

**DESARROLLO DE UN PROTOTIPO DE ROBOT MEDIDOR DE PH EN SUELOS
LLANOS**



**DESARROLLO DE UN PROTOTIPO DE ROBOT MEDIDOR DE PH EN SUELOS
LLANOS**

DANIEL EDUARDO ALDANA HERNANDEZ.

**UNIVERSIDAD PEDAGÓGICA NACIONAL
FACULTAD DE CIENCIA Y TECNOLOGÍA
DEPARTAMENTO DE TECNOLOGÍA
LICENCIATURA EN ELECTRÓNICA
BOGOTÁ D.C.**

- 2016

**DESARROLLO DE UN PROTOTIPO DE ROBOT MEDIDOR DE PH EN SUELOS
LLANOS**

**DANIEL EDUARDO ALDANA HERNANDEZ
TRABAJO DE GRADO**

**DIRECTOR
AUGUSTO GUARÍN RODRÍGUEZ
INGENIERO ELECTRÓNICO**

**UNIVERSIDAD PEDAGÓGICA NACIONAL
FACULTAD DE CIENCIA Y TECNOLOGÍA
DEPARTAMENTO DE TECNOLOGÍA
LICENCIATURA EN ELECTRÓNICA
BOGOTÁ D.C.**

- 2016

Nota de aceptación:

Firma del director del trabajo de grado

Augusto Guarín Rodríguez

Firma del Jurado 1

Firma del Jurado 2

DEDICATORIA

Este trabajo de grado se lo dedico a mi Dios, quién supo guiarme por el buen camino, darme fuerzas para seguir adelante y no desmayar en los problemas que se presentaban, enseñándome a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento. A mi Madre quien siempre me apoyo con sus consejos, comprensión, amor, ayuda en los momentos difíciles, y por ayudarme con los recursos necesarios para estudiar. Y me han dado todo lo que soy como persona, mis valores, mis principios, mi carácter, mi empeño, mi perseverancia, mi coraje para conseguir mis objetivos. A mis abuelos maternos Rosa y Adriano por estar siempre presentes, acompañándome para poderme realizar.

AGRADECIMIENTOS

Agradezco a mi familia materna quienes dieron mucho interés en que pudiera cumplir mis sueños y me ayudaron con buenos consejos. Especialmente a mis tíos (Diana, Wilman y Patricia) que de una u otra forma intervinieron en mis estudios universitarios y a mi madre que siempre ha dado todo por mí (gracias Madre hermosa)

RESUMEN ANALÍTICO EN EDUCACIÓN - RAE

| 1. Información General | |
|-------------------------------|--|
| Tipo de documento | Trabajo de grado. |
| Acceso al documento | Universidad Pedagógica Nacional. Biblioteca Central |
| Título del documento | DESARROLLO DE UN PROTOTIPO DE ROBOT MEDIDOR DE PH EN SUELOS LLANOS |
| Autor(es) | Aldana Hernández, Daniel Eduardo |
| Director | Guarín Rodríguez, Augusto |
| Publicación | Bogotá. Universidad Pedagógica Nacional, 2016. 76 p. |
| Unidad Patrocinante | Universidad Pedagógica Nacional. |
| Palabras Claves | CONTROL, JAVA, ROBOT MÓVIL, PROTOTIPO, PH. |

| 2. Descripción |
|--|
| <p>: Este trabajo de grado presenta, el desarrollo de un prototipo de robot medidor de ph en suelos, basado en una estrategia de sistema control básico, teniendo como objeto su desplazamiento horizontal. Se realizó mediante la ayuda de la tarjeta ARDUINO MEGA 2560, principalmente por su capacidad de almacenamiento de información y los puertos análogos y digitales que contiene. el uso de esta tarjeta nos permite utilizar los datos almacenados con una interfaz gráfica en la plataforma NetBeans IDE 7.2.1 utilizando el lenguaje orientado a objetos, donde se aprecia la toma de los datos graficándolos y permitiendo interactuar al usuario con los datos tomados. Los datos se obtienen mediante un pH metro análogo.</p> |

| 3. Fuentes |
|--|
| <p>http://aplicacionesdeloselectro2.blogspot.com.co(Sequeda M & Orozco D 2006) https://www.arduino.cc/en/Main/ArduinoBoardMega2560</p> |

| 4. Contenidos |
|--|
| <p>Identificación del problema, Objetivos del proyecto Marco de referencia: Robot terrestre Robot móviles Tipos de ruedas Tipos de locomoción Estructura general de un robot móvil Estructura mecánica Estructura física Ruedas Sensores y actuadores</p> |

Sistema de control inteligente Comunicación de datos USB 2.0
Aspectos metodológicos: Diseño y construcción Conexión del ph metro
Trayectorias Control: Diseño de control software en Arduino
Prototipo Real
Desarrollo de interfaz en java con software netbeans

5. Metodología

La metodología que se utilizará para el desarrollo del prototipo de robot medidor de pH será basada por ciclos de diseño, donde estarán compuestos por diferentes fases los cuales nos darán una visión del proyecto a medida que se va avanzando.

6. Conclusiones

Podemos concluir que de acuerdo a la dureza del suelo donde se toma la muestra de ph facilita o dificulta la introducción del pH metro al suelo
El tamaño, forma y material de las ruedas nos permite un mejor agarre y posicionamiento en el terreno
La utilización de los servos de giro continuo nos dificulto un poco el posicionamiento del prototipo porque no nos daba un giro en una sola dirección (modificar el potenciómetro de los servos)
Para que el prototipo girara 180 grados para que iniciara cada metro nos tocó medir el tiempo y la velocidad con el cual giraba se realizó con motores dc de tv
El software arduino nos dio facilidad de programación para ejecutar la tarea del prototipo
La librería hilteck_panama nos ayudó en la comunicación entre la tarjeta arduino mega y el software netbeans para realizar la interface grafica

| | |
|-----------------------|---------------------------------|
| Elaborado por: | Daniel Eduardo Aldana Hernández |
| Revisado por: | Augusto Guarín Rodríguez |

| | | | |
|--|----|----|------|
| Fecha de elaboración del Resumen: | 03 | 11 | 2016 |
|--|----|----|------|

ÍNDICE GENERAL

| | |
|--|-----------|
| 1.1. Identificación del problema | 13 |
| 1. Planteamiento del problema | 13 |
| 1.2. Justificación | 13 |
| 1.3. Delimitación | 15 |
| 1.4. Objetivos del proyecto | 16 |
| 1.4.1. Objetivo general | 16 |
| 2. Marco de referencia | 18 |
| 2.1. Robot terrestre | 18 |
| 2.2. Robot móvil | 18 |
| 2.2.1. Tipos de ruedas | 18 |
| 2.2.1.1. Tipos de locomoción | 19 |
| 1.4.2. Objetivos específicos | 16 |
| 1.5. Antecedentes | 16 |
| 2.2.2. Robots de patas | 20 |
| 2.3. Estructura general de un robot móvil | 21 |
| 2.4. Estructura mecánica | 21 |
| 2.4.1. Estructura física | 21 |
| 2.5.1. Sensores | 22 |
| 2.5.1.1. Electrodo de referencia | 23 |
| 2.5.1.2. Electrodo indicadores metálicos | 24 |
| 2.5.1.3. Electrodo indicadores de membrana | 24 |
| 2.5.1.4. Electrodo de vidrio para la medida del pH | 25 |
| 2.5.2. Actuadores | 27 |
| 2.5.2.1. Motores Dc | 27 |
| 2.5.2.2. Servomotores | 28 |
| 2.6. Sistema de control inteligente | 29 |
| 2.6.1. Tarjeta Arduino mega 2560 | 31 |
| 2.6.2. Software Arduino | 32 |
| 2.7. Comunicación de datos USB 2.0 | 33 |
| 2.7.1. Topología | 34 |
| 2.7.2. Funcionamiento | 35 |
| 2.8. Interfaz | 35 |
| 2.4.2. Ruedas | 21 |
| 3.1. Diseño y construcción | 35 |
| 2.5. Sensores y actuadores | 22 |
| 3. Aspectos metodológicos | 35 |
| 3.1.1. Diseño | 35 |
| 3.1.2. Estructura | 37 |
| 3.1.2.1. Soporte de los servos y ruedas traseras | 38 |
| 3.1.2.2. Transmisión delantera | 38 |
| 3.1.2.3. Soporte del sensor medidor de pH | 39 |
| 3.1.2.4. Estructura total del prototipo | 40 |

| | |
|--|-----------|
| 3.2. Materiales | 41 |
| 3.2.1.Cantidad de materiales | 41 |
| 3.3. Hardware | 42 |
| 3.3.1.Conexión servo rueda izquierda | 42 |
| 3.3.2.Conexión servo rueda derecha | 42 |
| 3.3.3.Conexión servo del sensor de pH (pH metro) | 42 |
| 3.3.4.Conexión motora dc para giro tracción delantera | 42 |
| 3.3.5.Funcionamiento del integrado l293 | 43 |
| 3.3.6.Conexión del pH metro | 44 |
| 3.4. Trayectorias | 44 |
| 3.5. Control | 45 |
| 3.5.1. Diagrama de flujo | 45 |
| 3.5.2.Diseño de control software en Arduino | 46 |
| 3.6. Prototipo Real | 51 |
| 3.7. Desarrollo de interfaz en java con software NetBeans..... | 54 |
| 3.7.1.Pasos del desarrollo..... | 54 |
| Conclusiones | 70 |
| Proyecciones futuras | 71 |
| Bibliografía | 72 |
| Anexos | 73 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Tipo de locomocion Diferencial | 19 |
| Figura 2. Tipo de locomocion sincrona | 19 |
| Figura 3. Tipo de locomocion triciclo | 20 |
| Figura 4. Tipo de locomocion Ackerman..... | 20 |
| Figura 5. Tipo de ruedas. | 22 |
| Figura 6. Sistema típico de electrodos para la medición potencio métrica del pH | 25 |
| Figura 7. Electrodo potenciométrica del pH | 26 |
| Figura 8. Óhmetro. | 26 |
| Figura 9. Instrumento de medición de bolsillo de pH suelo | 27 |
| Figura 10. Motor DC. | 28 |
| Figura 11. servomotor | 29 |
| Figura 12. Sistema de control. | 30 |
| Figura 13. Lazo abierto. | 30 |
| Figura 14. Lazo cerrado | 30 |
| Figura 15. Software Arduino. | 33 |
| Figura 16. Ventana Arduino. | 33 |
| Figura 17. Velocidad de transferencia | 34 |
| Figura 18 simulación de prototipo configuración de tracción Ackerman | 36 |
| Figura 19. Diseño del prototipo vista superior. | 36 |
| Figura 20. Diseño del prototipo vista superior carcasa del prototipo. | 37 |
| Figura 21. Diseño del prototipo vista lateral carcasa del prototipo. | 37 |
| Figura 22. Diseño del prototipo vistas soporte ruedas traseras | 38 |
| Figura 23. Diseño del prototipo vistas soporte ruedas delanteras. | 39 |
| Figura 24. Diseño del prototipo vistas soporte servomotor. | 39 |
| Figura 25. Estructura final del prototipo vistas | 40 |
| Figura 26. Diseño del prototipo vistas 2. | 40 |
| Figura 27. Prototipo final. | 41 |
| Figura 28 datasheet L293D | 43 |
| Figura 29. Diseño del prototipo vista superior carcasa del prototipo. | 51 |
| Figura 30. Diseño del prototipo vista trasera | 52 |

| | |
|--|----|
| Figura 31. Diseño del prototipo vista lateral | 52 |
| Figura 32. Diseño del prototipo vista inferior | 53 |
| Figura 33 ventana NetBeans creación del paquete | 54 |
| Figura 34 ventana NetBeans creación del JFrame | 54 |
| Figura 35 librería panamahiteck | 55 |
| Figura 36 importación de librería panamahiteck | 55 |
| Figura 37 ventana JFrame se empieza a realizar el código | 55 |
| Figura 38 ventana JFrame se empieza a ejecutar el código | 69 |
| Figura 39 ventana JFrame código ejecutado | 69 |

Resumen: Este trabajo de grado presenta, el desarrollo de un prototipo de robot medidor de pH en suelos, basado en una estrategia de sistema control básico, teniendo como objeto su desplazamiento horizontal. Se realizó mediante la ayuda de la tarjeta ARDUINO MEGA 2560, principalmente por su capacidad de almacenamiento de información y los puertos análogos y digitales que contiene. el uso de esta tarjeta nos permite utilizar los datos almacenados con una interfaz gráfica en la plataforma NetBeans IDE 7.2.1 utilizando el lenguaje orientado a objetos, donde se aprecia la toma de los datos graficándolos y permitiendo interactuar al usuario con los datos tomados. Los datos se obtienen mediante un pH metro análogo.

Palabras claves: Control, sensor de pH, Prototipo, Arduino.

Abstract

This degree work presents the development of a prototype robot meter pH soils, based on a strategy of basic control system , aiming its horizontal displacement. It was performed with the aid of the card Arduino Mega 2560 , mainly for its storage capacity and the digital and analog ports it contains. the use of this card allows us to use the stored data with a graphical interface in the NetBeans IDE 7.2.1 platform using the object-oriented language, where the taking of the data is seen by plotting them and allowing the user to interact with the data collected . Data is obtained by an analogous pH meter .

Key words: Control, PH sensor, Prototype, Arduino.

1. PLANTEAMIENTO DEL PROBLEMA.

1.1 Identificación del problema

En la producción agrícola es importante saber el nivel de pH (potencial de hidrogeno) que tiene el suelo con el fin de observar que tan viable es el terreno para ser cultivado Sin llevar las muestras a un laboratorio.

Por esta razón se pretende diseñar un prototipo de robot, que tome muestras de pH en cada metro cuadrado del predio, para ser trasmitidas a un computador el cual nos mostrará por medio de un software las dimensiones de la finca y señalará los puntos en que tome las muestras con el fin de compararlas con una tabla de referencia entendiendó qué sectores del terreno se deberán mejorar aportándoles fertilizantes que escaseen para que no se pierda ningún tramo de la cosecha.

La idea surge por la necesidad de las fincas en el municipio de Chiquinquirá, vereda Quiche que constantemente tienen perdidas en sus cultivos por no conocer el pH de sus terrenos.

Es fundamental obtener un pH homogéneo en toda la finca permitiendo hacer una buena siembra de cultivos como maíz, arracacha, papa etc. ya que si no tiene un pH balanceado puede afectar el crecimiento vegetal y proceso fisiológico de absorción de los nutrientes desde las raíces. Todas las especies vegetales presentan unos rangos característicos de pH en los que su absorción es idónea.

Fuera de este rango la absorción radicular se ve afectada por la diferencia de los valores de pH deteriorando el sistema radical o presentarse toxicidades debidas a la excesiva absorción de elementos fitotóxicos (aluminio).

1.2 Justificación

La idea de este trabajo de grado es contar con un prototipo de robot medidor de pH para mejorar la calidad de los cultivos sembrados en las fincas de la vereda Quiche. Aprovechando la totalidad de sus terrenos.

Este robot medidor de pH va ser una herramienta que integre elementos electrónicos y software de programación que podrán ser incorporados como maquinaria para fortalecer la agricultura.

El empleo de este robot es con el fin de ayudar a los hacendados y campesinos en el mejoramiento de sus predios dedicados a la agricultura dándoles a conocer la calidad de tierra que tiene en su finca para que puedan hacer los respectivos cambios. Ya que éste determina el nivel de acidez o alcalinidad de la tierra

El resultado de cada muestra tomada por el robot nos indicara el nivel de pH en cada metro cuadrado permitiéndonos mejorar ese tramo de tierra. Con ayuda del software quien nos indicara que sector toca mejorar, de acuerdo al cultivo que pensamos sembrar. Haciendo una comparación con el dato tomado por el robot y el dato estipulado en una tabla de referencia. Esto permite corregir el nivel de pH de la tierra y así mejorando la calidad de los cultivos ya sean para la alimentación de personas o de animales (pastos).

