedas de Tracción (mm).	29
Figura 2.08. Garrucha de Dirección.	29
Figura 2.09. Plano Garrucha de Dirección (mm).	30
Figura 2.10. Acople motor-rueda.	30
Figura 2.11. Plano Acople motor-rueda (mm).	31
Figura 2.12. Abrazadera para motor.	31
Figura 2.13. Plano Abrazadera para motor (mm).	32
Figura 2.14. Ángulos Sujetadores de Baterías.	32
Figura 2.14. Plataforma.	41
Figura 2.15. Plano Ángulos Sujetadores de Baterías (mm).	33
Figura 2.16. Soporte de Aspersores.	33
Figura 2.17. Plano Soporte de Aspersores.	34
Figura 2.18. Plataforma.	34
Figura 2.19. Plano Plataforma (mm).	35
Figura 2.20. Ángulos de soporte.	35
Figura 2.21. Plano Ángulos de soporte (mm).	36
Figura 2.22. Caja Aislante.	36
Figura 2.22. Plano Caja Aislante (mm).	37
Figura 2.23. Representación Cinemática Arreglo Diferencial (Ruiz del Solar, Salazar, 2011).	38
Figura 3.01. Conexión e cascada del Puente H.	46
Figura 3.02. Circuito Electrónico de Activación.	50
Figura 3.03. Circuito Electrónico de Activación Electroválvula.	53
Figura 3.04. Tipos de Redes con Módulos Xbee (Archundia, 2004).	61
Figura 3.05. Diagrama de Bloques de Control.	62
Figura 3.06. Forma de Medición Sensor Sharp.	65
Figura 3.07. Conexión sensores de Nivel.	68
Figura 3.08. Distribucion de Voltajes.	69
Figura 4.01. Diagrama de Flujo Principal.	74
Figura 4.02. Diagrama de Flujo Modo Manual.	77
Figura 4.03. Método Nivel.	78
Figura 4.04. Método Adelante.	79
Figura 4.05. Método Alinea.	80

Figura 4.06. Método Atrás.....	81
Figura 4.07. Método Izquierda.....	82
Figura 4.08. Método Derecha.....	83
Figura 4.09. Modo Automático.....	86
Figura 4.10. Distribución de Herramientas Netbeans.....	92
Figura 4.11. HMI Completa.....	92
Figura 4.12. Panel de Títulos.....	93
Figura 4.13. Panel Esquema de Trayectoria.....	94
Figura 4.14. Panel MI-T-NATOR.....	94
Figura 4.15. Panel Selección de Modo.....	95
Figura 4.16. Panel Control Modo Manual.....	96
Figura 4.17. Panel Estado de Robot.....	96
Figura 4.18. Barra de Archivo.....	97
Figura 4.19. Ventana de Reportes.....	98
Figura 5.01. Láminas Base.....	102
Figura 5.02. Soldadura Angulo Base.....	103
Figura 5.03. Vértices Sujetadores Contenedor.....	104
Figura 5.04. Abrazaderas de motor.....	104
Figura 5.05. Acople Llanta Motor.....	105
Figura 5.06. Garrucha de Dirección.....	106
Figura 5.07. Ángulos Sujetadores.....	107
Figura 5.08. Soportes de Aspersores.....	107
Figura 5.09. Caja de control.....	108
Figura 5.10. Bomba de Succión Sumergible.....	109
Figura 5.11. Sistema de Distribución de Fungicida.....	110
Figura 5.12. Conexión Electroválvula.....	110
Figura 5.13. Conexión Aspersor Manguera.....	111
Figura 5.14. Diseño Placa de Control.....	112
Figura 5.15. Placa de Control Implementada.....	113
Figura 5.16. Plano de Construcción Estructuras Metálicas (mm).....	116
Figura 5.17. Representación Estructuras Metálicas.....	117
Figura 5.18. Distribución de las Masetas en el Campo de Pruebas.....	117
Figura 5.19. Plano de Construcción Tabla de Madera (mm).....	118
Figura 5.20. Representación del Campo Total.....	119
Figura 5.21. Representación de la Plataforma en el Campo.....	119
Figura 5.22. Plataforma Sobre el Campo.....	120
Figura 5.23. Láminas de Sensores de Nivel.....	121
Figura 5.24. Termo Fundente aislante.....	121
Figura 5.25. Protectores de Bornes de Batería.....	122
Figura 5.26. Amarra de Sujeción.....	122
Figura 5.27. Espiral Protector Modulo Xbee PRO y Tarjeta de Control.....	123
Figura 5.28. Espiral Protector Válvulas de Distribución.....	123
Figura 5.29. Espiral Protector Sensores Infrarrojos.....	124
Figura 5.30. Estado Final MI-T-NATOR.....	124
Figura 6.01. Falla en la Dirección de la Plataforma sin Carga.....	126
Figura 6.02. Falla en la Dirección de la Plataforma con Carga Máxima.....	127
Figura 6.03. Funcionamiento de Aspersores.....	128
Figura 6.04. Funcionamiento de Junturas de las Válvulas.....	129
Figura 6.05. Funcionamiento T de Distribución.....	129
Figura 6.06. Carpa para limitar la Cantidad de Luz Solar en el Campo.....	130
Figura 6.07. Curva de Transferencia de los Sensores Infrarrojos Sharp (Datasheet-Sharp).....	131
Figura 6.08. Superficie Metálica Sobre la Cual el Haz Infrarrojo Varía el Grado de Reflexión.....	132
Figura 6.09. Papel Adhesivo colocado Sobre la Superficie Metálica.....	133
Figura 6.10. Curva de Transferencia Práctica de los Sensores de Distancia.....	135
Figura 6.11. Primer Nivel Activado.....	136

