TRABAJO DE
GRADO PARA OPTAR
POR EL TÍTULO DE
INGENIERO ELECTRÓNICO

PROTOTIPO ROBÓTICO RECOLECTOR DE OBJETOS BASADO EN EL TRATAMIENTO DE IMÁGENES Y LÓGICA DIFUSA EN MATLAB

POR:

GERALDINE ESTHER STAN HERNANDEZ
LEONARDO ANDRES PERDOMO DURAN

DIRECTOR: JOSE RAMON IGLESIAS

MAYO 2014

PROTOTIPO ROBÓTICO RECOLECTOR DE OBJETOS BASADO EN EL TRATAMIENTO DE IMÁGENES Y LÓGICA DIFUSA EN MATLAB

LEONARDO ANDRES PERDOMO DURÁN
2008119041
GERALDINE ESTHER STAN HERNÁNDEZ
2008119057

DIRECTOR:
M.SC. ING. JOSÉ RAMÓN IGLESIAS GAMARRA PH.D.(C)

UNIVERSIDAD DEL MAGDALENA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA

SANTA MARTA D.T.C.H.
MAYO 2, 2014

NOTA DE ACEPTACIÓN

_____

_____

_____

_____
PRESIDENTE DEL JURADO

_____
JURADO

_____
JURADO

Santa Marta, _____

Dedicatoria

Quiero dedicarles esta investigación a todas aquellas personas que creyeron en mi capacidad de salir adelante.

A Dios por llenarme de paciencia y sabiduría en cada momento de esta etapa de mi vida.

A mis padres, por inculcarme los valores para ser una buena persona, por brindarme su apoyo, comprensión y confianza en todo momento.

A mis sobrinas por ser la alegría de mi hogar y demostrarme que las personas llevan siempre un niño en su interior.

A Daniela, compañera incondicional en mi vida, bella persona, por estar en los momentos más difíciles y brindarme los mejores consejos para tomar decisiones.

A ellos este proyecto, que sin ellos, no hubiese podido ser.

Leonardo Andrés Perdomo Durán

Dedicatoria

Dedico este proyecto de tesis primeramente a Dios; porque desde antes de nacer ya me protegía y ha permanecido conmigo en cada paso que doy, cuidándome y dándome fortaleza para continuar.

A mi familia, en especial mis padres, María y Cosme, quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento. Depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y capacidad. Es por ellos que soy lo que soy ahora y espero seguir superándome para algún día poder devolverles un poco de lo mucho que me han dado.

A Jhormmy, por ser un gran hombre, que sin necesidad de solicitarle siempre me acompaño y apoyo, brindándome su invaluable atención y amor.

Geraldine Esther Stan Hernández

Agradecimientos

Los autores de este proyecto le agradecemos a:

La Universidad del Magdalena, nuestra alma mater, por otorgarnos una formación académica de alta calidad y brindarnos las herramientas para crecer tanto profesional y personalmente.

Nuestros docentes quienes nos sirvieron como guías para aprender las técnicas de la profesión más hermosa de la vida, la Ingeniería.

José Ramón Iglesias Ingeniero Electrónico, docente de la Universidad del Magdalena y director de este trabajo de grado, por brindarnos su apoyo a la hora de emprender el proyecto y sacarlo adelante con sus recomendaciones y observaciones.

Índice General

Índice General..... 7
Índice de Tablas..... 9
Índice de Figuras..... 10
Resumen..... 11
1. Descripción del Proyecto..... 12
1.1. Planteamiento del Problema..... 12
1.2. Justificación..... 14
1.3. Alcances y Limitaciones..... 15
1.4. Objetivos..... 16
1.4.1. Objetivo General..... 16
1.4.2. Objetivos Específicos..... 16
1.5. Estado del Arte..... 17
1.6. Marco teórico conceptual..... 18
1.6.1. Cámara inalámbrica..... 18
1.6.2. Comunicación por bluetooth..... 19
1.6.3. Procesamiento de las imágenes..... 24
1.6.4. Pruebas iniciales..... 28
1.6.5. Lógica Difusa..... 30
1.6.6. Interfaz gráfica..... 33
1.6.7. Plataforma Rover 5..... 38
1.6.8. Firmware de Arduino..... 38
1.6.9. Board control de los motores..... 43
1.6.10. Sensor de distancia..... 44
1.6.11. Pinza para recolectar los objetos..... 45
1.6.12. Alimentación del prototipo..... 46
1.7. Diseño metodológico..... 47
1.7.1. Tipo de investigación..... 47
1.7.2. Métodos de investigación..... 48
1.7.3. Recolección de la información..... 49
1.8. Cronograma..... 51