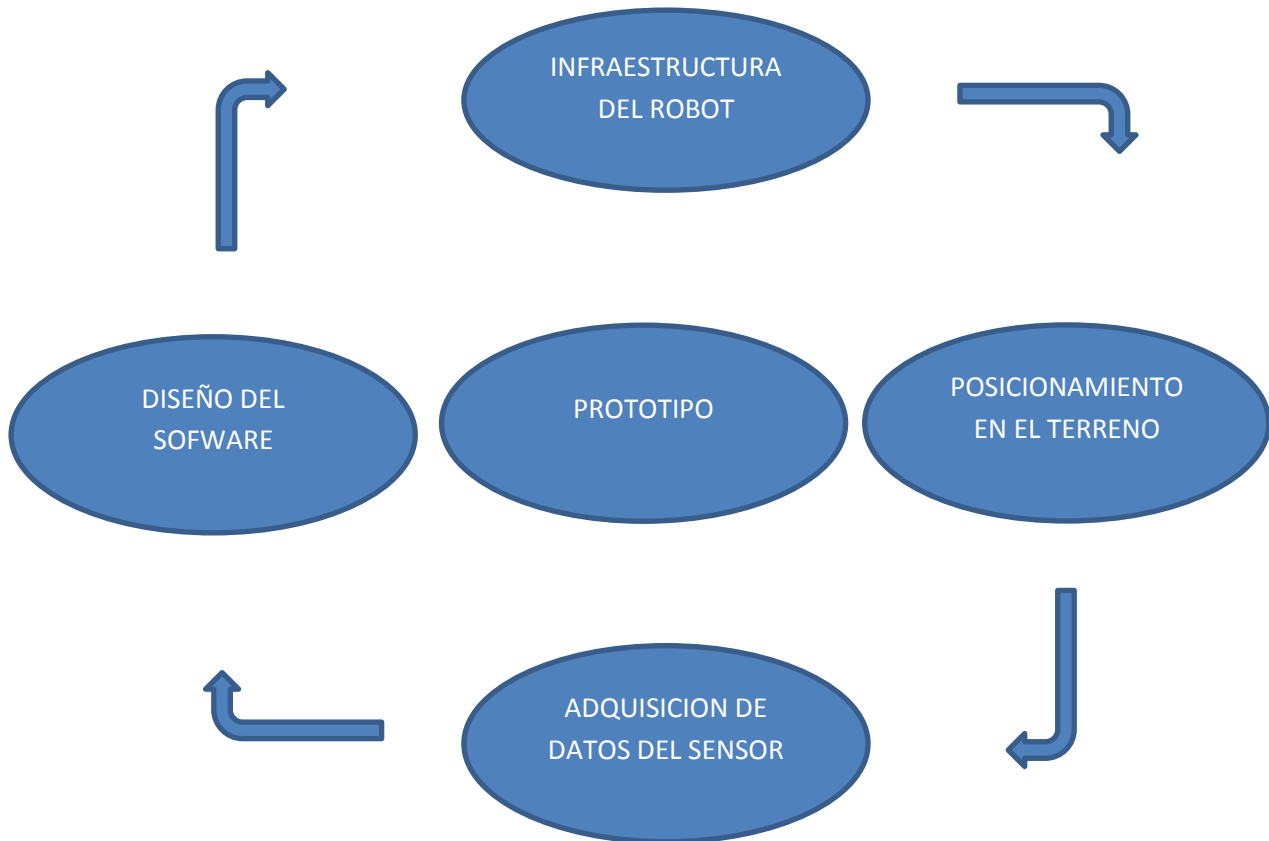
Tabla 1 Niveles de PH apto para cultivar

| Hortícolas | pH óptimo | Frutales | pH óptimo | Extensivos | pH óptimo |
|-------------------|----------------------|-----------------|----------------------|-------------------|----------------------|
| Acelga | 6.0-7.5 | Albaricoque | 6.0-6.8 | Alfalfa | 6.5-7.8 |
| Apio | 6.1-7.4 | Almendro | 6.0-6.8 | Algodón | 5.0-6.2 |
| Berenjena | 5.4-6.0 | Avellano | 6.0-7.0 | Alpiste | 6.0-7.0 |
| Boniato | 5.1-6.0 | Café | 5.0-7.0 | Altramuz | 5.0-7.0 |
| Brócoli | 6.0-7.2 | Castaño | 5.0-6.5 | Arroz | 5.0-6.5 |
| Calabaza | 5.6-6.8 | Encina | 4.8-6.0 | Avena | 5.2-7.1 |
| Cebolla | 6.0-7.2 | Grosellero | 6.0-7.0 | Batatas | 5.3-6.5 |
| Col | 6.0- | Limonero | 6.0-7.5 | Cacahuet | 5.3-6.5 |

| | | | | | |
|--|-----|--|--|---|--|
| | 7.5 | | | e | |
|--|-----|--|--|---|--|

1.3 Delimitación

Diseñar el trabajo por módulos.



INFRAESTRUCTURA DEL ROBOT

Se elaborará un prototipo capaz de desplazarse en el terreno llano con desplazamiento a velocidad constante

POSICIONAMIENTO EN EL TERRENO

Se programará el prototipo de robot con unas dimensiones del terreno a trabajar que son 20 m de fondo por 20 m de ancho en total 400 m cuadrado el cual será una parcela

rectangular con un terreno llano. Se hará el control específico que nos garantice la toma de la muestra cada metro cuadrado

ADQUISICIÓN DE DATOS POR SENSOR

Se tomarán las muestras y el valor en cada punto que se guardarán en un en la memoria eeprom de la tarjeta Arduino mega 2560, para después transmitirlo al computador por medio de USB y los recibirá un software en java

DISEÑO DEL SOFTWARE

Se elaborará un algoritmo en el software NetBeans programando las especificaciones dadas y mostrando los puntos donde se tomó la muestra de pH

1.4 Objetivos

1.4.1 Objetivo general

Construir un prototipo de robot que permita la toma de muestras de pH en suelos. basado en una estrategia de sistema de control básico de posicionamiento en terrenos llanos con una trayectoria definida.

1.4.2 Objetivos específicos

Diseñar y construir una estructura que permita un correcto desplazamiento del prototipo de robot medidor de pH sobre el terreno llano.

Desarrollar una estrategia de sistema de control básico de posicionamiento, que garantice el óptimo funcionamiento del prototipo.

Determinar el tipo de sensores mínimos y necesarios para alimentar la estrategia de sistema de control para el desplazamiento del prototipo.

Determinar e implementar la instrumentación requerida para la medición de pH.

1.5 Antecedentes

- Proyecto realizado por RoboticsDesignSystem (Wiki VEX).

Magbot Modelo 1.

Este proyecto fue la realización del robot VEX escalador magnético que fue diseñado para demostrar cómo un simple robot podría lograr el desafío técnico de subir una superficie metálica. Teniendo en cuenta las numerosas aplicaciones comerciales para este tipo de robot. La refinería de petróleo debe inspeccionar el interior de sus contenedores de almacenamiento. Este Robot tuvo gran éxito comercial con su robot aspiradora Roomba, así como sus numerosos robots de limpieza; robots capaces de limpiar o inspeccionar las superficies metálicas de los tanques de petróleo, edificios, torres de agua, arenas y siendo de gran valor comercial.

- Concurso realizado en la Universidad de Catania; durante la 3^a Conferencia Clawar (Madrid, 2000), por Domenico Longo.

ROBOT SCID.

Este trabajo describe la solución óptima de un robot SCID (deslizamiento de dispositivos escalada de Inspección), desarrollado en el laboratorio de DEES, ha sido diseñada para deslizarse sobre una superficie vertical ferromagnética, utilizando dos electroimanes que se puede desplazar en zonas de difícil acceso para el ser humano, mejorando la supervisión de los procesos industriales.

El sistema es totalmente autónomo, puede ser fácilmente reprogramado con un PC estándar con una interfaz de usuario sencilla y se puede llevar a algunos sensores como una cámara CCD para inspeccionar la superficie, utilizando una conexión RF para transferir datos de imágenes a una consola fija.

- 6^a Conferencia Internacional de Robótica Móvil y Servicio - FSR 2007 42 (2007) Instituto de Robótica y Sistemas Inteligentes (IRIS)

ROBOTS PARA SUPERFICIES FINAS CON OBSTÁCULOS ESPECÍFICOS.

Este trabajo que describe una solución de robot móvil sobre ruedas magnéticas, diseñado para la inspección de las superficies interiores de los tanques de gas estén hechas de láminas de metal fino.

El criterio principal para la optimización de este robot es para el diseño lo más ligero posible, ya que la superficie tiende también a ser muy frágiles.

2. MARCO REFERENCIA

Los robot son máquinas construidas para desarrollar alguna actividad, combinando diversas áreas (mecánica, sistemas y electrónica)

Los robots se clasifican en tres funciones:

- en función de su aplicación (industriales y servicios)
- En función del medio (terrestres, aéreos, acuáticos y híbridos)
- En función del control de movimiento (autónomos, tele operados, robots fijos, robots móviles y otros)

en donde el prototipo a trabajar hace parte de la clasificación en función del medio y de tipo terrestre.

2.1 Robots terrestres

son la solución más simple para navegar en diferentes terrenos suficientemente duros y libres de obstáculos poseen diversas aplicaciones como: exploración minera, exploración planetaria, misión de búsqueda y rescate de personas, limpieza de desechos peligrosos, vigilancia, reconocimiento del terreno.

2.2 Robot móviles

Un robot móvil es una máquina automática que es capaz de trasladarse en cualquier ambiente teniendo la capacidad de moverse en su entorno.

Los robots móviles se clasifican según su sistema de locomoción en categorías: (Robots de ruedas, Robots de patas, Robots de orugas, marinos, articulados, aéreos)

2.2.1 Robots de ruedas

Los robots con ruedas son más sencillos y más fáciles de construir, pueden transportar una carga mayor relativamente. Las ruedas son la solución más apropiada para la mayoría de aplicaciones para la estabilidad del vehículo.

2.2.1.1 Tipos de ruedas

- Rueda motriz: La que proporciona fuerza de tracción al robot
- Rueda directriz: Ruedas de direccionamiento de orientación controlable.
- Ruedas fijas: Sólo giran en torno a su eje sin tracción motriz.
- Ruedas locas o ruedas de castor.
- Ruedas orientables no controladas.

2.2.1.2 Tipos de Locomoción

❖ Diferencial

Este tipo de direccionamiento viene dado por la diferencia de velocidades de las ruedas laterales. La tracción se consigue también con estas mismas ruedas. Dos ruedas montadas en un único eje son independientemente propulsadas y controladas, proporcionando ambas, tracción y direccionamiento.



Figura 1. Locomoción Diferencial; Configuración cinemática del robot móvil.

Fuente: (Azcón, 2003).

❖ Síncrona

Todas las ruedas (usualmente 3) se mueven en forma síncrona para dar vuelta y avanzar, las 3 ruedas están ligadas de forma que siempre apuntan en la misma dirección y para dar vuelta giran las ruedas sobre el eje vertical.



Figura 2. Locomoción Síncrona; Configuración cinemática del robot móvil.

Fuente: (Azcon, 2003).

❖ Triciclo

Dos ruedas fijas que le dan tracción y una rueda para dirección que normalmente no tiene tracción Buena estabilidad y simplicidad mecánica • Facilidad para ir recto • Cinemática más compleja

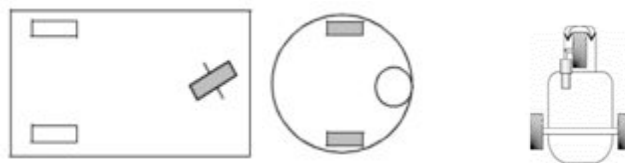


Figura 3. Locomoción Triciclo; Configuración cinemática del robot móvil.

Fuente: (Azcon, 2003).

❖ Ackerman

se basa en un sistema que trabaja con cuatro ruedas donde las dos ruedas traseras se montan de forma paralela en un chasis principal del vehículo, mientras que las ruedas delanteras son de tipo de direccionamiento, y se utilizan para seguir la trayectoria del robot. La rueda delantera interior gira a un ángulo ligeramente superior a la rueda exterior, de forma tal que los ejes de prolongación de las ruedas delanteras y se controlan con el centro instantáneo de rotación.

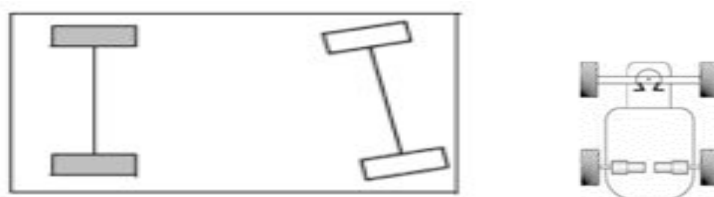


Figura 4. Locomoción Ackerman Configuración cinemática del robot móvil.

Fuente: (Azcon, 2003).

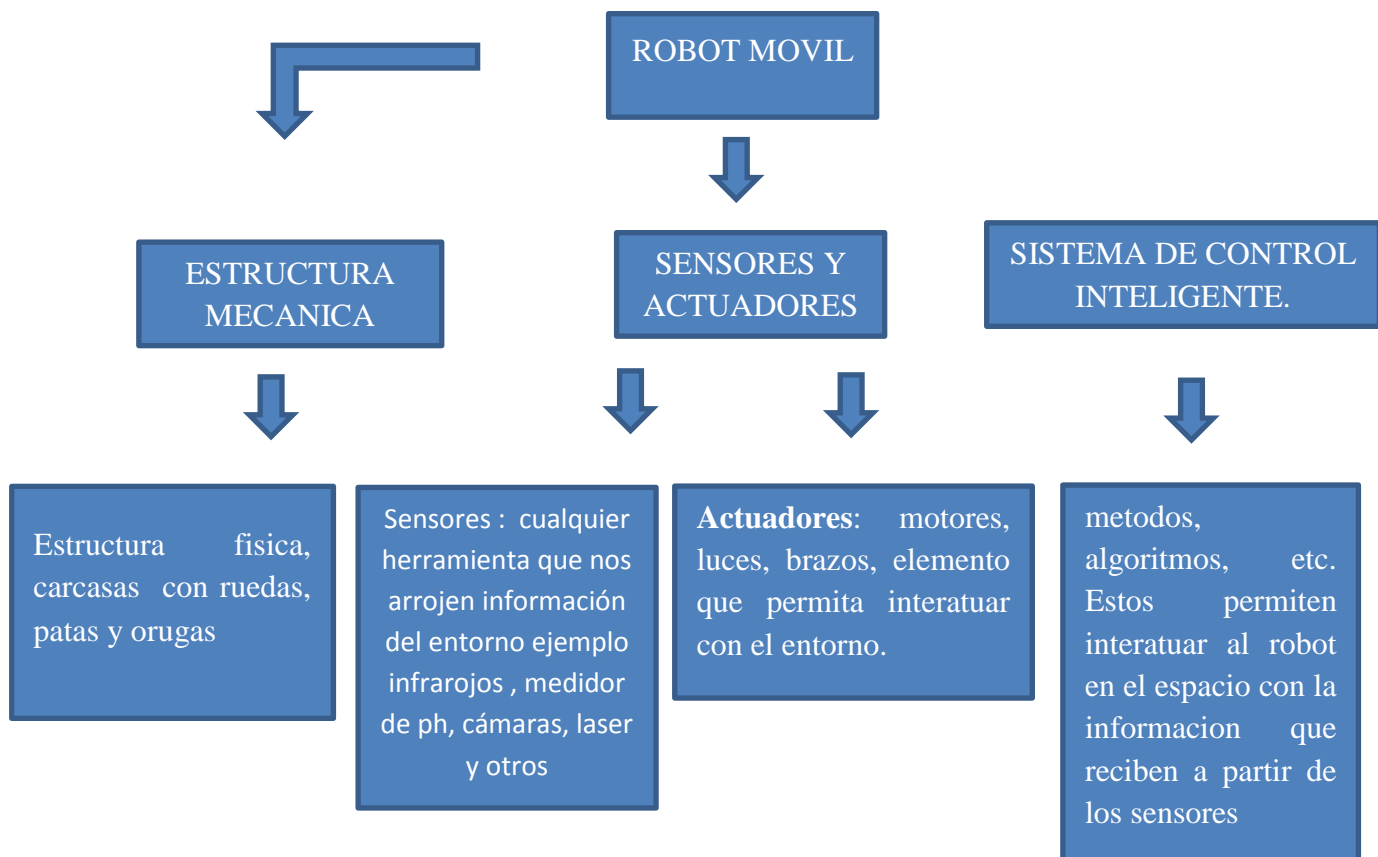
2.2.2 robots de patas

Tanto los robots basados en cadenas como en patas se pueden considerar más complicados y pesados, generalmente, que los robots de ruedas para una misma carga útil. A esto podemos añadir el que se pueden transformar vehículos de ruedas de radio control para usarlos como bases de robots.



2.3 Estructura general de un robot móvil

La estructura de un robot móvil se diseña simulando comportamientos de objetos o seres vivos. De acuerdo al entorno donde se va usar. La construcción de todos los robots se divide en: una estructura mecánica, sensores y actuadores, y un sistema de control inteligente.



2.4 ESTRUCTURA MECÁNICA

2.4.1 Estructura física

la estructura física es la forma que se diseña, la apariencia del robot según el entorno a usar y el peso a soportar de acuerdo a esto seleccionan los materiales para la construcción (madera, aluminio, vidrio, plástico y otros)

2.4.2 Ruedas

En el diseño de robots móviles las ruedas son el elemento que proporciona la capacidad de movilidad, se mueven por el contacto superficial o fricción con la superficie, se desplaza $2\pi r$ por vuelta, donde r es el radio de la rueda.

Se utiliza cuatro tipos de ruedas: tipo convencionales, tipo castor, tipo ruedas locas y omnidireccionales. Figura 5.

a) **Tipo convencional o también llamada fija:** El movimiento se produce en la dirección de las ruedas. No posee articulación de dirección, por lo que su posición respecto a la estructura es fija.

b) **Tipo castor:** Tiene articulación de dirección, es decir es orientable respecto a la estructura del vehículo, pasando su eje de dirección por el centro de rotación de la rueda.

c) **Tipo ruedas locas:** Gira sobre su mismo eje.

d) **Tipo omnidireccional u sueco:** Además de moverse en la dirección de la rueda, se mueve en dirección perpendicular la dirección de la rueda.