Figura 6.12. Segundo Nivel Activado.....	137
Figura 6.13. Tercer Nivel Activado.....	137
Figura 6.14. Pantalla Principal X-CTU.....	138
Figura 6.15. Distancia Máxima de Separación en Módulos Xbee.....	139
Figura 6.16. Módulos Xbee Serie 1 separados por una Pared.....	139
Figura 6.17. Separación de la Cámara Inalámbrica y el Módulo Xbee Serie 1.....	140
Figura 6.18. Prueba de Envío y Recepción de Datos Desde y Hacia el Controlador.....	141
Figura 6.19. Prueba de Activación de Dispositivos Eléctricos.....	141
Figura 6.20. Selección de Modo Automático.....	142
Figura 6.21. Selección de Modo Manual.....	143
Figura 6.22. Reportes de Base de Datos.....	144
Figura 6.23. Numeración Cama 1.....	145
Figura 6.24. Numeración Cama 2.....	146
Figura 6.25. Numeración Cama 3.....	146
Figura 6.26. Selección del Modo Manual.....	147
Figura 6.27. Acción Adelante con La Flecha Arriba.....	147
Figura 6.28. Acción Parar con La Flecha Abajo.....	148
Figura 6.29. Acción Encender Aspersor Derecho con la Letra Q.....	148
Figura 6.30. Acción Giro a la Izquierda con la Flecha Izquierda.....	149
Figura 6.31. Desplazamiento Hacia Adelante entre Cama 1 y Cama 2.....	150
Figura 6.32. Ingreso a Cama 2.....	150
Figura 6.33. Ubicación de la Plataforma en la Mitad de la Cama 2.....	151
Figura 6.34. Activación de los Dos Aspersores con la Letra W.....	151
Figura 6.35. Giro a la Derecha con la Flecha Derecha.....	152
Figura 6.36. Desplazamiento Hacia Adelante entre Cama 2 y Cama 3.....	152
Figura 6.37. Ingreso a Cama 3.....	153
Figura 6.38. Activación Aspersor Derecho con la Letra E.....	153
Figura 6.39. Fin de Modo Manual.....	154
Figura 6.40. Fumigación Automática Cama 1.....	155
Figura 6.41. Final de Cama 1.....	156
Figura 6.42. Empieza el Giro.....	156
Figura 6.43. Primer Giro a la Izquierda.....	157
Figura 6.44. Detección de las Plantas en el Giro Izquierdo 1.....	157
Figura 6.45. Corrección de Dirección Hacia Giro 2.....	158
Figura 6.46. Corrección de Dirección Hacia Giro 2.....	158
Figura 6.47. Inicio del Segundo Giro a la Izquierda.....	159
Figura 6.48. Fin del Segundo Giro 2 a la Izquierda.....	159
Figura 6.49. Ingreso a la Cama 2.....	160
Figura 6.50. Fumigación Automática Cama 2.....	160
Figura 6.51. Final de Cama 2.....	161
Figura 6.52. Inicio del Primer Giro a la Derecha.....	162
Figura 6.53. Deteccion de Plantas en el Giro Derecho 1.....	163
Figura 6.54. Corrección de Dirección Hacia el Giro 2.....	163
Figura 6.55. Corrección de Dirección Hacia el Giro 2.....	163
Figura 6.56. Inicio del Giro 2 a la Derecha.....	164
Figura 6.57. Fin del Giro 2 a la Derecha.....	165
Figura 6.58. Fumigación Automática Cama 3.....	165
Figura 6.59. Fin de Modo Automático.....	166
Figura 6.60. Esquema de Trayectoria Dividida en Sectores.....	167
Figura 6.61. Trayectorias del Sector 2.....	168
Figura 6.62. Trayectorias del Sector 4.....	169