1.9. Resultados..... 52
1.9.1. Productos y Logros..... 52
1.9.2. Efectos (Impacto)..... 52
1.9.3. Recomendaciones..... 53
2. Presupuesto..... 54
2.1. Recursos Humanos..... 54
2.2. Recursos Institucionales..... 54
2.3. Recursos Adicionales..... 55
2.4. Costo total del proyecto..... 56
3. Referencias..... 57
3.1. Bibliografía y Webgrafía..... 57
4. Anexos..... 59
4.1. Anexo A: Datasheet Cámara inalámbrica..... 59
4.2. Anexo B: Datasheet bluetooth..... 60
4.3. Anexo C: Especificaciones de Arduino Uno..... 61
4.4. Anexo D: Datasheet Ardumoto..... 62
4.5. Anexo E: Datasheet Sensor SRF05..... 64
4.6. Anexo F: Datasheet Servomotor..... 65

Índice de Tablas

Tabla 1 – Cronograma.....51
Tabla 2 – Costos de los Componentes.....55
Tabla 3 – Costo total del proyecto.....56

Índice de Figuras

Figura 1 – Icono de Bluetooth....20
Figura 2 – Agregar dispositivo Bluetooth....20
Figura 3 – Elegir un dispositivo....21
Figura 4 – Escribir código del dispositivo....21
Figura 5 – Instalando MercuryOne Bluetooth....22
Figura 6 – Comunicación establecida....23
Figura 7 – Error al conectarse con el robot....23
Figura 8 – BlueSmirf Silver....24
Figura 9 – Cámara encendida....28
Figura 10 – Objeto detectado....29
Figura 11 – Robot detectando el objeto de color rojo....29
Figura 12 - Fuzzy Inference System....30
Figura 13 – Funciones de Membresía....31
Figura 14 - Editor de Reglas....32
Figura 15 - Prueba de Mercuryfuzzy....33
Figura 16 – Interfaz del robot MercuryOne....34
Figura 17 – Interfaz Gráfica, Comunicación establecida....35
Figura 18 – Interfaz Gráfica, Cámara encendida....36
Figura 19 – Interfaz Gráfica, Imágenes procesadas y objeto detectado....36
Figura 20 – Desconectando el robot....37
Figura 21 – Objeto recolectado por el robot....37
Figura 22 – Plataforma Rover 5....38
Figura 23 – Placa Arduino Uno....39
Figura 24 – Entorno Arduino (IDE)....39
Figura 25 – Shield Ardumoto para Arduino....43
Figura 26 – Shield conectado al Arduino....43
Figura 27 – Sensor de distancia SRF05....44
Figura 28 – Modelo 3D de la pinza....45
Figura 29 – Pinza fabricada....45
Figura 30 – Arreglo de baterías AA....46
Figura 31 – Batería AA....46
Figura 32 – Batería 9V....46

Resumen

En el presente proyecto se describe el diseño y la construcción de un prototipo robótico que reconoce, rastrea y recolecta objetos pequeños mediante la aplicación del tratamiento digital de imágenes y lógica difusa, con ayuda de Image Processing Toolbox y Fuzzy ToolBox de Matlab y de la plataforma Arduino.

Mercury One es el sistema robótico que adquiere e interpreta la información visual del medio para recoger cubos pequeños hechos de cartón paja de distintos colores, en base a teorías en los campos del procesamiento de señales digitales y la robótica.

Dicho sistema cuenta con una interfaz visual amigable, procesamiento de imágenes y lógica difusa desarrollados en Matlab, comunicación inalámbrica por bluetooth, cámara inalámbrica con tecnología RF, sensor de distancia, base firmware desarrollado en Arduino, boards de Arduino Uno y Ardumoto, pinza hecha en madera y un servomotor para recolectar los objetos.