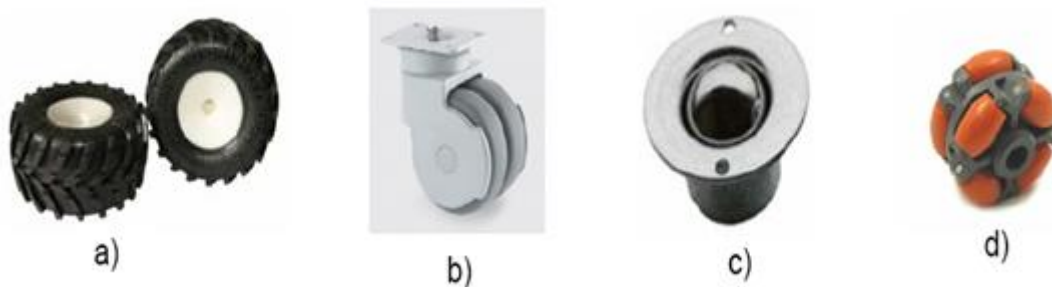


Figura 5. Tipos de ruedas.

Fuente: Tipos de ruedas para su locomoción (Goris,2005)

2.5 SENSORES Y ACTUADORES

2.5.1 Sensores

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser, por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor).

Sensor. (s.f.). En Wikipedia. Recuperado el 16 de diciembre de 2013 de <https://es.wikipedia.org/wiki/Sensor>

Este prototipo de robot va trabajar con un sensor llamado pH metro donde se está midiendo el nivel de nitrógeno de la tierra (PH). El pH es una medida utilizada por la química para evaluar la acidez o alcalinidad de una sustancia por lo general en su estado líquido (también se puede utilizar para gases). Se entiende por acidez la capacidad de una sustancia para aportar a una disolución acuosa iones de hidrógeno, hidrogeniones (H^*) al medio. La alcalinidad o base aporta hidroxilo OH^- al medio. Por lo tanto, el pH mide la concentración de iones de hidrógeno de una sustancia, a pesar de que hay muchas definiciones al respecto.

Como cualquier medida, el pH posee una escala propia que indica con exactitud un valor. Ésta es una tabla que va del número cero al catorce, siendo de esta manera el siete el número del medio. Si el pH es de cero a seis, la solución es considerada ácida; por el contrario, si el pH es de ocho a catorce, el sol unión se considera alcalina. Si la sustancia es más ácida, más cerca del cero estará; y entre más alcalina el resultado será más cerca del catorce. Si la solución posee un pH siete, es considerada neutra. Sin embargo, el pH siete neutro se limita con seguridad, tan sólo a las soluciones acuosas, pues las que no son, si no están a una temperatura y presión normal, el valor de la neutralidad puede variar.

La manera más exacta para la medición del pH, es utilizando un pH metro y dos electrodos, uno de referencia y otro de cristal. Un pH metro es un voltímetro que, junto con los electrodos, al ser sumergidos en una sustancia, generan una corriente eléctrica. Esta corriente eléctrica dependerá de la concentración de iones de hidrógeno que presente la solución. El pH metro mide la diferencia de potencial entre el electrodo de referencia (plata) y el de cristal que es sensible a los iones de hidrógeno. Para obtener con exactitud el pH de

una sustancia, se debe calibrar el pH con soluciones de valores de pH llamadas buffer que resisten los cambios experimentados por el pH y tiene un valor de pH específico.

2.5.1.1 Electrodo de Referencia

En muchas aplicaciones es deseable que el potencial de media celda de uno de los electrodos sea conocido, constante y completamente insensible a la composición de la solución en estudio. Un electrodo con estas características, se denomina electrodo de referencia.

Un electrodo de referencia debe ser fácil de montar, proporcionar potenciales reproducibles y tener un potencial sin cambios. Dos electrodos comúnmente utilizados que satisfacen estos requisitos son el Electrodo de Calmell y el Electrodo de Plata-Cloruro de Plata.

Electrodos Indicadores Junto con el electrodo de referencia se utiliza un electrodo indicador cuya respuesta depende de la concentración del analito. Los electrodos indicadores para las medidas potenciométricas son de dos tipos fundamentales, denominados metálicos y de membrana.

2.5.1.2 Electrodo Indicador Metálico

Se utilizan para la cuantificación del catión proveniente del metal con que está construido el electrodo.

2.5.1.3 Electrodo Indicador de Membrana

Desde hace muchos años, el método más adecuado para la medida del pH consiste en medir el potencial que se desarrolla a través de una membrana de vidrio que separa dos soluciones con diferente concentración de ion hidrógeno.

Es conveniente clasificar los electrodos de membrana en base a la composición de dicha Membrana.

A. Electrodo de membrana cristalina

1. Cristal simple (Ejemplo: LaF_3 para determinar de F^-)
2. Cristal poli cristalino o mezcla (Ejemplo: Ag_2S para determinar S^{2-} o Ag^+)

B. Electrodo de membrana no cristalina

1. Vidrio (Ejemplo: vidrios al silicato para determinar H^+ y cationes monovalentes como Ni^+)
2. Líquida (Ejemplo: intercambiadores de iones líquidos para determinar Ca^{+2} y transportadores neutros para K^+)
3. Líquido inmovilizado en polímero rígido (Ejemplo: matriz de PVC para Determinar Ca^{+2} , NO_3^-)

El mecanismo general por el cual se desarrolla un potencial selectivo al ion en estos elementos es enteramente diferente de la fuente de potencial en electrodos de indicadores metálicos. El potencial de un electrodo metálico surge de la tendencia de una reacción química de oxidación/reducción a ocurrir en la superficie de un electrodo. En electrodos de membrana, en cambio, el potencial observado es una clase de potencial de unión que se desarrolla a través de la membrana que separa a la solución del analito de una solución de referencia.

2.5.1.4 Electrodo de Vidrio Para la Medida del pH

La Figura 6 muestra una celda para la medida del pH. Consiste en un par de electrodos, uno de plata/cloruro de plata y otro de vidrio sumergidos en la solución cuyo pH se desea medir.

Obsérvese que la celda contiene dos electrodos de referencia, cada uno con un potencial constante e independiente del pH; uno de estos electrodos de referencia es el electrodo interno de plata/cloruro de plata, que es un componente del electrodo de vidrio pero que no es sensible al pH. Es la delgada membrana en el extremo del electrodo, la que responde a los cambios de pH.

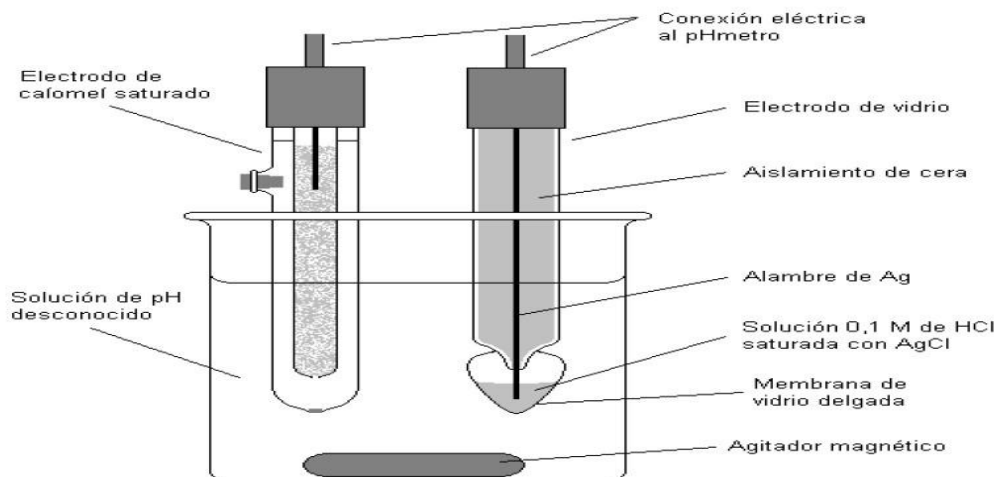


Figura 6. Sistema típico de electrodos para la medición potencio métrica del pH

Fuente : <http://aplicacionesdeloselectro2.blogspot.com.co>(Sequeda M & Orozco D 2006)

Actualmente se utiliza un solo electrodo que condensa los dos electrodos anteriores y que se denomina electrodo combinado de vidrio, tal como se indica en la figura 7.

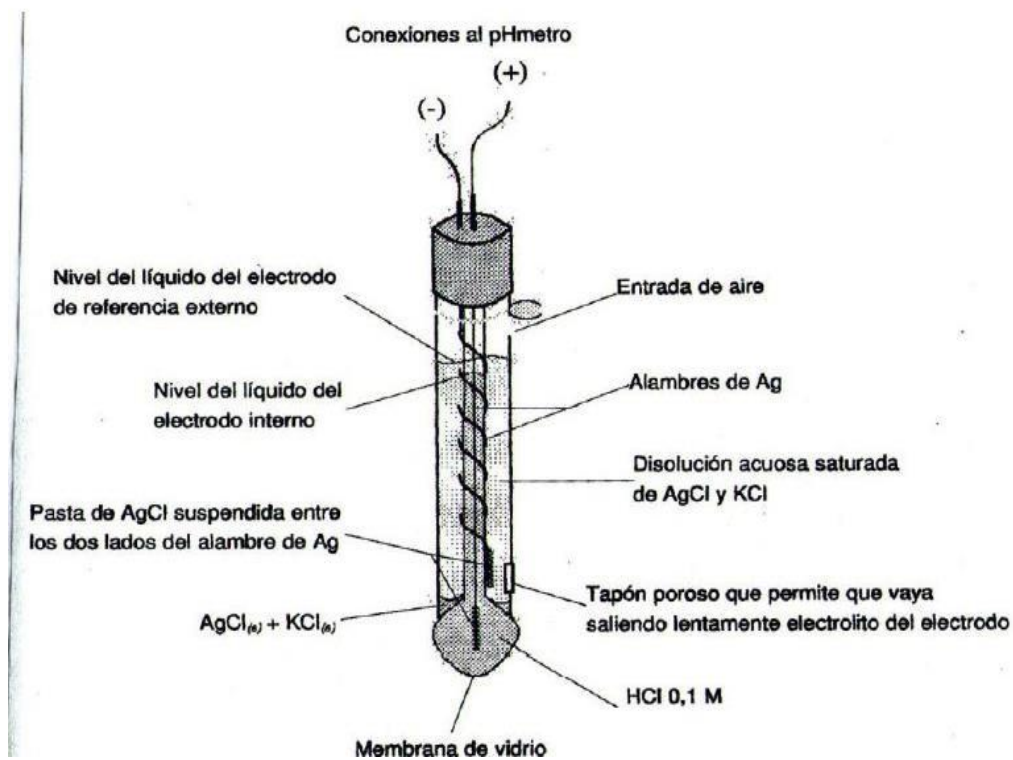


Figura 7 – Electrodo potenciométrica del pH

Fuente: <http://aplicacionesdeloselectro2.blogspot.com.co>(Sequeda M & Orozco D 2006)

Generalmente se utilizan dispositivos de medición de pH que están directamente conectados al factor de medición, uno de ellos es el pH metro portátil Hanna HI99121 el cual mide el pH directamente en el suelo tal como se ilustra en la figura 8.



Figura 8. pH metro

Fuente : http://www.infoagro.com/abonos/pH_informacion.htm (Moratiel R)

La medida del pH de una disolución acuosa, ya sea un agua de riego, una disolución nutritiva para fertirrigación o los lixiviados (drenaje) procedentes de un cultivo con recirculación de la disolución nutritiva, requiere: a, Equipos adecuados, como los que HANNA Instruments HI98130 y HI98127 que son equipos de bolsillo ilustrados en la figura 9.



Figura 9. Instrumento de medición de bolsillo de pH suelo

Fuente: http://www.infoagro.com/abonos/pH_informacion.htm (Moratiel R)

2.5.2 ACTUADORES

2.5.2.1 MOTORES DC

motor de corriente continua (denominado también motor de corriente directa, motor CC o motor DC) es una máquina que convierte la energía eléctrica en mecánica, provocando un movimiento rotatorio, gracias a la acción que se genera del campo magnético. Una máquina de corriente continua (generador o motor) se compone principalmente de dos partes. El estator da soporte mecánico al aparato y contiene los devanados principales de la máquina, conocidos también con el nombre de polos, que pueden ser de imanes permanentes o devanados con hilo de cobre sobre un núcleo de hierro. El rotor es generalmente de forma cilíndrica, también devanado y con núcleo, alimentado con corriente directa



Figura 10 Motor DC

Fuente:www.pololu.com

2.5.2.2 **SERVOMOTORES**

un servomotor es un dispositivo pequeño que tiene un eje de rendimiento controlado. Este puede ser llevado a posiciones angulares específicas al enviar una señal codificada. Con tal de que una señal codificada exista en la línea de entrada, el servo mantendrá la posición angular del engranaje. Cuando la señal codificada cambia, la posición angular de los piñones cambia. En la práctica, se usan servos para posicionar superficies de control como el movimiento de palancas, pequeños ascensores y timones. Ellos también se usan en radio control, títeres, y por supuesto, en robots.

Los Servos son sumamente útiles en robótica. Los motores son pequeños, tiene internamente una circuitería de control interna y es sumamente poderoso para su tamaño. Un servo normal o Standard. Un servo, por consiguiente, no consume mucha energía. Se muestra la composición interna de un servo motor en el cuadro de abajo. Podrá observar la circuitería de control, el motor, un juego de piñones, y la caja. También puede ver los 3 alambres de conexión externa. Uno es para alimentación Vcc (+5volts), conexión a tierra GND y el alambre blanco es el alambre de control.

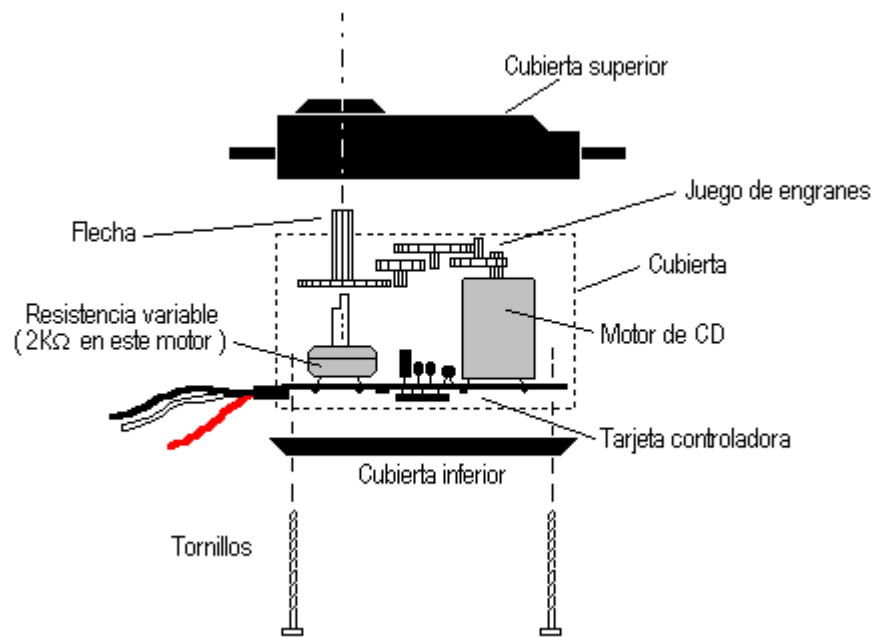


Figura 11 servomotor

Fuente: <http://www.info-ab.uclm.es/labelec/solar/electronica/elementos/servomotor.htm>

FUNCIONAMIENTO

El motor del servo tiene algunos circuitos de control y un potenciómetro (una resistencia variable) esta es conectada al eje central del servo motor. Este potenciómetro permite a la circuitería de control, supervisar el ángulo actual del servo motor. Si el eje está en el ángulo correcto, entonces el motor está apagado. Si el circuito chequea que el ángulo no es el correcto, el motor girará en la dirección adecuada hasta llegar al ángulo correcto. El eje del servo es capaz de llegar alrededor de los 180 grados o 360 grados si es de rotación continua

2.6 SISTEMA DE CONTROL INTELIGENTE

Un sistema de control es un sistema que una vez puesto en funcionamiento sigue su proceso sin intervención del ser humano. Se caracteriza por estar formado por elementos variables de entrada, sistema, variable de salida.

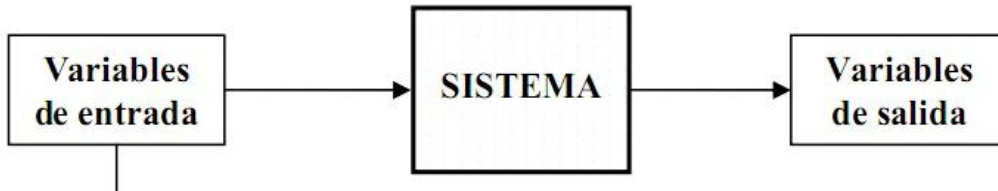


Figura 12 sistema de control

Fuente: <https://thecontrolengineer-ep.wikispaces.com/Control-Sistemas-Variables>

Existen dos tipos de sistemas de control:

a) **Sistema de control de lazo abierto:** son los que en su ejecución no influyen las variables de entrada ni de salida.