ÍNDICE DE TABLAS

Tabla 1.01. Generaciones Robóticas (Jorge Valverde Rebaza-2009.	5
Tabla 3.01. Descripción de Motores de CC.	44
Tabla 3.02. Descripción de Tipos de Módulos Xbee.	56
Tabla 3.03. Características del Módulo Xbee seleccionado (Archundia, 2004	60
Tabla 4.01. Distribución de Puertos.	72
Tabla 4.02. Atributos de la Base de Datos.	99
Tabla 6.01. Valores de Medición de Distancia de los Sensores en la Mañana y en la Tarde.	131
Tabla 6.02. Valores Reales de Distancia Medidos en la Mañana.	133
Tabla 6.03. Teclas de Acción.	144
Tabla 6.04. Resultados Sector 1.	167
Tabla 6.05. Resultados Sector 2.	168
Tabla 6.06. Comparación de Valores PWM Sector 2.	168
Tabla 6.07. Resultados Sector 3.	169
Tabla 6.08. Resultados Sector 4.	170
Tabla 6.09. Comparación de Valores PWM Sector 4.	170
Tabla 6.10. Resultados Sector 5.	171
Tabla 6.11. Cualificación de Pruebas por Sector.	171

GLOSARIO

A

Abrazadera:

Una pieza de ferretería que se usa para *abrazar* un conducto en fontanería.

AC (abreviada CA en español y AC en inglés, *dealternating current*):

Es la corriente eléctrica en la que la magnitud y el sentido varían cíclicamente. La forma de oscilación de la corriente alterna más comúnmente utilizada es la de una oscilación sinusoidal (figura 1), puesto que se consigue una transmisión más eficiente de la energía. Sin embargo, en ciertas aplicaciones se utilizan otras formas de oscilación periódicas, tales como la triangular o la cuadrada.

Ácaros:

Son una subclase de arácnidos, aunque durante mucho tiempo fueron considerados un orden. Existen casi 50.000 especies descritas, y se estima que existen entre 100.000 y 500.000 especies que todavía no han sido halladas. La mayoría de los ácaros son diminutos y alcanzan unos pocos milímetros de longitud.

API:

API es la abreviatura de **A**plication **P**rogramming **I**nterface. Un API no es más que una serie de servicios o funciones que el Sistema Operativo ofrece al programador, como por ejemplo, imprimir un carácter en pantalla, leer el teclado, escribir en un fichero de disco, etc. Visto desde la perspectiva del código máquina, el API aparece como una serie de llamadas (en otros sistemas operativos se hace mediante saltos a supervisor; en OS/2 se implementan como *Far Calls*), mientras que si lo vemos desde la de un lenguaje de alto nivel, el API aparece como un conjunto de procedimientos y funciones.