1. Descripción del Proyecto

1.1. Planteamiento del Problema

La ingeniería surge por la necesidad que ha tenido el ser humano a través de los tiempos de buscar la forma eficaz de solucionar los problemas que día a día afronta. Se sabe que el hombre es un ser pensante, por ende sus ideas y proyectos evolucionan, lo que se ve reflejado en las diversas aplicaciones que se desarrollan gracias a la intervención de ésta en los diferentes campos de estudio.

Uno de estos campos es la electrónica, que se basa en la creación, desarrollo e implementación de sistemas de control, monitoreo, automatización y sistematización de variables y tratamiento de datos, que influyen en procesos tecnológicos que buscan la innovación de métodos para mejorar las condiciones de vida de la sociedad. A partir de estos métodos se han obtenido progresos en otros campos; ya que éstos ayudan a resolver de manera rápida determinadas operaciones que resultan tediosas a la hora de elaborarlas manualmente.

Los progresos en la ingeniería electrónica hacen que temas como el tratamiento de imágenes en el área del procesamiento digital de señales y la lógica difusa en robótica, sean importantes, haciendo que se investigue aún más en estas teorías y a su vez sean aplicadas de forma más específicas, como determinar el color o textura de un objeto, o saber a qué distancia se encuentra para recogerlo.

1.2. Justificación

Hoy en día vivimos en un mundo visualmente encantador, que se puede percibir a través de gran variedad de formas, objetos, texturas y colores. El ser humano es capaz de adquirir e interpretar toda esta información visual que lo rodea y así mismo es un reto procesar y transmitir esas capacidades a una computadora con el fin de interpretar dicha información visual.

Procesar esta información a través de software es una forma efectiva y automática para obtener mejores resultados, aparte de facilitar una digitalización de la señal obtenida.

Por otra parte la lógica difusa es útil cuando la complejidad de un proceso a estudiar es muy alta o no existen procesos matemáticos precisos, con el propósito de cambiar el funcionamiento o el estado del proceso en cuestión. Por esto, con la lógica difusa es posible controlar un proceso por medio de reglas de sentido común, las cuales se refieren a cantidades indefinidas.

Teniendo en cuenta esto, es preciso realizar un estudio de los campos mencionados, donde se refleje una investigación a fondo y beneficie a los autores de manera que puedan profundizar sus conocimientos.

Cabe aclarar que existen varias formas de llevar a cabo una investigación, y que para ello se necesita tiempo, dedicación y recursos; además de motivación por el tema trabajado. Por ello, es conveniente trabajar con un prototipo robótico; dado que el panorama actual de la robótica es muy variado, y en el mercado hay muchas opciones para escoger, desde kits para armar plataformas rodantes hasta manuales de como armar robots caseros con piezas de computadoras.

Además, el tratamiento de imágenes, la lógica difusa y la construcción física de robots permite estudiar de manera interesante las teorías de procesamiento de señales digitales y la robótica.

1.3. Alcances y Limitaciones

Con este proyecto se afianzó el conocimiento de los autores en los temas de procesamiento digital de imágenes, lógica difusa, comunicación inalámbrica por bluetooth, robótica y la forma de implementarlos.

Se exploró las formas de reconocimiento de objetos a través de una cámara inalámbrica y el uso de la lógica difusa para determinar de forma más accesible a que distancia está el objeto reconocido y a qué velocidad debe moverse la plataforma robótica para recoger tal objeto.

Por otra parte, debido a la falta de tiempo, dado a que ambos desarrolladores de este trabajo de grado se encontraban en su periodo de prácticas profesionales, la interfaz gráfica no pudo realizarse en lenguaje Java.

El reconocimiento de los objetos se limitó a tres colores (verde, azul y rojo) dado que la gama de colores es muy amplia y el trabajo debía tener objetivos alcanzables.

Por falta de presupuesto, partes del prototipo se fabricaron en madera, las cuales poseen la cualidad de ser resistente, se escogió comunicación vía bluetooth por encima de módulos Xbee que poseen mayor alcance de transmisión pero son más costosos.