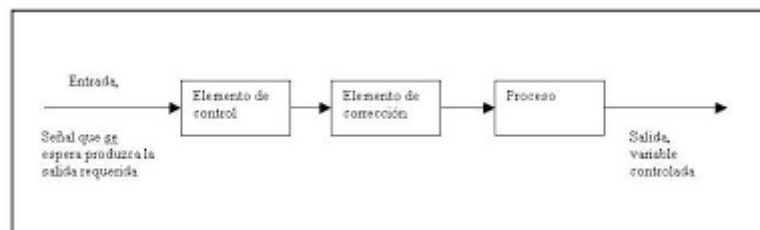


Figura 13 lazo abierto

Fuente: <https://thecontrolengineer-ep.wikispaces.com/Control-Sistemas-Variables>

b) **Sistema de control de lazo cerrado:** son aquellos en los que influyen las variables de salidas, retroalimentando el sistema

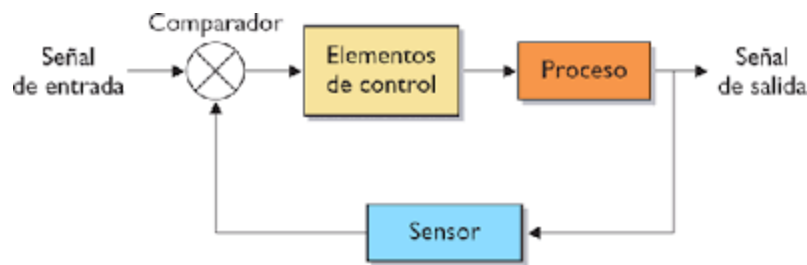


Figura 14 lazo cerrado

Fuente: <http://guinea-edeso.blogspot.com.co/2012/02/ejemplos-de-control-en-lazo-cerrado.html>

Estos sistemas de control de lazo abierto y cerrado se pueden diseñar por medio de control analógico control digital, control difuso.

2.6.1 Tarjeta Arduino mega 2560

El Arduino Mega 2560 es una placa electrónica basada en el Atmega2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos serie de hardware), un oscilador de 16MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB o la corriente con un adaptador de CA a CC o una batería para empezar

Especificaciones técnicas

| | |
|----------------------------------|---|
| Microcontrolador | Atmega2560 |
| Tensión de funcionamiento | 5V |
| Voltaje de entrada (recomendado) | 7-12V |
| Voltaje de entrada (límite) | 6-20V |
| E / S digitales prendedores | 54 (de los cuales 15 proporcionan salida PWM) |
| Pines de entrada analógica | Dieciséis |

| | |
|---------------------------------|--|
| Corriente continua para Pin I/O | 20 mA |
| Corriente CC para Pin 3.3V | 50 mA |
| Memoria flash | 256 KB, 8 KB utilizado por el gestor de arranque |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Velocidad de reloj | 16 MHz |
| Longitud | 101.52 mm |
| Anchura | 53,3 mm |
| Peso | 37 g |

Tabla 2 especificaciones técnicas

Fuente <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

Memoria

El Atmega2560 tiene 256 KB de memoria flash para almacenar el código (de la que se utiliza 8 KB para el cargador de arranque), 8 KB de SRAM y 4 KB de EEPROM

2.6.2 Software Arduino

la plataforma Arduino se programa mediante el uso de un lenguaje de programación que sirve para controlar los distintos sensores que se encuentran conectados a la placa, por medio de instrucciones y parámetros que nosotros establecemos al conectar la placa a un ordenador. Este lenguaje que opera dentro de Arduino se llama Wiring, basado en la plataforma Processing y primordialmente en el lenguaje de programación C/C++, Arduino soporta varios lenguajes de programación de alto nivel derivados de C, haciendo de esto una ventaja para los diseñadores que trabajan en varios o en 1 sólo entorno de desarrollo de programación. Para poder trabajar desde el nivel programación del procesador, debe

descargarse el software que incluye las librerías necesarias para poder utilizar el lenguaje de manera completa.



Figura 15 software Arduino

Fuente: imagen realizada por Autor

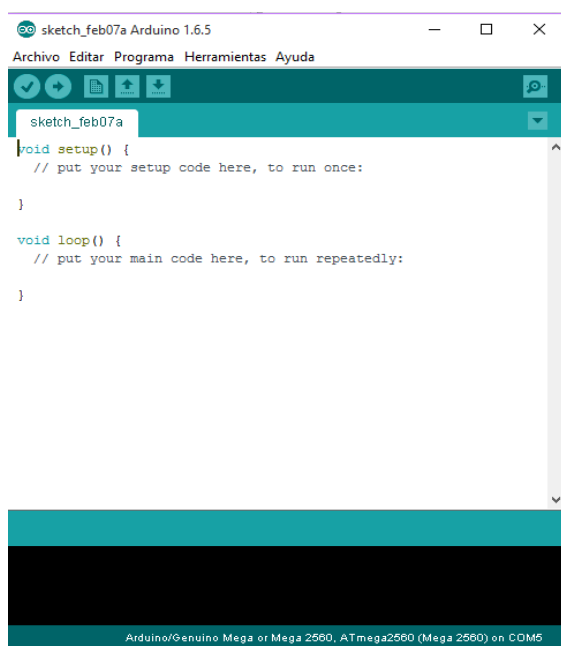


Figura 16 ventana Arduino

Fuente: imagen realizada por Autor

2.7 Comunicación de datos USB 2.0

El USB (Universal Serial Bus) es un puerto que sirve para conectar periféricos a un ordenador. Fue originalmente pensado para conectar dispositivos a los ordenadores, eliminando la necesidad de conectar tarjetas PCI (o similares), y también para conectar y desconectar dispositivos sin tener que reiniciar el ordenador.

Existen tres versiones del protocolo (1.0, 1.1 y 2.0). A diferencia de las anteriores, la última versión (2.0) soporta tasas de transferencia de altas velocidades, comparables (o incluso superiores) a la de un disco duro o almacenamiento magnético, lo cual ha permitido ampliar el uso del USB a aplicaciones de video y almacenamiento (discos duros externos). Una de las razones a la cual se atribuye su gran aceptación es que todas las versiones del protocolo son compatibles con las anteriores. Es decir, que cualquier dispositivo 2.0 puede ser conectado a un dispositivo 1.0, aunque funcionará a la velocidad del más lento.

Existen tres tipos de velocidades en la comunicación:

| <u>Tipo</u> | <u>Velocidad</u> |
|---------------------------------|--------------------------|
| Baja velocidad (low speed) | 183 Kbytes/s (1.5Mbit/s) |
| Velocidad completa (full speed) | 1.4 Mbytes/s (12Mbit/s) |
| Alta velocidad (high speed) | 57 Mbytes/s (480 Mbit/s) |

Figura 17. Velocidades de transferencia.

Fuente: PDF Control de un motor paso a paso: PIC, USB y C#, Pág. 29.

2.11.1 Topología

USB tiene un diseño asimétrico ya que consiste de un host controlador conectado a múltiples dispositivos conectados en daisy-chain (Esquema de cableado). Tomado del PDF (Control de un motor paso a paso: PIC, USB y C#, Pág. 30)

USB conecta varios dispositivos a un host controlador a través de cadenas de hubs. Los hubs (al igual que en redes) son dispositivos que permiten, a partir de un único punto de conexión, poder conectar varios dispositivos, es decir, disponer de varios puntos de conexión. De esta forma se crea una especie de estructura de árbol. El estándar admite hasta 5 niveles de ramificación por host controlador con un límite absoluto de 127 dispositivos conectados al mismo bus (incluyendo los hubs). Siempre existe un hub principal (conocido como el hub raíz) que está conectado directamente al host controlador.

Un mismo dispositivo USB puede cumplir varias funciones. Por ejemplo, un mouse puede ser también lector de tarjetas, y de esa forma sería como dos dispositivos conectados al bus USB. Por lo tanto, es preferible hablar de funciones en lugar de dispositivos.

2.11.2 Funcionamiento

Los dispositivos tienen asociados unos canales lógicos unidireccionales (llamados pipes) que conectan al host controlador con una entidad lógica en el dispositivo llamada endpoint. Los datos son enviados en paquetes de longitud variable. Típicamente estos paquetes son de 64, 128 o más bytes (64 bytes en el caso del software de este trabajo).

Estos endpoints (y sus respectivos pipes) son numerados del 0 al 15 en cada dirección, por lo que un dispositivo puede tener hasta 32 endpoints (16 de entrada y 16 de salida). La dirección se considera siempre desde el punto de vista del host controlador. Así un endpoint de salida será un canal que transmite datos desde el host controlador al dispositivo. Un endpoint solo puede tener una única dirección. El endpoint 0 (en ambas direcciones) está reservado para el control del bus. Tomado del PDF (Control de un motor paso a paso: PIC, USB y C#), Pág. 30.

2.13. Interfaz

La interfaz fue realizada en el software de java NetBeans

3. ASPECTOS METODOLÓGICOS

3.1 Diseño y construcción de la planta

3.1.1 Diseño

El diseño, del prototipo de robot medidor de pH en suelos llanos, se muestra en la figura Se basó en la configuración de tracción Ackerman de los robots móviles por ser un esquema sencillo, basado en dos ruedas en la parte trasera controladas independientemente y dos ruedas delanteras en el mismo eje controladas por un mismo motor para el direccionamiento del prototipo.

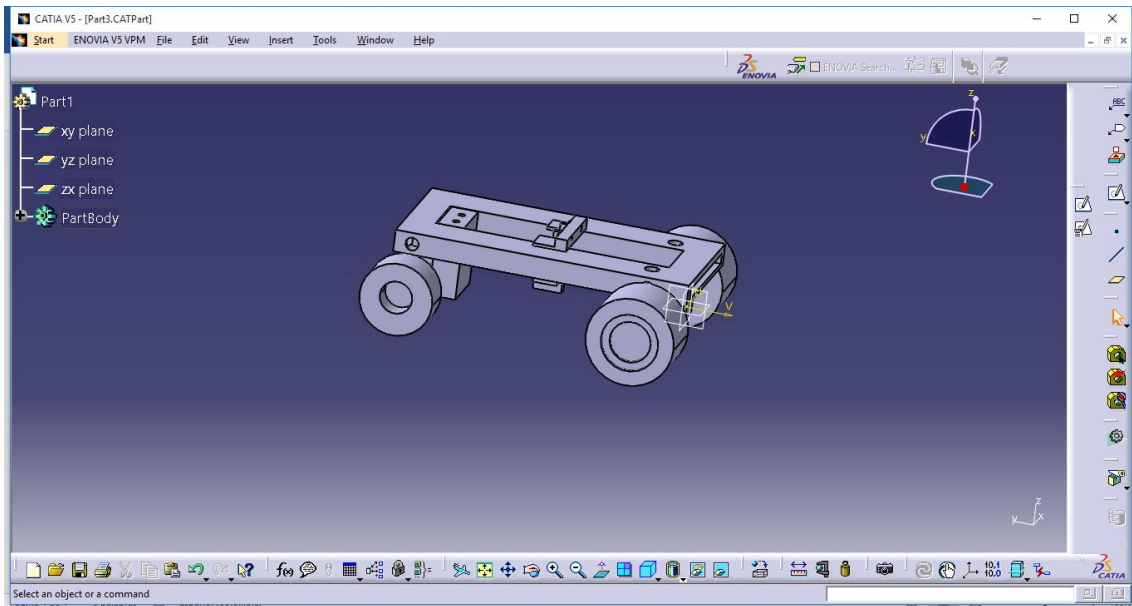


Figura 18 simulación de prototipo configuración de tracción Ackerman

Fuente: Planos realizada por Autor en Catia v5.

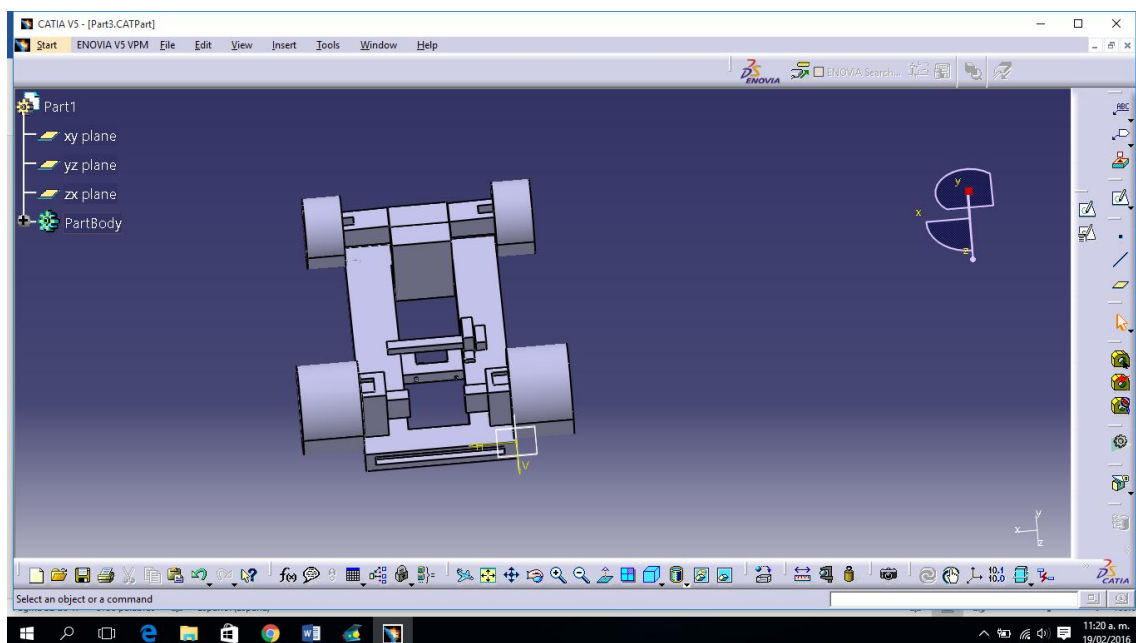


Figura 19. Diseño del prototipo vista superior.

Fuente: Planos realizada por Autor en Catia v5.

3.1.2 Estructura

La estructura que se realizó está diseñada en aluminio por su poco peso y dureza las dimensiones fueron de 33cm de largo por 12.5cm de ancho con un hueco en el centro de 26cm de largo por 4cm de ancho y un grosor de 2cm fig.

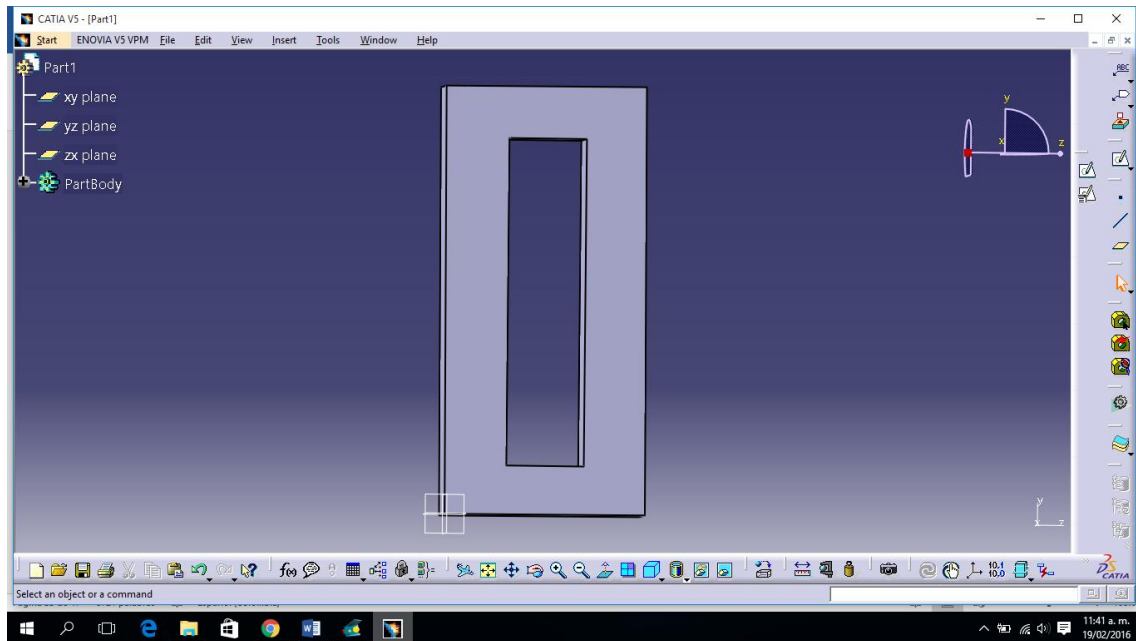


Figura 20. Diseño del prototipo vista superior carcasa del prototipo.

Fuente: Planos realizada por Autor en Catia v5.