GLOSARIO

Aspersor:

Un aspersor es un dispositivo mecánico que en la mayoría de los casos transforma un flujo líquido presurizado y lo transforma en rocío, asperjándolo para fines de riego.

AT:

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal modem.

B

Bomba hidráulica:

Es una máquina generadora que transforma la energía (generalmente energía mecánica) con la que es accionada en energía hidráulica del fluido incompresible que mueve.

C

CCI (Centro de curvatura instantáneo):

Un robot móvil puede contar con varias ruedas, y existe un punto alrededor del cual cada rueda sigue una trayectoria circular cuyo punto se llama Centro de Curvatura Instantáneo (CCI) en el caso de una trayectoria recta el CCI está en infinito.

Chipset (circuito integrado auxiliar):

Es el conjunto de circuitos integrados diseñados con base a la arquitectura de un procesador(en algunos casos, diseñados como parte integral de esa arquitectura), permitiendo que ese tipo de procesadores funcionen en una placa base. Sirven de puente de comunicación con el resto de componentes de la placa, como son la memoria, las tarjetas de expansión, los puertos USB, ratón, teclado, etc.

GLOSARIO**D****Datasheet:**

Es un documento que resume el funcionamiento y otras características de un componente (por ejemplo, un componente electrónico) o subsistema (por ejemplo, una fuente de alimentación) con el suficiente detalle para ser utilizado por un ingeniero de diseño y diseñar el componente en un sistema.

DC (corriente continua): Se refiere al flujo continuo de carga eléctrica a través de un conductor entre dos puntos de distinto potencial, que no cambia de sentido con el tiempo.

G**Garrucha:**

Mecanismo con una rueda que sirve para mover o levantar pesos.

J**JAVA:**

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como *WORA*, o "*write once, run anywhere*"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

Jdk(Java development kit):

Es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red. En la unidad de red se pueden tener las herramientas distribuidas en varias computadoras y trabajar como una sola aplicación.

I

Infrarroja:

O radiación **IR** es un tipo de radiación electromagnética y térmica, de mayor longitud de onda que la luz visible, pero menor que la de las microondas. Consecuentemente, tiene menor frecuencia que la luz visible y mayor que las microondas. Su rango de longitudes de onda va desde unos 0,7 hasta los 1000 micrómetros.

M

Min OPD:

Representa el Mínimo Diferencial de Presión de Operación. Todas las válvulas operadas por piloto (como las series 200 y 240 de Emerson) requieren una pequeña cantidad de diferencial de presión para permitir que el pistón o el diafragma se levanten del puerto principal.

Típicamente, se requiere un diferencial de 2 a 5 psi para lograr esto. Si el diferencial de presión es menor que el Min OPD, entonces la válvula no operará cuando se active o fallará en permanecer abierta. Si una válvula es considerablemente de mayor tamaño para la aplicación, puede sufrir esta una afectación, puesto que, la caída de presión a través de la válvula con bajos índices de flujo puede estar por debajo del Min OPD

MOPD:

La clasificación MOPD para la válvula es el máximo diferencial de presión contra el cual la válvula operará confiablemente. Las válvulas de solenoide están clasificadas en términos del Máximo Diferencial de Presión de Operación (MOPD Maximum Operating Pressure Differential por sus siglas en Inglés) en contra del cual la válvula abrirá. Si el diferencial de presión es más alto que el MOPD al que se clasificó esa válvula, la válvula no operará.

GLOSARIO

Microcontrolador (abreviado μ C, UC o MCU):

Es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.

N

Netbeans:

Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. Netbeans IDE es un producto libre y gratuito sin restricciones de uso.