1.4. Objetivos

1.4.1. Objetivo General: Diseñar un prototipo robótico que reconozca, rastreé y recolecte objetos basado en tratamiento de imágenes digitales y lógica difusa en MATLAB.

1.4.2. Objetivos Específicos:

- ✓ Escribir un script en Matlab para el procesamiento de imágenes.
- ✓ Construir una plataforma robótica donde estén ubicados todos los componentes del hardware.
- ✓ Obtener imágenes en tiempo real por medio de una cámara inalámbrica ubicada en la plataforma robótica.
- ✓ Realizar un mecanismo capaz de recolectar objetos pequeños (cubos de cartón paja de 4 cm de arista).
- ✓ Desarrollar un sistema de control con una placa de Arduino para integrar los módulos en la plataforma robótica.
- ✓ Establecer una comunicación inalámbrica entre el computador y el Arduino.
- ✓ Diseñar una interfaz visual amigable para que el usuario pueda escoger el color (rojo, azul, verde) del objeto e interactuar con el sistema fácilmente.
- ✓ Elaborar un manual técnico con las especificaciones del prototipo.

1.5. Estado del Arte

Con el paso del tiempo la robótica ha ido desarrollándose vertiginosamente, tanto así que los investigadores han desarrollado algoritmos robustos para que los robots adopten funciones como las que cumplen los humanos en la vida diaria. Estos robots son capaces de realizar estas tareas gracias a esos algoritmos basados en inteligencia artificial que hace que éste vaya aprendiendo de forma autónoma.

En su tesis de máster *Algoritmo de reconocimiento de forma y color para una plataforma robótica*, Jon Aristondo Etxeberria desarrolló un estudio de teorías tales como la Estimadores de posición, sistemas de navegación inercial, posicionamiento basado en mapas, formas de objetos en imágenes y color tracking, para llevar a cabo ésta, en la cual él desarrolla una plataforma robótica que cumple con la función de reconocer formas y colores de los objetos.

Con ayuda de estas teorías estudiadas, se puede plantear un algoritmo de inteligencia artificial, en el que el robot vaya detectando y reconociendo patrones, y guardando y asimilando cambios de entorno.

En la metodología de su tesis, Jon Aristondo Etxeberria planteó que el objetivo principal del tratamiento de la imagen capturada era identificar el objeto y determinar cuál era el punto medio en coordenadas. Y los pasos a seguir eran: captura de imagen, tratamiento de color, detección de bordes, reconocimiento de formas y control del robot, repitiendo cíclicamente.

Por otro lado, Fernando Quirós en su tesis *Navegación robótica basada en aprendizaje evolutivo de acciones mediante lógica difusa*, se apoyó en teorías como redes neuronales artificiales, algoritmos genéticos y lógica difusa, donde desarrolla un robot el cual es entrenado a base de las redes neuronales para la navegación de espacios o habitaciones Cabe destacar que los algoritmos desarrollados en las dos tesis mencionadas anteriormente, tratan en todo momento de darle autonomía al robot y hacer uso de inteligencia artificial, tal cual como se debe investigar en esta tesis para poder combinar técnicas y desarrollar una plataforma robótica que rastree y recoja objetos.

1.6. Marco Teórico Conceptual

1.6.1. Cámara inalámbrica

El robot posee una cámara inalámbrica, la cual funciona con tecnología RF a frecuencias de 2.4Ghz, tiene una resolución de 720x480 en formato YUY2, ésta es alimentada por una batería de 9V y es fácilmente transportable. Además, contiene un receptor USB que va conectado al PC y de donde se puede acceder a la imagen a través del Matlab.

Para llevar a cabo la adquisición de imágenes es necesario configurar la cámara.