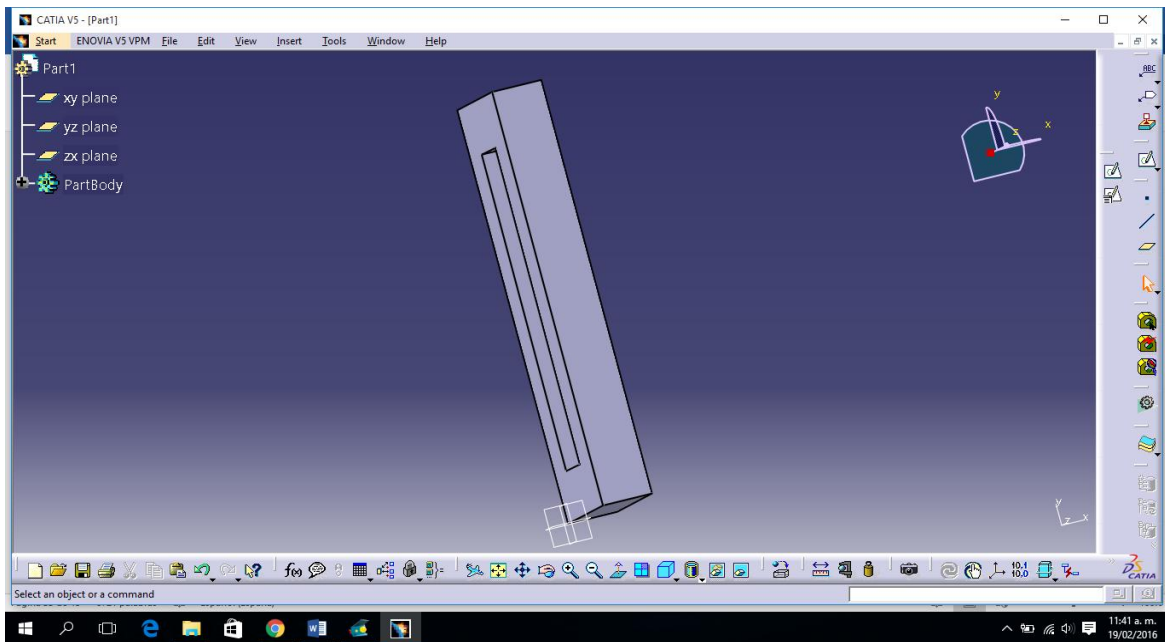


Figura 21. Diseño del prototipo vista lateral carcasa del prototipo.

Fuente: Planos realizada por Autor en Catia v5.

3.1.2.1 Soporte de los servomotores y ruedas traseras

El soporte se diseñó en aluminio con las siguientes dimensiones del largo de 9cm Y ancho de 4cm por un frente destapado y por el otro una cavidad de 4cm de largo por 2cm de ancho donde se ubica el servomotor. de estos se realizaron dos uno para cada llanta trasera

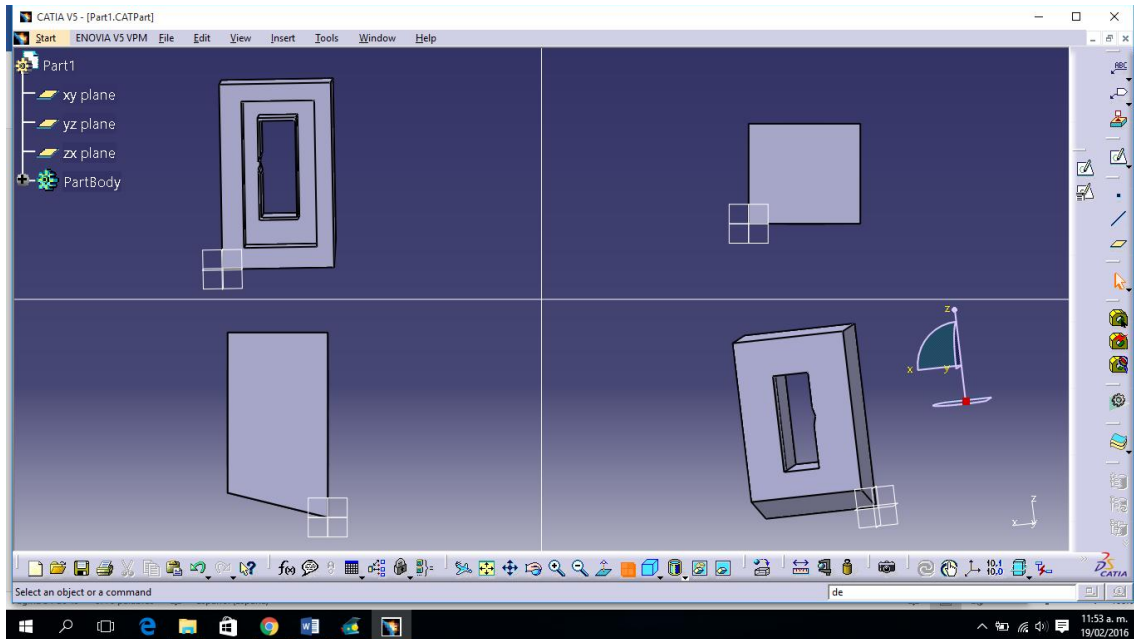


Figura 22. Diseño del prototipo vistas soporte ruedas traseras.

Fuente: Planos realizada por Autor en Catia v5.

3.1.2.2 Transmisión delantera

Para la transmisión delantera se consiguió una de un carro de juguete marca DaimlerChrysler Dodge con sus respectivas ruedas la transmisión tiene una altura de 9cm y ancho en la parte arriba de 3.5 cm y la parte inferior de 13cm

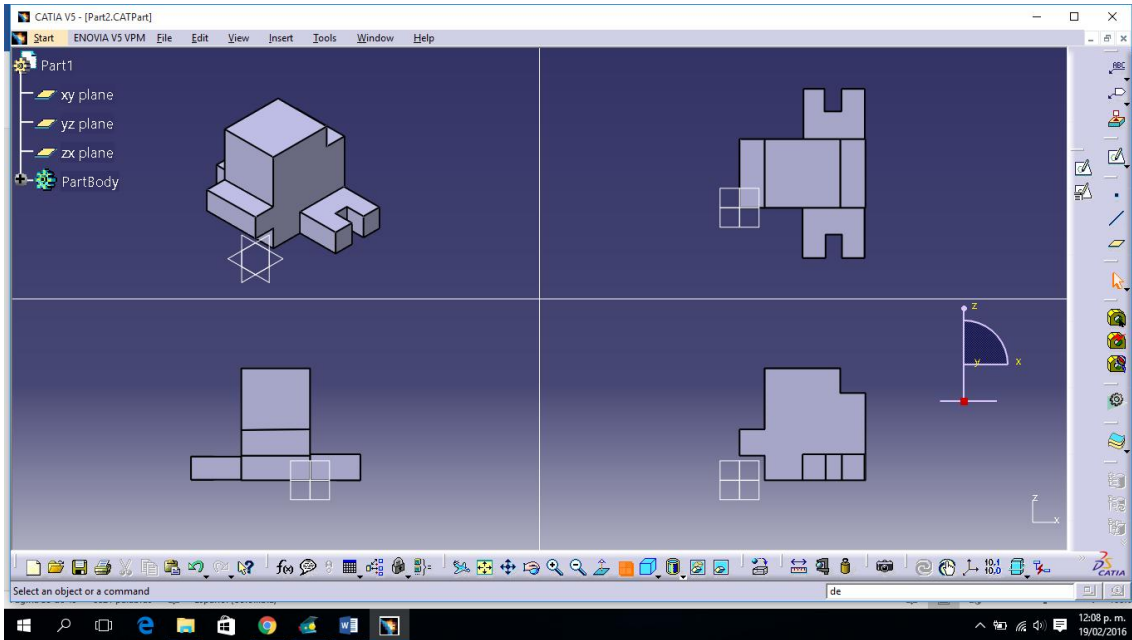


Figura 23. Diseño del prototipo vistas soporte ruedas delanteras.

Fuente: Planos realizada por Autor en Catia v5.

3.1.2.3 Soporte del sensor medidor de pH (PHMETRO)

Se diseñó en partes donde el sensor se conecta a una cremallera de 10 cm de largo por 1.5 cm, se sostiene por medio de un Angulo elaborado en aluminio de 6 cm de largo por 2cm de ancho y un soporte para el motor de 8 cm por 2cm de ancho

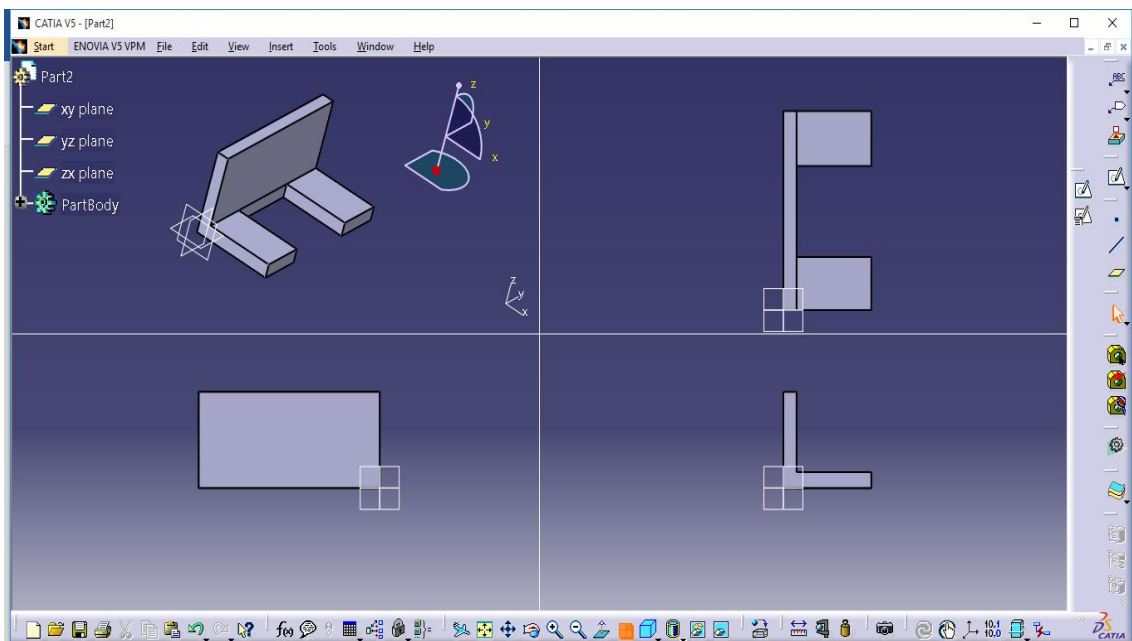


Figura 24. Diseño del prototipo vistas soporte servomotor.

Fuente: Planos realizada por Autor en Catia v5.

3.1.2.4 Estructura total del prototipo

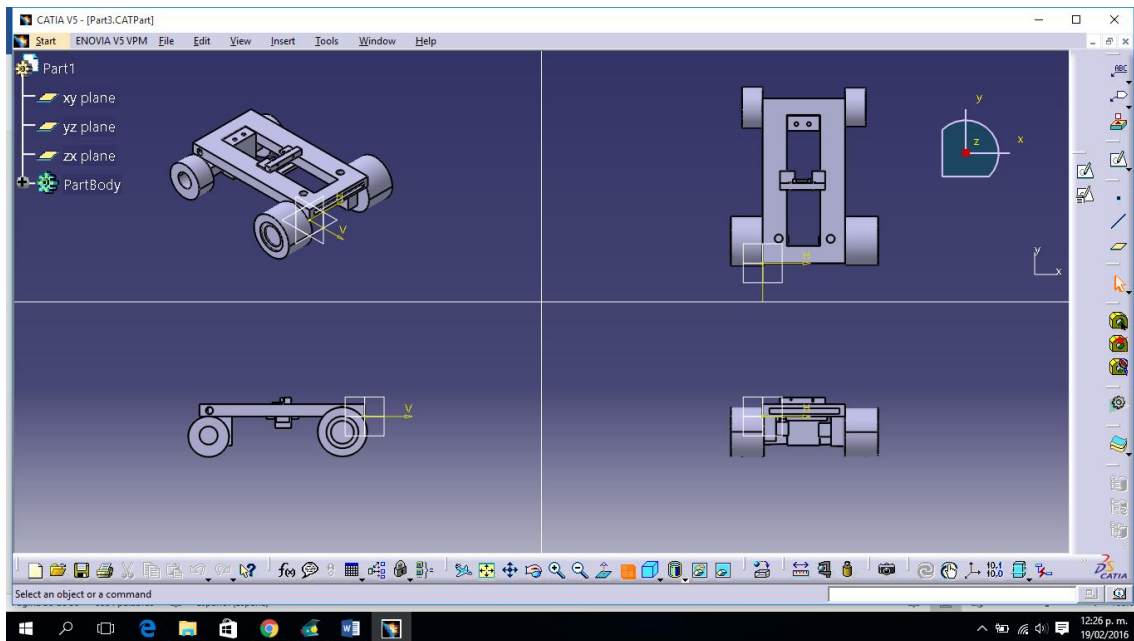


Figura 25. Estructura final del prototipo vistas.

Fuente: Planos realizada por Autor en Catia v5.

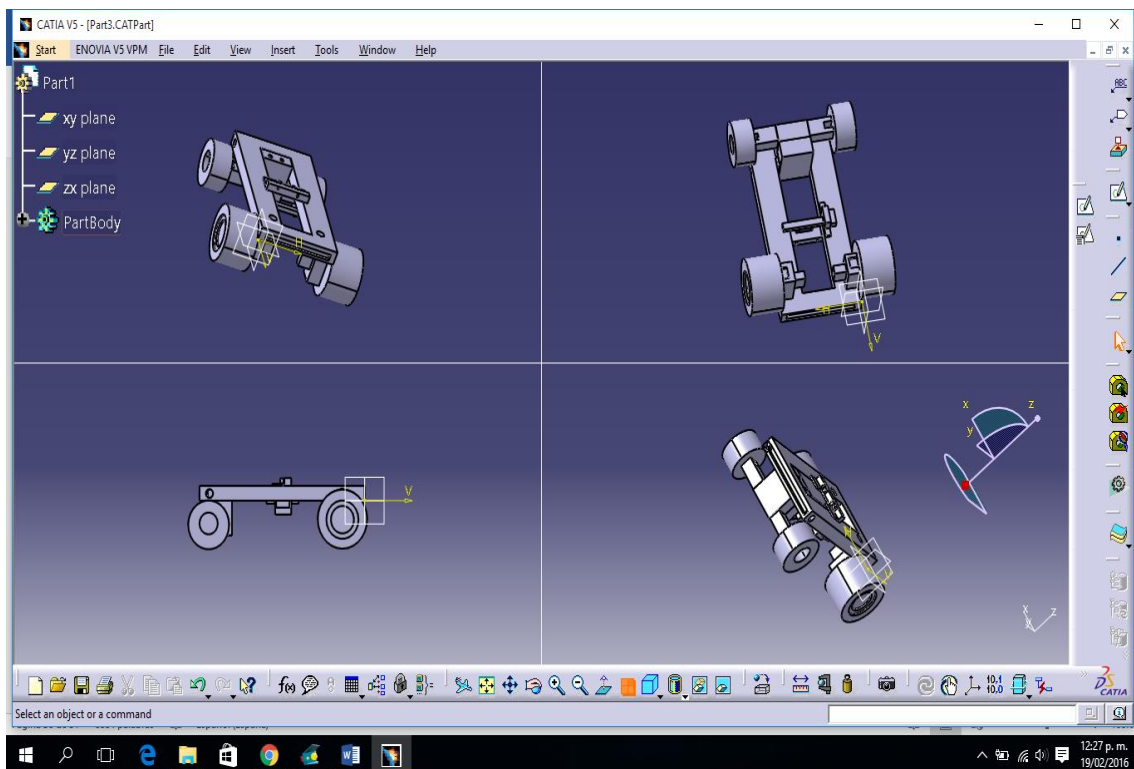


Figura 26. Diseño del prototipo vistas 2.

Fuente: Planos realizada por Autor en Catia v5.

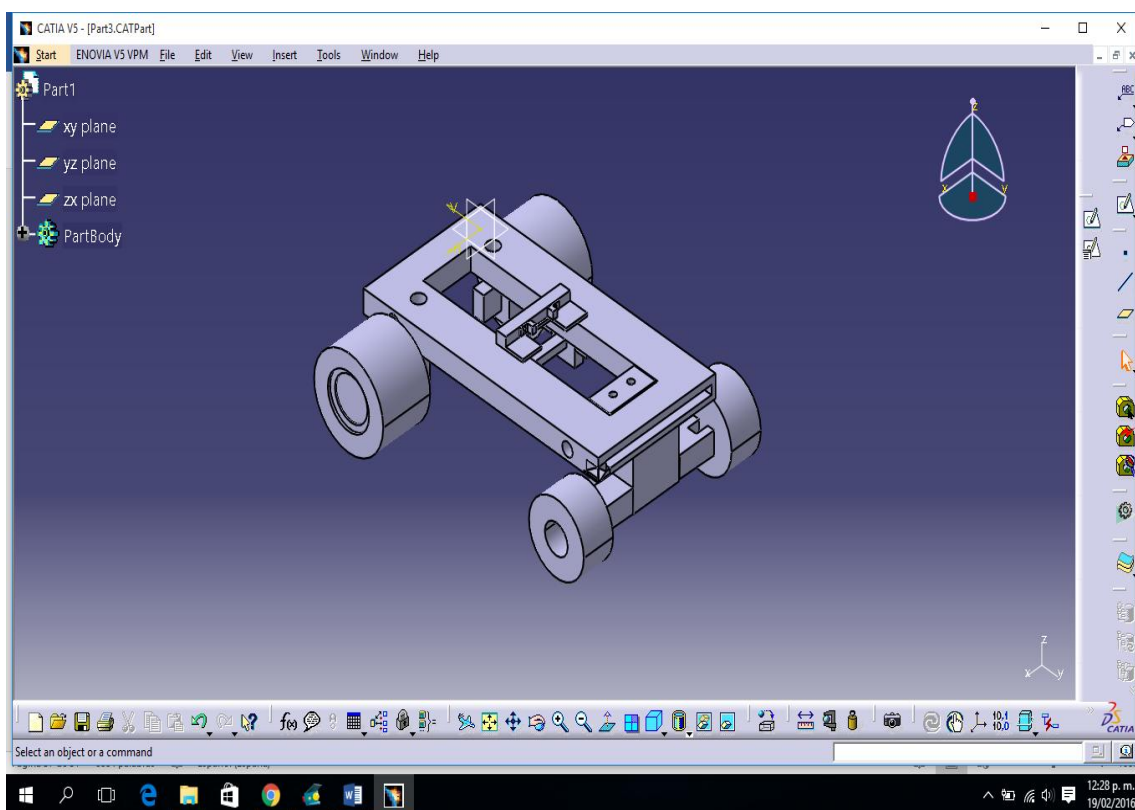


Figura 27. Prototipo final.