P

PCB (del inglés *printed circuit board*):

Es una superficie constituida por *caminos* o *pistas* de material conductor laminadas sobre una base no conductora. El circuito impreso se utiliza para conectar eléctricamente - a través de los caminos conductores, y sostener mecánicamente - por medio de la base, un conjunto de componentes electrónicos. Los caminos son generalmente de cobre mientras que la base se fabrica de resinas de fibra de vidrio reforzada (la más conocida es la FR4), cerámica, plástico, teflón o polímeros como la baquelita.

Peer to peer:

Una red *peer-to-peer* es una red de computadoras en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red. Las redes P2P permiten el intercambio directo de información, en cualquier formato, entre los ordenadores interconectados.

GLOSARIO

PICC:

Es un compilador que nos permite escribir los programas en lenguaje C en vez de assembler, con lo que se logra un menor tiempo de desarrollo, y mucha facilidad en la programación.

Pin-out:

Es un término anglosajón que, en traducción libre, significa patillaje, o más correctamente asignación de patillaje. Es usado en electrónica para determinar la función de cada pin en un circuito integrado, o bien en un dispositivo electrónico discreto. En informática, para describir cómo un conector es cableado. Cada patica del conector tiene un propósito que se describe brevemente en el pin-out.

PH:

Es una medida de acidez o alcalinidad de una disolución. El pH indica la concentración de iones hidronio $[H_3O^+]$ presentes en determinadas sustancias.

PSI (libra-fuerza por pulgada cuadrada):

Más conocida como psi (del inglés *pounds per square inch*) es una unidad de presión en el sistema anglosajón de unidades. La escala más común se mide en psi, cuyo cero es la presión ambiente, que equivale a una atmósfera (o sea 15 psi aproximadamente).

Equivalencias:

$$1 \text{ psi} = 6894,75729 \text{ pascales} = 6.8948 \text{ kPa (kilopascales)}$$

$$1 \text{ psi} = 0,0698 \text{ bares}$$

$$1 \text{ pascal} = 0,000145 \text{ psi}$$

$$14,7 \text{ psi} = 1 \text{ atmósfera}$$

$$14,7 \text{ psi} = 1,013 \text{ Bar}$$

$$1 \text{ kp/cm}^2 = 14,2065 \text{ psi}$$

$$1 \text{ psi} = 0.070307 \text{ kg/cm}^2$$

S

Sensor:

Es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas.

Subrutina o subprograma:

En computación (también llamada procedimiento, función o rutina), se presenta como un sub-algoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica

SWP (Presion segura de trabajo):

Presión normal de trabajo de una válvula.

V

Vástago:

Varilla, barra que transmite el movimiento a algún mecanismo.

Velocidad angular:

Es una medida de la velocidad de rotación. Se define como el ángulo girado por una unidad de tiempo y se designa mediante la letra griega ω . Su unidad en el Sistema Internacional es el radián por segundo (rad/s).

VI:

Velocidad lineal rueda izquierda

GLOSARIO**Vr:**

Velocidad rueda derecha

X**Xbee:**

Es el nombre comercial de Digi International para una familia de módulos de radio frecuencia. Los primeros XBee se introdujeron bajo la marca MaxStream en el año 2005 y se basaron en el estándar 802.15.4 -2003 diseñado para comunicaciones punto a punto y en estrella con velocidades de transmisión de 250 kbit / s.

FECHA DE ENTREGA

El proyecto de grado “DISEÑO E IMPLEMENTACIÓN DE UN ROBOT MÓVIL AUTÓNOMO Y TELEOPERADO PARA LABORES AGRÍCOLAS. FASE 1: FUMIGACIÓN PARA PLANTACIONES DE ROSAS” fue entregado al departamento de Eléctrica y Electrónica y reposa en la Escuela Politécnica del Ejército.

Sangolquí, Junio 2013.

Elaborado por:

Iván Andrés León Vásquez.

Adrián Jonatan Vásquez Tapia.

Autoridades:

Ing. Luis Orozco.
Director de la Carrera
Ingeniería Electrónica, Automatización y Control