Para configurar la cámara en el Matlab, es necesario crear un objeto el cual se encargue de alojar las características de los adaptadores de imagen que posea el PC que se está utilizando. Se puede saber que adaptadores posee el equipo simplemente con teclear la siguiente línea de código en el comand window de Matlab:

```
>> imaqhwinfo
```

El Matlab arrojará el siguiente tipo respuesta dependiendo de los adaptadores que haya instalado:

```
ans =

    InstalledAdaptors: {'gentl'  'gige'  'matrox'  'winvideo'}
        MATLABVersion: '8.2 (R2013b)'
          ToolboxName: 'Image Acquisition Toolbox'
       ToolboxVersion: '4.6 (R2013b)'
```

Para acceder a la información de los adaptadores de video es necesario teclear la siguiente línea de código:

```
>> imaqhwinfo('winvideo')
```

Donde la información arrojada por el Matlab será la siguiente:

```
        AdaptorDllName: [1x81 char]
    AdaptorDllVersion: '4.6 (R2013b)'
          AdaptorName: 'winvideo'
            DeviceIDs: {[1]}
```

```
DeviceInfo: [1x1 struct]
```

El número alojado en DeviceIDs representa la cantidad de adaptadores de video que posee el PC. En este caso solo hay un adaptador de video el cual corresponde a la cámara inalámbrica del prototipo.

El algoritmo completo para iniciar la cámara inalámbrica se observa a continuación:

```
vid = videoinput('winvideo', 1, 'YUY2_720x480');
vid.ReturnedColorspace = 'rgb';
src = getselectedsource(vid);
src.AnalogVideoFormat = 'ntsc_m';
src.Contrast = 153;
src.Brightness = 153;
src.Saturation = 228;
src.Sharpness = 253;
vid.FramesPerTrigger = 1;
vid.TriggerRepeat = inf;
imaqmem(6000000000);
preview(vid);
```

En el cual se inicializa el objeto de video, se le asigna un espacio de color RGB y las propiedades de contraste, saturación, brillo y sombras. Y para terminar se establece un espacio de memoria para alojar las imágenes adquiridas y se muestra un plano previo de lo que observa el lente de la cámara.

Cabe destacar, que para cuestiones de la interfaz gráfica del proyecto y el mejor funcionamiento de ésta, se añadieron líneas de código las cuales están encargadas de informarle al usuario cuando ha habido un error al inicializar la cámara del prototipo, cuando ha sido encendida correctamente o cuando se encendió la cámara del PC.

Con la cámara configurada para ser encendida en la interfaz, se procede a configurar la comunicación inalámbrica entre el PC y el robot.

1.6.2. Comunicación por bluetooth

La comunicación inalámbrica se realiza vía Bluetooth, la cual fue escogida como mejor alternativa dado a los costes del módulo y el alcance en metros (10 mts en línea de vista según el fabricante) del mismo.

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

- Facilitar las comunicaciones entre equipos móviles.
- Eliminar los cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales."[1]

Para establecer la comunicación inalámbrica por primera vez con el modulo bluetooth desde el PC, se procede de la siguiente manera:

Dar click derecho en dispositivos bluetooth que se encuentra en la barra de herramientas de Windows y luego en Agregar un dispositivo Bluetooth.

Figura 1 – Icono de Bluetooth

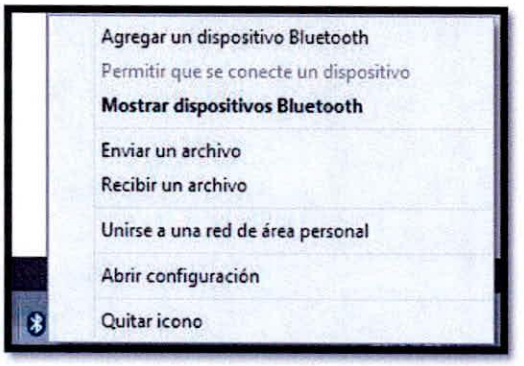


Figura 2 – Agregar dispositivo Bluetooth

Aparecerá la siguiente ventana indicándonos el dispositivo bluetooth encontrado, en este caso MercuryOne es el nombre del módulo bluetooth del robot.