Fuente: imagen realizada por Autor.

3.2 Materiales

3.2.1 Cantidad de materiales

El material que se usó para la construcción del prototipo fue

- Aluminio 92 cm
- 2 ruedas de 9 cm de diámetro por 6cm de ancho elaboradas en plástico
- 2 ruedas de 11 cm de diámetro por 5.5 cm de ancho elaboradas en plástico
- 2 servomotores de marca hitico de giro continuo con un torque de 5kg
- 1 servo de 180 grados de giro con un torque de 2.5 kg
- 1 motor dc a 5v
- 1 cremallera de 10 cm
- 1 piñón 1.5cm de diámetro
- Tornillos

- 1 tarjeta Arduino mega 2560
- 1 lm293
- Resistencias
- 1 pH metro

3.3 Hardware

3.3.1 Conexión servo rueda izquierda

Para instalar el servo se conecta el cable rojo a Vcc, el cable café a gen y el cable amarillo al pin de la Arduino donde se le envía el comando en forma digital

3.3.2 Conexión servo rueda derecha

Para instalar el servo se conecta el cable rojo a Vcc, el cable café a gen y el cable amarillo al pin de la Arduino donde se le envía el comando en forma digital

3.3.3 Conexión servo del sensor de pH (pH metro)

Para instalar el servo se conecta el cable rojo a Vcc, el cable café a gen y el cable amarillo al pin de la Arduino donde se le envía el comando en forma digital

3.3.4 Conexión motor dc para giro tracción delantera

Se conecta el integrado l293d y se toma el pin 10 y 14 como entradas. Estos Reciben la información desde la tarjeta Arduino por los pines

Del integrado l293d se toma los pines 11 y 14 como salidas y se conecta a los cables del motor para realizar el giro en ambos sentidos

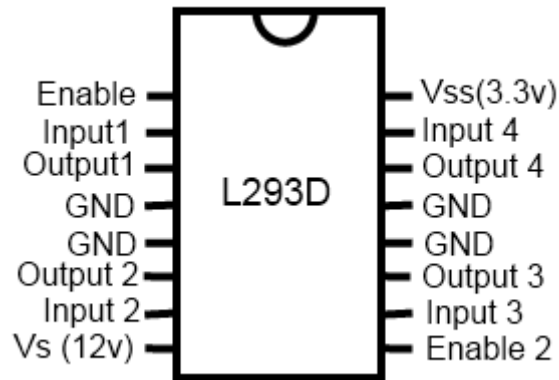


Figura 28 datasheet L293D

Figura tomada desde el pago <http://electronics.stackexchange.com/questions/23702/can-i-wire-the-two-sides-of-a-l293d-dual-h-bridge-together-if-i-only-need-one-h>

Ventajas del driver l293b (Data Sheet L293B).

- Cada canal es capaz de entregar hasta 1A de corriente.
- Posee una entrada de alimentación independiente que alimenta los 4 Drivers, es decir la que requieren los motores...
- El control de los Drivers es compatible con señales TTL es decir con 5 voltios (estamos hablando de señales lógicas).
- Cada uno de los 4 Drivers puede ser activado de forma independiente (por su terminal de entrada), o habilitado de dos en dos con un sólo terminal (Enable).

3.3.5 Funcionamiento del integrado l293

Este circuito se conoce como un “puente en H”, en el sentido de que está esperando dos entradas (1 ó 0 lógicos) por un lado, y repite exactamente estas dos entradas (1 ó 0 lógicos) por el otro lado, pero con la posibilidad de cambiar la fuente de alimentación, (que puede llegar a 36 V), y garantizando el consumo de intensidad del dispositivo a conectar. (Data Sheet L293B).

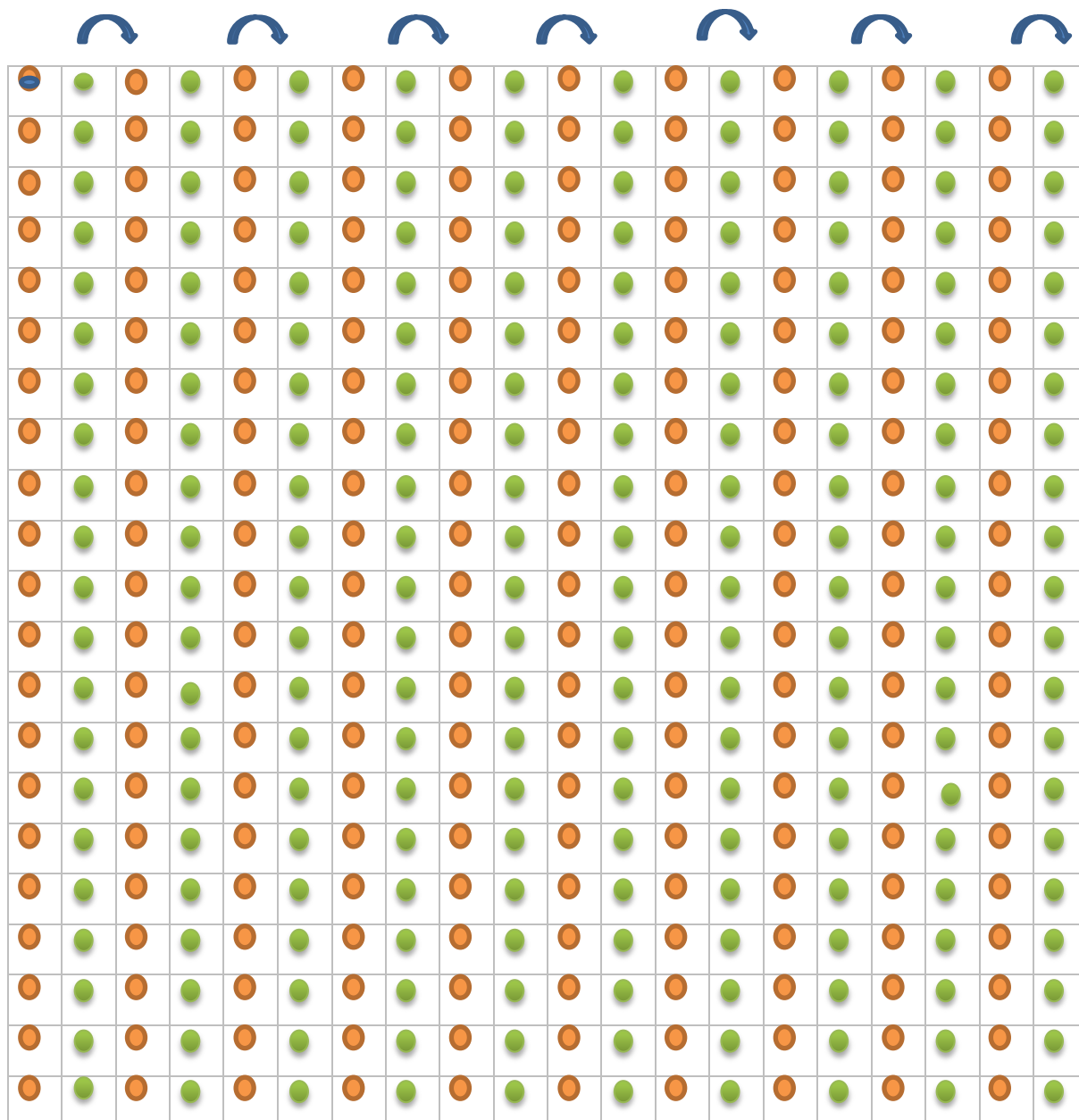
El L293B puede convertir las señales lógicas de la tarjeta Arduino Mega en señales de potencia aplicables al motor, este driver se controla directamente desde la tarjeta Arduino Mega y amplificara las señales para poder activar el motor.

3.3.6 Conexión del pH metro

El sensor de pH metro se conecta el cable de salida al pin de la tarjeta Arduino donde se conecta al pin como entrada

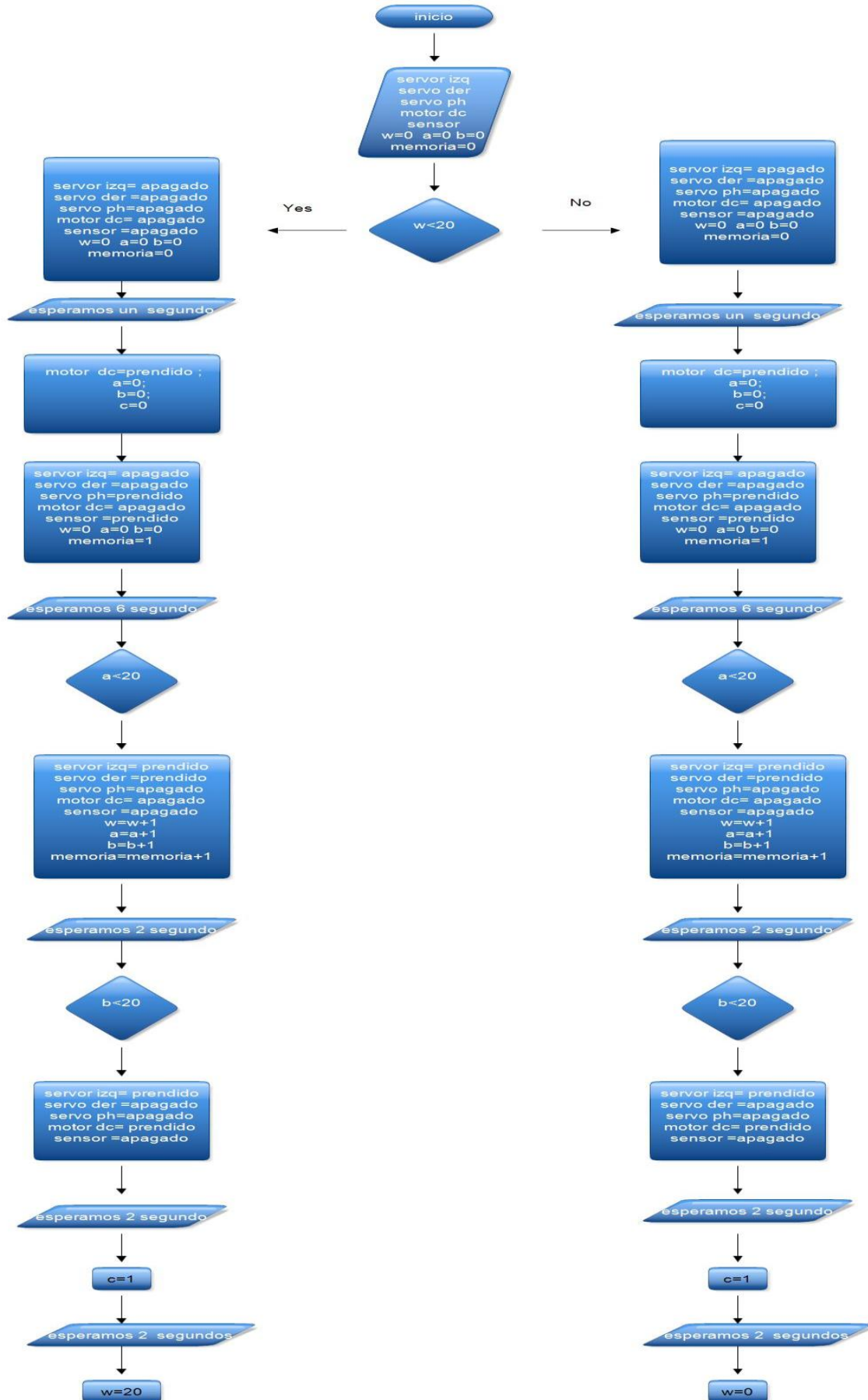
3.4 Trayectoria

la trayectoria que debe seguir el prototipo es en línea recta parando cada metro hasta la parada numero 20 donde hará un giro de 90 grados hacia la derecha ubicándose nuevamente en la parada 1 y va hasta la parada 20 y gira 90 grados, pero hacia el lado izquierdo y así sucesivamente hasta completar 20 veces la rutina



3.5 Control

3.5.1 Diagrama de flujo



3.5.2 Diseño de control software en Arduino

3.6. Prototipo Real



Figura 29. Diseño del prototipo vista superior carcasa del prototipo.

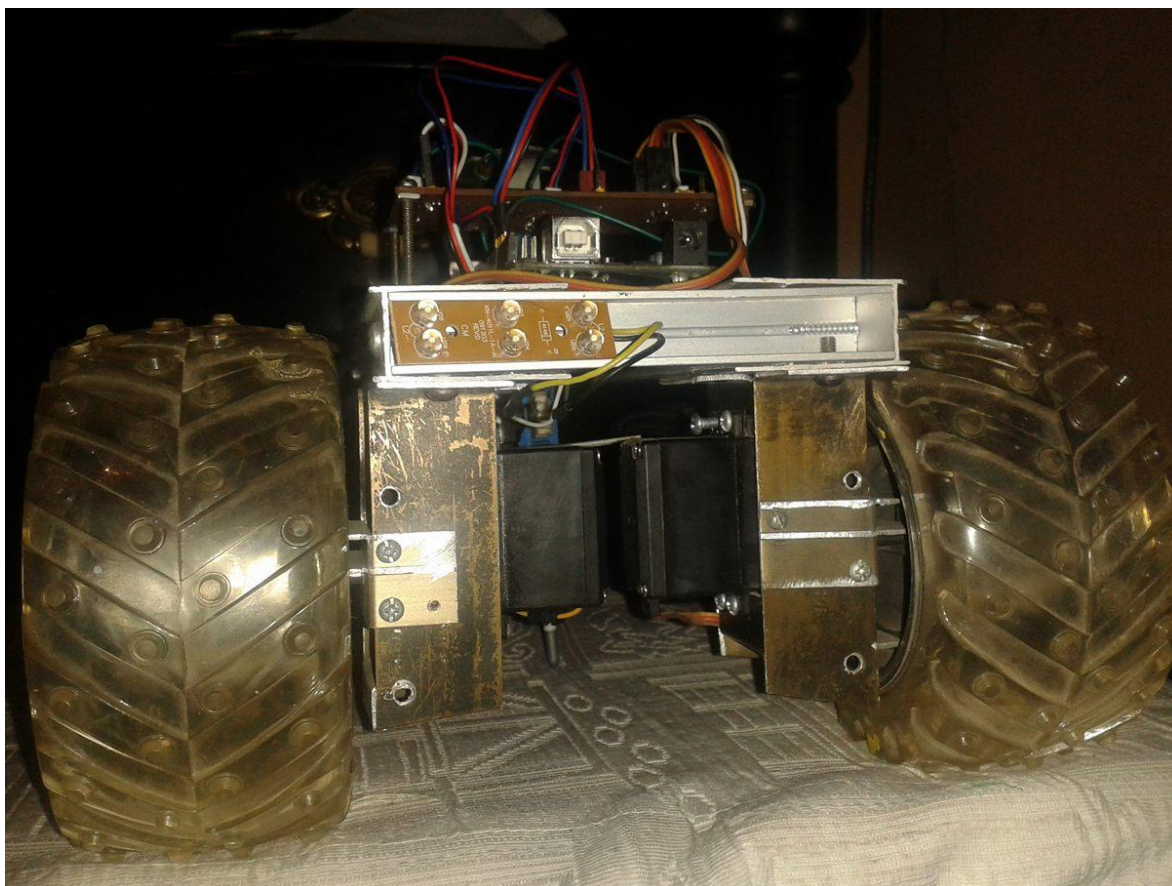


Figura 30. Diseño del prototipo vista trasera



Figura 31. Diseño del prototipo vista lateral



Figura 32. Diseño del prototipo vista inferior

3.7 Desarrollo de interfaz en java con software NetBeans

La interfaz se desarrolla mediante el software NetBeans para interacción del usuario con los datos arrojados por el prototipo de robot medidor de pH

3.7.1 Pasos del desarrollo

1. crear el paquete con la clase trabajo de grado

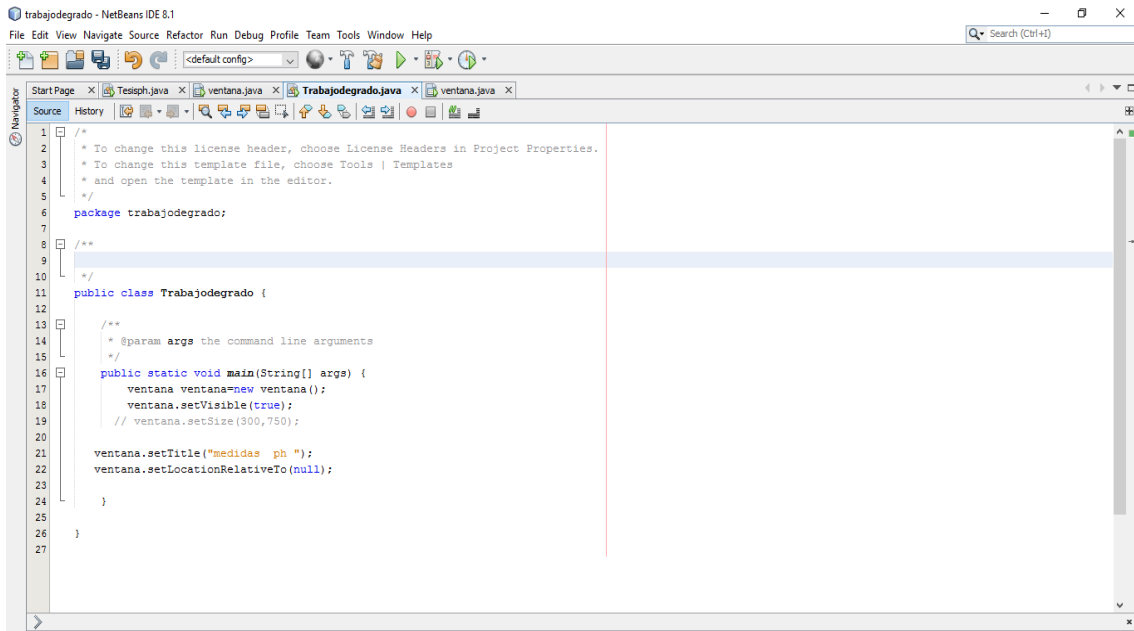


Figura 33 ventana NetBeans creación del paquete
Fuente: imagen realizada por Autor

2. Creamos un JFrame con el nombre ventana y ubicamos botones, jlist, label, jtext

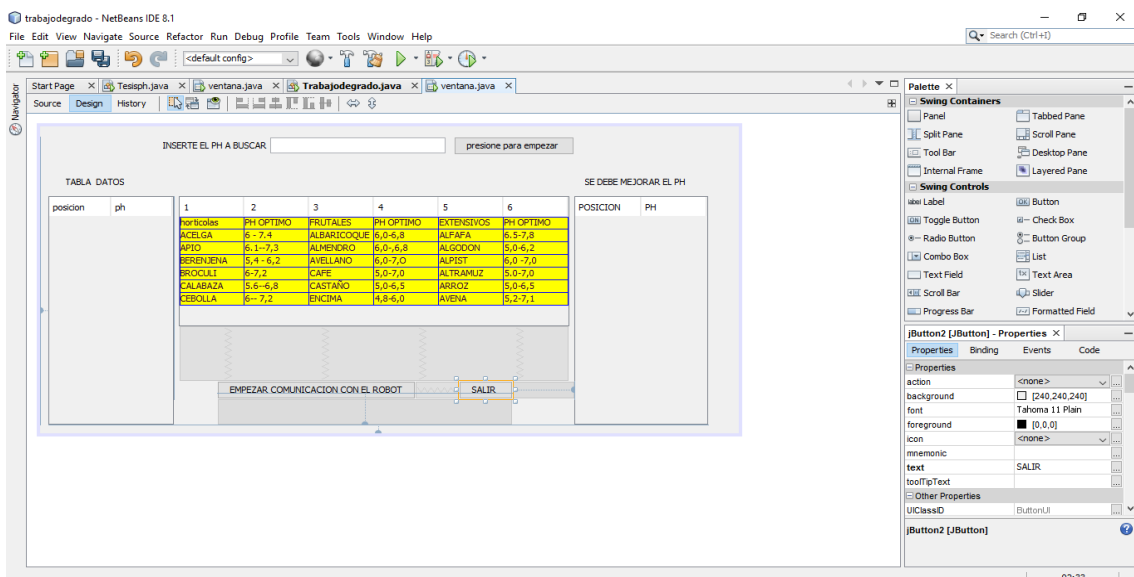


Figura 34 ventana NetBeans creación del JFrame

Fuente: imagen realizada por Autor

3. importamos la librería PanamaHitek_Arduino v2.7.0 Biblioteca creado por Antonio García González Estudiante de la Universidad Tecnológica de Panamá y el creador de panamahitek.com



PanamaHitek_Arduino v2.7.0.jar

Figura 35 librería panamahitek

Fuente: imagen realizada por Autor

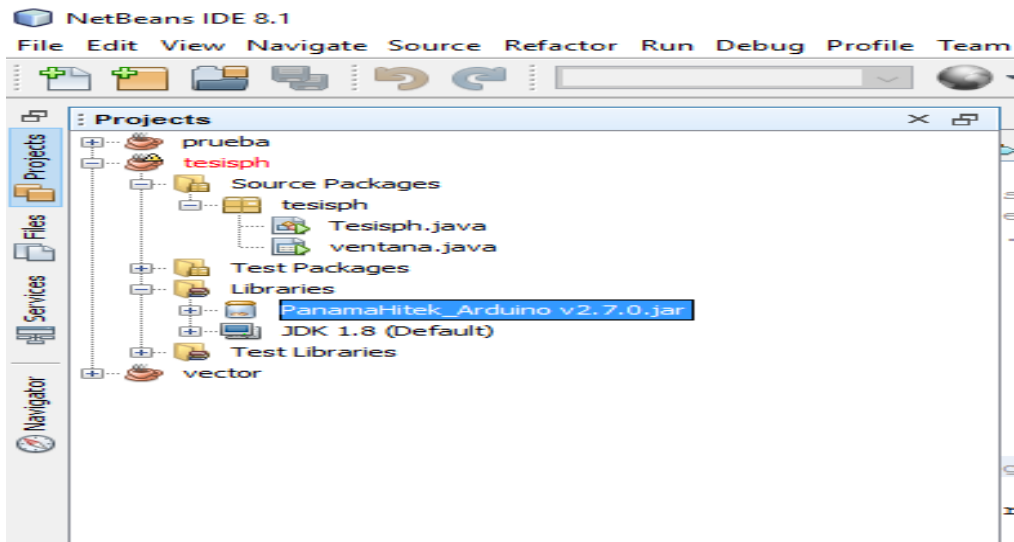


Figura 36 importación de librería panamahitek

Fuente: imagen realizada por Autor

4. Empezamos a programar el código de la interfaz en el JFrame ventana

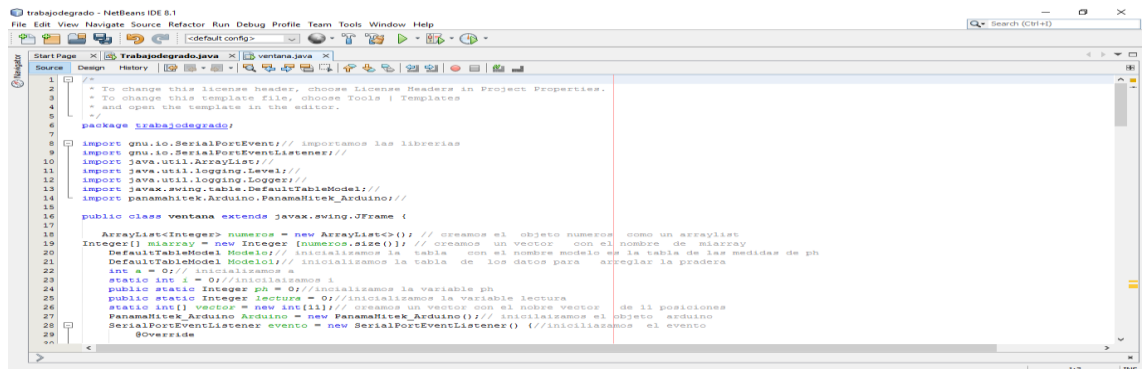


Figura 37 ventana JFrame se empieza a realizar el código
Fuente: imagen realizada por Autor

5. Código final de la programación

6. Código de la ventana principal trabajodegrado.java

```
package trabajodegrado;
```

```
public class Tesisph {  
    public static void main(String[] args) {  
        ventana ventana=new ventana(); //creamos el objeto ventana  
        ventana.setVisible(true); // decimos que sea visible  
        ventana.setTitle("medidas ph");// damos nombre a la ventana  
        ventana.setLocationRelativeTo(null); lo localización de la ventana al ejecutarse  
    }  
}
```

7. Imágenes de ejecución del código final



Figura 38 ventana JFrame se empieza a ejecutar el código

Fuente: imagen realizada por Autor



Figura 39 ventana JFrame código ejecutado

Fuente: imagen realizada por Autor

CONCLUSIONES

1. Podemos concluir que de acuerdo a la dureza del suelo donde se toma la muestra de pH facilita o dificulta la introducción del pH metro al suelo
2. El tamaño, forma y material de las ruedas nos permite un mejor agarre y posicionamiento en el terreno
3. La utilización de los servos de giro continuo nos dificulto un poco el posicionamiento del prototipo porque no nos daba un giro en una sola dirección (modificar el potenciómetro de los servos)
4. Para que el prototipo girara 180 grados para que iniciara cada metro nos tocó medir el tiempo y la velocidad con el cual giraba se realizó con motores dc de 5v
5. El software arduino nos dio facilidad de programación para ejecutar la tarea del prototipo
6. La librería hilteck_panama nos ayudó en la comunicación entre la tarjeta arduino mega y el software NetBeans para realizar la interface grafica

Proyección Futuras.

Mejorar el sistema de posicionamiento del prototipo por medio de GPS o sensores; y establecer una comunicación en tiempo real entre el prototipo y el pc

BIBLIOGRAFIA

<http://aplicacionesdeloselectro2.blogspot.com.co>(Sequeda M & Orozco D 2006)

<https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

ANEXOS

a. Código de programación de la tarjeta Arduino

```
#incluye <Sermo.>
#incluye <Eeprom> //Librería para controlar la EEPROM de la Arduino
flota kph; //variable donde se va a guardar los datos del sensor
Servo serbio;// iniciamos el 1 servo de la rueda izquierda
Servo servidor;// iniciamos el 2 servo de la rueda derecha
Servo servo pH;// iniciamos el 3 servo del sensor medidor de pH
int a=0; //
int b=0;//
int der=6;// iniciamos la variable der en el pin 6 de la Arduino
int izq=5;// iniciamos la variable der en el pin 5 de la arduino
int pinph =2;// iniciamos la variable del sensor de pH
int c=0;//
int w=0;
int umemoria=0; // ubicacion de la memoria eeprom
void setup()
{
  Serial.begin(9600);// comunicacion serial
  servoizq.attach(10);// ubicamos servo izquierdo en el pin 10 de la arduino
  servoder.attach(8); // ubicamos servo derecho en el pin 8 de la arduino
  pinMode(izq, OUTPUT);// colocamos como salida la variable izq del motor dc
  pinMode (pinph,INPUT );
  pinMode(der,OUTPUT); //colocamos como salida la variable izq del motor dc
  servoph.attach(7);// ubicamos servo del sensor en el pin 7 de la arduino
}

void loop()
{
```



```
If ( w<20 )
{
  //ciclo de medicion y adquisicion de datos
  digitalWrite(der,LOW); //colocamos el motor dc en bajo para que no gire ala
derecha y se mantenga quieto
  digitalWrite(izq,LOW);//colocamos el motor dc en bajo para que no gire ala
izquierda y se mantenga quieto

  servoder.write(90);servo derecho apagado
  servoizq.write(90); servo izquierdo apagado
  delay(1000); esperamos 1 segundo

  servoph.write(180); servo del medidor de ph prende y baja a tomar la muestra

  sph = analogRead(pinph); //Toma los datos del sensor

  // guardamos los datos adquiridos en la memoria eeprom
  Serial.println((byte)sph); //Envía los datos al computador mediante el puerto serial
  EEPROM.write(umemoria, sph );
  umemoria = umemoria + 1; //Por cada dato recibido lo guarda en una de los 399 espacios
disponibles en la EEPROM
  if (umemoria == 399) //Sí el número total de datos excede los 512
    umemoria = 0; //Borrar todo y comenzar de nuevo

  delay(6000); //Espera 6 segundos antes de enviar y almacenar la información en la
EEPROM y enviarla al computador
  digitalWrite(der,LOW); // motor dc apagado
  digitalWrite(izq,LOW);//motor dc apagado

  servoder.write(90); //mantenemos los servos quietos mientras guarda los datos
  servoizq.write(90);//mantenemos los servos quietos mientras guarda los datos
  delay (5000);// esperamos 5 segundos

  digitalWrite(der,LOW);//mantenemos el motor dc mientras sube el phmetro
```

```
digitalWrite(izq,LOW);//mantenemos el motor dc mientras sube el phmetro
servoph.write(0);//subimos el ph metro
servoder.write(90);// /mantenemos los servos quietos mientras guarda los datos
servoizq.write(90);//mantenemos los servos quietos mientras guarda los datos
delay(5000);// esperamos 5 segundos

a=a+1; // le agregamos 1 a la variable a
b=b+1;// aumentamos en 1 variable b
Serial.println(b);// imprimimos en pantalla la variable b
Serial.println(a);// imprimimos en pantalla la variable b
Serial.println(c);// imprimimos en pantalla la variable b

// ciclo de movimiento del robot para parar cada metro
if (a<20) //el ciclo se realiza 20 veces
{

servoph.write(0);// servo del sensor de ph apagado
servoder.write(120);// servo derecho con una velocidad de 30
servoizq.write(60);//servo izquierdo con una velocidad de 30
delay (2000);// lo realiza durante 2 segundos
}
// giro de 90 grados la traccion delantera
if(b==21) // cuando termina de hacer el recorrido realiza el giro hacia la izquierda
{digitalWrite(der,LOW ); se apaga el giro hacia la derecha
digitalWrite(izq,HIGH); // prendemos el giro hacia la derecha
servoder.write(120); // servo derecho lo encendemos con una velocidad de 30
servoizq.write(60); // servo izquierda lo encendemos con una velocidad de 30
delay (3000); esperamos 3 segundos

c=1; //colocamos la variable c en 1
}

if(c==1)
{
```

```
digitalWrite(der,HIGH ); //prendemos el motor dc para que nos gire hacia la derecha
digitalWrite(izq,LOW); // mantenemos el pulso del motor apagado
a=0; variable a 0
b=0;variable a 0
c=0;variable a 0
delay(2000); esperamos 2 segundos

w=20;
}
}

Else

//ciclo de medicion y adquision de datos

digitalWrite(der,LOW); //colocamos el motor dc en bajo para que no gire ala derecha
y se mantenga quieto
digitalWrite(izq,LOW);//colocamos el motor dc en bajo para que no gire ala izquierda
y se mantenga quieto

servoder.write(90);servo derecho apagado
servoizq.write(90); servo izquierdo apagado
delay(1000); esperamos 1 segundo

servoph.write(180); servo del medidor de ph prende y baja a tomar la muestra

sph = analogRead(pinph); //Toma los datos del sensor

// guardamos los datos adquiridos en la memoria emprom
Serial.println((byte)sph); //Envía los datos al computador mediante el puerto serial
EEPROM.write(umemoria, sph );
umemoria = umemoria + 1; //Por cada dato recibido lo guarda en una de los 399 espacios
disponibles en la EEPROM
if (umemoria == 399) //Sí el número total de datos excede los 512
```

```
umemoria = 0; //Borrar todo y comenzar de nuevo

delay(6000); //Espera 6 segundos antes de enviar y almacenar la información en la
EEPROM y enviarla al computador
digitalWrite(der,LOW); // motor dc apagado
digitalWrite(izq,LOW);//motor dc apagado

servoder.write(90); //mantenemos los servos quietos mientras guarda los datos
servoizq.write(90);//mantenemos los servos quietos mientras guarda los datos
delay (5000);// esperamos 5 segundos

digitalWrite(der,LOW);//mantenemos el motor dc mientras sube el phmetro
digitalWrite(izq,LOW);//mantenemos el motor dc mientras sube el phmetro
servoph.write(0);//subimos el ph metro
servoder.write(90);// /mantenemos los servos quietos mientras guarda los datos
servoizq.write(90);//mantenemos los servos quietos mientras guarda los datos

delay(5000);// esperamos 5 segundos

a=a+1; // le agragamos 1 a la variable a

b=b+1;// aumentamos en 1 variable b

Serial.println(b);// imprimimos en pantalla la variable b

Serial.println(a);// imprimimos en pantalla la variable b

Serial.println(c);// imprimimos en pantalla la variable b

// ciclo de movimiento del robot para parar cada metro

if (a<20) //el ciclo se realiza 20 veces

{

servoph.write(0);// servo del sensor de ph apagado
servoder.write(120);// servo derecho con una velocidad de 30
servoizq.write(60);//servo izquierdo con una velocidad de 30
```

```
delay (2000);// lo realiza durante 2 segundos
}
// giro de 90 grados la traccion delantera
if(b==21) // cuando termina de hacer el recorrido realiza el giro hacia la izquierda
{digitalWrite(der,HIGH); se prende el giro hacia la derecha
digitalWrite(izq,LOW); // apagamos el giro hacia la derecha
servoder.write(120); // servo derecho lo encendemos con una velocidad de 30
servoizq.write(60); // servo izquierda lo encendemos con una velocidad de 30
delay (3000); esperamos 3 segundos
c=1; //colocamos la variable c en 1
}
if(c==1)
{
digitalWrite(der,LOW); //apagamos el motor dc para que nos gire hacia la derecha
digitalWrite(izq, HIGH); // prendemos el motor para que gire ala izquierda
a=0; variable a 0
b=0;variable a 0
c=0;variable a 0
delay(2000); esperamos 2 segundos
w=0;
}
}
```