[1] http://es.wikipedia.org/wiki/Bluetooth

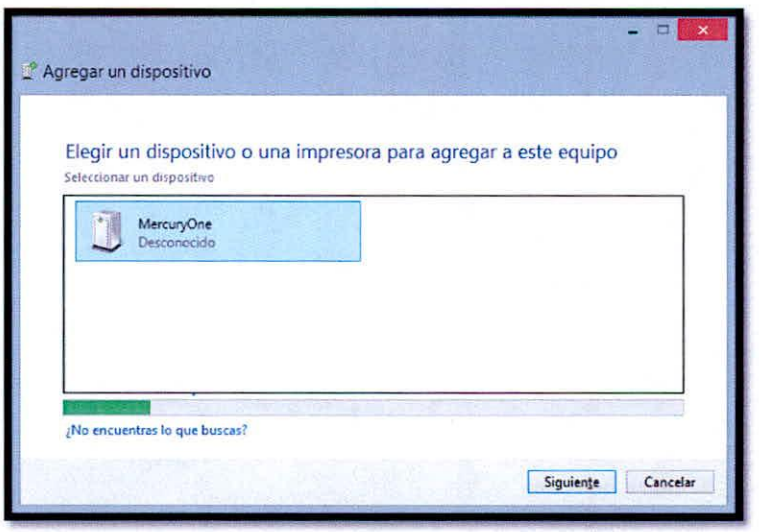


Figura 3 – Elegir un dispositivo

Luego la instalación pedirá el código para acceder al dispositivo, éste es: 0000.

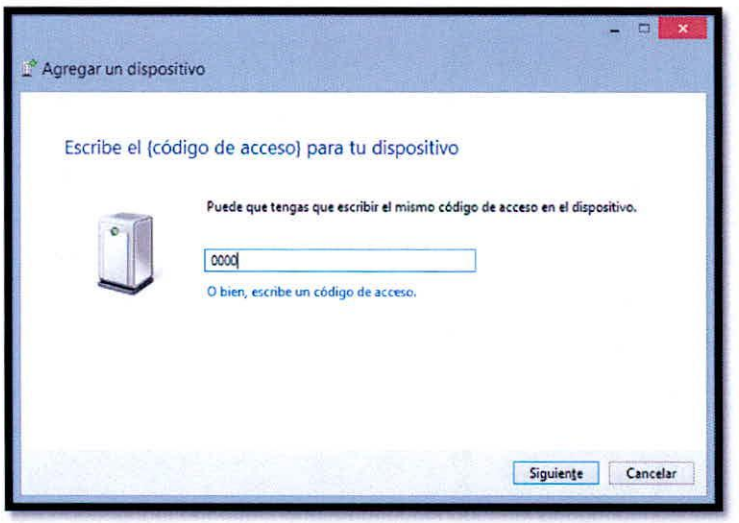


Figura 4 – Escribir código del dispositivo

Luego que reconozca el código de acceso instalará satisfactoriamente el dispositivo para ser utilizado con el Matlab.

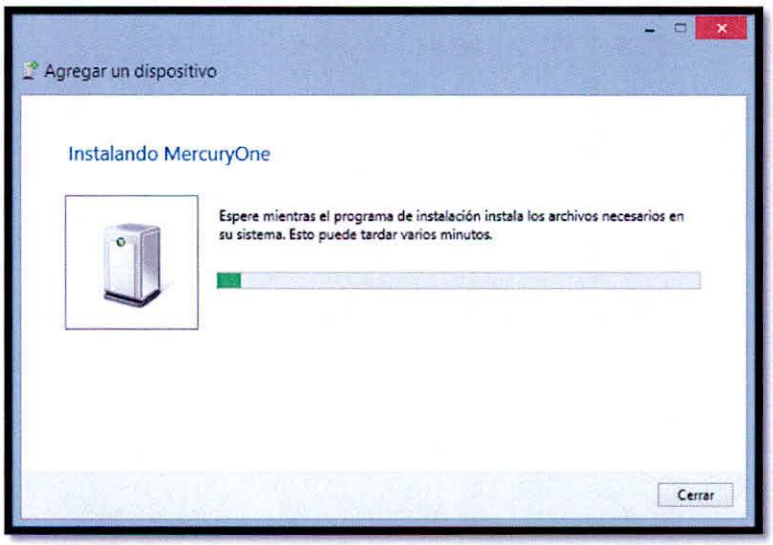


Figura 5 – Instalando MercuryOne Bluetooth

Para realizar la comunicación inalámbrica en el Matlab con el modulo bluetooth se procede a crear un objeto Serial, el cual debe ser configurado con diferentes parámetros tales como, la tasa de baudios (velocidad de transmisión de los datos), numero de bits a enviar y nombre del dispositivo.