b. Código de programación Software NetBeans

```
package trabajodegrado;
/* importamos las librerias */
import gnu.io.SerialPortEvent;// importamos las librerias
import gnu.io.SerialPortEventListener;//
import java.util.ArrayList;//
import java.util.logging.Level;//
import java.util.logging.Logger;//
import javax.swing.table.DefaultTableModel;//
```

```
import panamahitek.Arduino.PanamaHitek_Arduino;//

public class ventana extends javax.swing.JFrame {
    ArrayList<Integer> numeros = new ArrayList<Integer>(); // creamos el objeto numeros
como un arraylist
    Integer[] miarray = new Integer [numeros.size()]; // creamos un vector con el nombre de
miarray
    DefaultTableModel Modelo;// inicializamos la tabla con el nombre modelo es la tabla de
las medidas de ph
    DefaultTableModel Modelo1;// inicializamos la tabla de los datos para arreglar la
pradera
    int a = 0;// inicializamos a
    static int i = 0;//inicilaizamos i
    public static Integer ph = 0;//inicializamos la variable ph
    public static Integer lectura = 0;//inicializamos la variable lectura
    static int[] vector = new int[11];// creamos un vector con el nobre vector de 11 posiciones
    PanamaHitek_Arduino Arduino = new PanamaHitek_Arduino();// inicilaizamos el objeto
arduino
    SerialPortEventListener evento = new SerialPortEventListener() { //iniciliazamos el evento
@Override

    public void serialEvent(SerialPortEvent spe) { // inicializamos el metodo serial event
pregunta si ahy una conexion entre el serial de la tarjeta arduino y el pc

    if (Arduino.isMessageAvailable() == true) { // ppreguntamos si hay mensajes si si
mandamos el dato

    ph = Integer.parseInt(Arduino.printMessage());// la variable ph toma el valor que envia el
puerto serial de la tarjeta arduino

    numeros.add(ph); //llenamos el vector numeros con los datos de ph
```

```
miarray = numeros.toArray(miarray);// convertimos los datos de arraylist a un array
llamado mi array
lectura = lectura + 1; //aumentamos el valor de uno a lectura

if(lectura==11)//preguntamos si la variable lectura es igual a 400
{
System.out.println(numeros); //imprimimos el arraylist numeros

}

if (lectura >1) { //cuando lectura sea mayor a uno ejecutamos el metodo tableupdate
tableupdate();//

}
}
};

public void tableupdate() { // metodo tableupdate
Modelo.addRow(new Object[]{lectura, ph}); // agregamos a la jtable modelo los valores
de lectura y ph
}
public void buscar (){// metodo buscar
a = Integer.parseInt(this.buscar.getText());//le damos el valor a la variable a por medio de
el get text

System.out.println(a);// imprimimos a
}
public void vectorf (){ // metodo vectorf
for (i=0; i<miarray.length; i++) // inicializamos el ciclo for
{
if (miarray[i] != a) //preguntamos si los datos del arreglo mi array
{
```

```
vector[i]=miarray[i]; // guadamos los datos de mi array en vector

Modelo1.addRow(new Object[]{i, vector[i]}); // agregamos a la tabla 2 los valores del nuevo
vector
}
}

for(Integer s : vector) // recorreremos el vector con nombre vector y le damos el nombre de s
{

System.out.print(s+ " "); // imprimimos el vector
}
}

/*
 * Creates new form ventana
 */
public ventana() {
    initComponents();

    try {
        Arduino.arduinoRXTX("COM3", 9600, evento); // comunicacion entre arduino y java
        // se recibe por el puerto 3 a 9600 baudios y lo recibe como un evento
    } catch (Exception ex) {
        Logger.getLogger(ventana.class.getName()).log(Level.SEVERE, null, ex);
    }

    Modelo = (DefaultTableModel) jTable1.getModel(); // guardamos en modelo la jTable 1
    tabla de datos del robot

    Modelo1 = (DefaultTableModel) jTable3.getModel(); // guardamos en modelo1 la jTable 3
    tabla de datos de busqueda del ph
}

/**
 * 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() {
```

```
    jScrollPane1 = new javax.swing.JScrollPane();
```

```
    jTable1 = new javax.swing.JTable();
```

```
    jButton1empezar = new javax.swing.JButton();
```

```
    jButton3salir = new javax.swing.JButton();
```

```
    jScrollPane2 = new javax.swing.JScrollPane();
```

```
    jTable2 = new javax.swing.JTable();
```

```
    buscar = new javax.swing.JTextField();
```

```
    bbuscar = new javax.swing.JButton();
```

```
    jScrollPane3 = new javax.swing.JScrollPane();
```

```
    jTable3 = new javax.swing.JTable();
```

```
    jLabel1 = new javax.swing.JLabel();
```

```
    jLabel2 = new javax.swing.JLabel();
```

```
    jLabel3 = new javax.swing.JLabel();
```

```
    jLabel4 = new javax.swing.JLabel();
```

```
    jLabel5 = new javax.swing.JLabel();
```

```
    jLabel6 = new javax.swing.JLabel();
```

```
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

```
    jTable1.setAutoCreateRowSorter(true);
```

```
    jTable1.setBackground(new java.awt.Color(204, 255, 204));
```

```
    jTable1.setModel(new javax.swing.table.DefaultTableModel(  
    new Object [][] {
```

```
    },
```

```
    new String [] {
```

```
        "posicion", "ph"
```

```
}
));
jTable1.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_NEXT_COLUMN);
jTable1.setEnabled(false);
jTable1.setGridColor(new java.awt.Color(153, 153, 255));
jTable1.setRowHeight(15);
jScrollPane1.setViewportView(jTable1);

jButton1empezar.setText("empezar");
jButton1empezar.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton1empezarActionPerformed(evt);
}
});

jButton3salir.setText("salir");
jButton3salir.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton3salirActionPerformed(evt);
}
});

jTable2.setAutoCreateRowSorter(true);
jTable2.setBackground(new java.awt.Color(102, 255, 51));
jTable2.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
{"hortícolas", "PH OPTIMO", "FRUTALES", "PH OPTIMO", "EXTENSIVOS", "PH
OPTIMO"},
{"ACELGA", "6 - 7,4", "ALBARICOQUE", "6,0-6,8", "ALFAFA", "6.5-7,8"},
{"APIO", "6.1--7,3", "ALMENDRO", "6,0-,6,8", "ALGODON", "5,0-6,2"},
{"BERENJENA", "5,4 - 6,2", "AVELLANO", "6,0-7,0", "ALPIST", "6,0 -7,0"},
{"BROCOLI", "6-7,2", "CAFE", "5,0-7,0", "ALTRAMUZ", "5.0-7,0"},
{"CALABAZA", "5.6--6,8", "CASTAÑO", "5,0-6,5", "ARROZ", "5,0-6,5"},
{"CEBOLLA", "6-- 7,2", "ENCIMA", "4,8-6,0", "AVENA", "5,2-7,1"}
}
```

```
    },  
    new String [] {  
        "1", " 2", "3", "4", "5", "6"  
    }  
    ));  
jTable2.setGridColor(new java.awt.Color(51, 51, 255));  
jTable2.setSelectionForeground(new java.awt.Color(51, 51, 255));  
jScrollPane2.setViewportView(jTable2);  
  
bbuscar.setText("presione para empezar ");  
bbuscar.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        bbuscarActionPerformed(evt);  
    }  
});  
  
jTable3.setModel(new javax.swing.table.DefaultTableModel(  
    new Object [][] {  
  
    },  
    new String [] {  
        "POSICION", "PH"  
    }  
    ));  
jScrollPane3.setViewportView(jTable3);  
  
jLabel1.setText("TABLA DATOS ");  
  
jLabel2.setText("TABLA DATOS  ");  
  
jLabel3.setText("DONDE SE DEBE MODIFICAR LA PRADERA");  
  
jLabel4.setText("TOMADOS POR EL ROBOT");
```



```
.addComponent(jButton1empezar)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(jButton3salir)
.addGap(108, 108, 108))
.addGroup(layout.createSequentialGroup())
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup())
.addGap(52, 52, 52)
.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,470,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(layout.createSequentialGroup())
.addGap(202, 202, 202)
.addComponent(jLabel6)))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 56,
Short.MAX_VALUE)))
.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE,194,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap()))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
.addGap(26, 26, 26)
.addComponent(jLabel4)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(jLabel3)
.addContainerGap())
);
layout.setVerticalGroup(
layout.createSequentialGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addContainerGap()
.addGroup(layout.createSequentialGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
.addGap(0, 0, Short.MAX_VALUE)
.addComponent(jLabel1)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jLabel4)
.addGap(18, 18, 18)
.addGroup(layout.createSequentialGroup()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(buscar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(bbuscar)
.addComponent(jLabel5))
.addGroup(layout.createSequentialGroup()
.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE,33,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jLabel3)))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 17,
Short.MAX_VALUE)))
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
.addComponent(jLabel6)
.addGap(18, 18, 18)
.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 164,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(39, 39, 39)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jButton3salir)
.addComponent(jButton1empezar))
.addGap(22, 22, 22))
```

```
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,395,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE,395,
javax.swing.GroupLayout.PREFERRED_SIZE)))
.addGap(0, 0, Short.MAX_VALUE)
);

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

```
private void jButton1empezarActionPerformed(java.awt.event.ActionEvent evt) {

try {
    Arduino.sendData("1"); // enviamos un 1 con el boton epear para comenzar la
transmision de datos entre la placa arduino y java
    // TODO add your handling code here:
} catch (Exception ex) {
    Logger.getLogger(ventana.class.getName()).log(Level.SEVERE, null, ex);
}

}
```

```
private void jButton3salirActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
System.exit(0); //si oprrios el boton salir nos cierra la ventana
}
```

```
private void bbuscarActionPerformed(java.awt.event.ActionEvent evt) {

buscar(); // se ejecuta el metodo buscar al pulsar el boton presione para empezar
vectorf();// se ejecuta el metodo vector al pulsar el boton presione para empezar
```

```
// TODO add your handling code here:
}

/**
 * @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())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(ventana.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(ventana.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(ventana.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(ventana.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    }
}
```



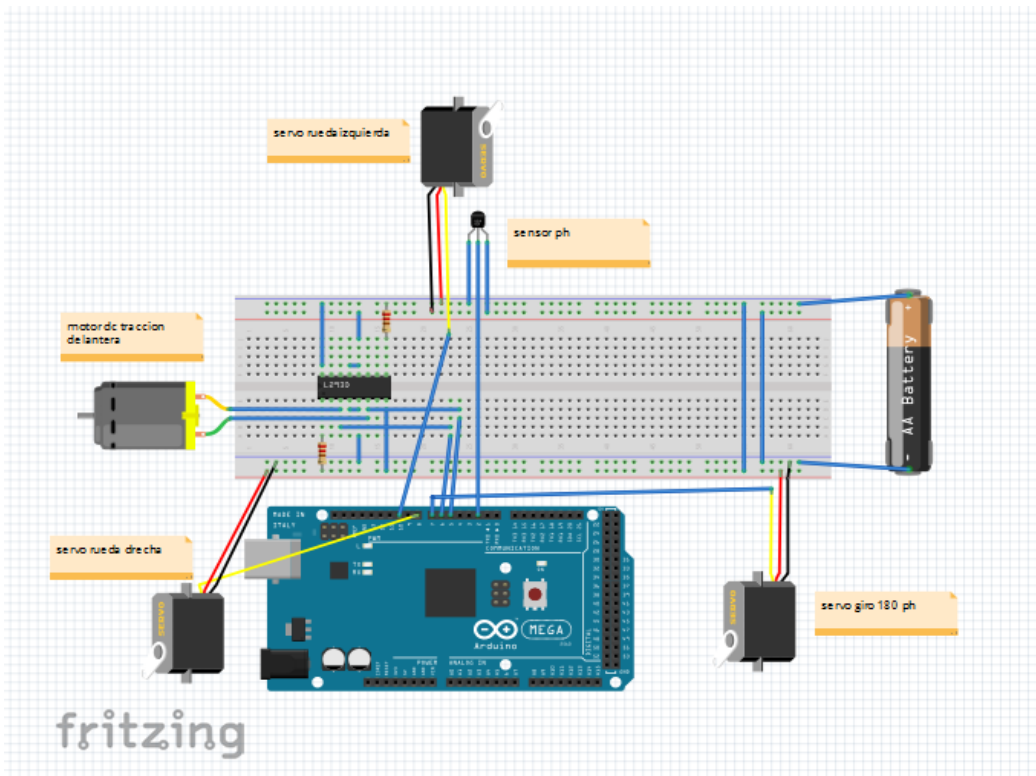
```
//</editor-fold>
```

```
/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new ventana().setVisible(true);// hacemos visible el JFrame ventana

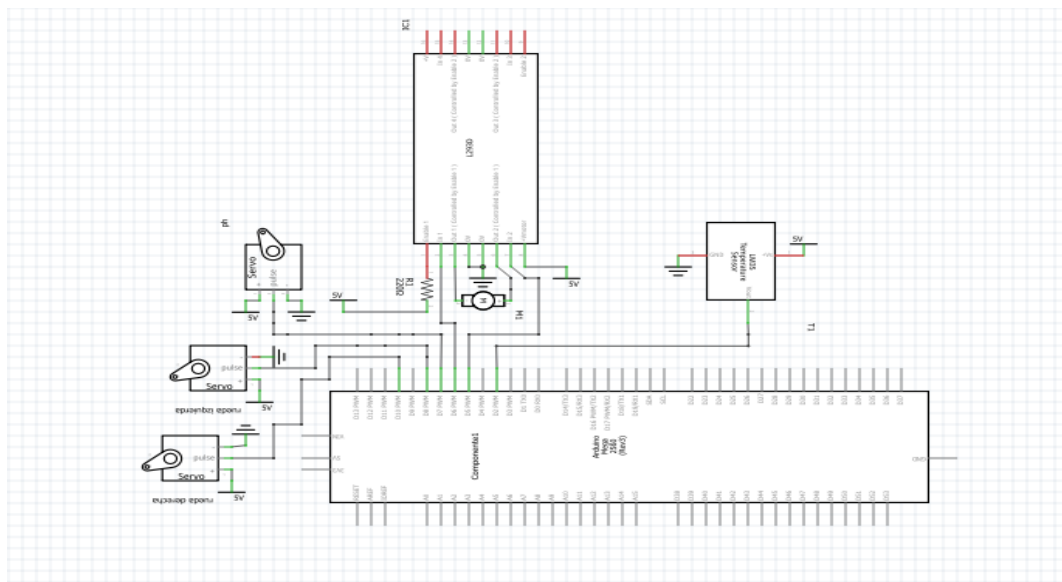
}
});
}

// Variables declaration - do not modify
private javax.swing.JButton bBuscar;
private javax.swing.JTextField buscar;
private javax.swing.JButton jButton1Empezar;
private javax.swing.JButton jButton3Salir;
private javax.swing.JLabel jLabel1;
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.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JTable jTable1;
private javax.swing.JTable jTable2;
private javax.swing.JTable jTable3;
// End of variables declaration
}
```

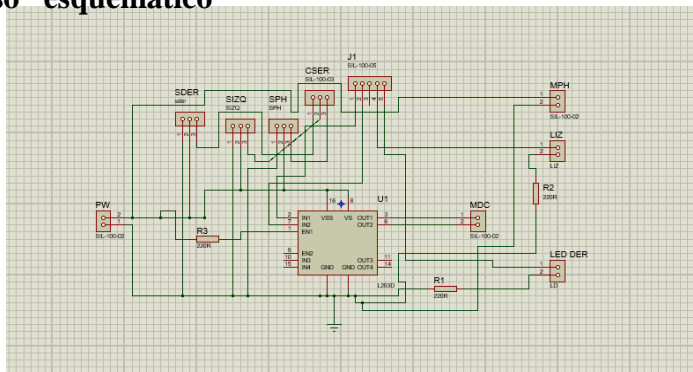
c. **Circuito en Protoboard.**



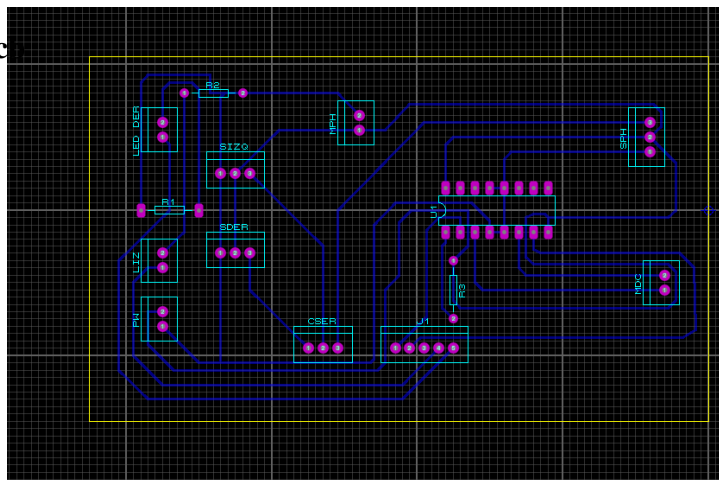
Circuito esquemático



Circuito impreso esquemático



Circuito impreso pcb



elaborado en proteus

Circuito impreso pcb