Para crear el objeto en Matlab basta con teclear la siguiente línea de código:

```
s = serial('COM9', 'Tag', 'MercuryOne', 'BaudRate', 9600...
,'FlowControl', 'hardware');
```

Y el algoritmo completo para abrir el puerto y establecer la comunicación con el robot seria el siguiente:

```
delete(instrfindall);

s = serial('COM9', 'Tag', 'MercuryOne', 'BaudRate', 9600...
,'FlowControl', 'hardware');

try
    fopen(s);
    [icon,map] = imread('check.png','BackGroundColor',[0.943 0.943
0.943]);
    msgbox('Comunicación establecida','Aviso','custom',icon,map);
    fprintf(s, '%c', '1');
catch
    [icon_b,map_b] = imread('error.png','BackGroundColor',[0.943 0.943
0.943]);
    msgbox('Error al tratar de conectarse con el Robot','Error'...
```

```
        ,'custom',icon_b,map_b);
end

end
```

En el algoritmo anterior, primero se borra cualquier objeto serial anteriormente creado para garantizar una comunicación satisfactoria, dado que si no se borra, el Matlab mostrará errores tales como que el objeto serial ya ha sido creado, o que este ya se encuentra en uso por otro programa. Luego se crea el objeto serial para el dispositivo bluetooth y después dentro de una sentencia try, catch se trata de abrir el puerto para trabajar más adelante con éste. Si existe algún error o la comunicación se ha establecido, será mostrado en una caja de mensajes al usuario, tal y como se observa en las siguientes figuras:

Figura 6 – Comunicación establecida

Figura 7 – Error al conectarse con el robot

El robot posee un módulo bluetooth llamado BlueSmirf Silver de Sparkfun, el cual está montado en una board tipo shield para ser adaptada a la placa base Arduino UNO.

El modulo es fácilmente configurable y tiene un alcance de 18 mts con línea de vista según el datasheet del fabricante, el cual puede ser encontrado en los anexos de este documento.

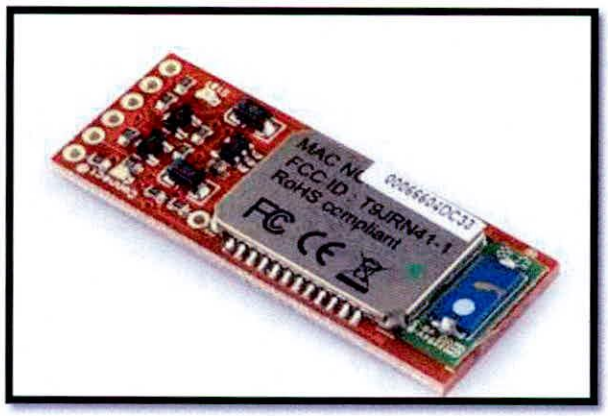


Figura 8 – BlueSmirf Silver

1.6.3. Procesamiento de las imágenes

Con la cámara ya configurada y encendida, y la comunicación establecida con el robot mediante el modulo bluetooth, se procede a realizar el procesamiento de las imágenes.

El algoritmo para el procesado de las imágenes y reconocimiento de objetos es el siguiente:

```
global on_off vid s
cuenta = 0;
if( on_off == 0)
    data = imread('error.png','BackGroundColor',[0.943 0.943 0.943]);
    msgbox('Encienda la camara','Error','custom',data);
else
        start(vid);

    try
        while (on_off==1)

            pause(0.1)
            imagen_o = getsnapshot(vid);
            imagen_reducida = imagen_o(1:2:end,1:2:end,1:1:end);

            imagen_roja = imagen_reducida(:,:,1);
            imagen_verde = imagen_reducida(:,:,2);
            imagen_azul = imagen_reducida(:,:,3);


            imagen_gris = rgb2gray(imagen_reducida);
            imagen_extraida = imsubtract(imagen_roja, imagen_gris);
```

```
        imagen_filtrada = medfilt2(imagen_extraida,[3 3]);

        imagen_binaria = im2bw(imagen_filtrada,0.18);
        binaria = strel('square',6);
        ibinaria = imerode(imagen_binaria,binaria);
        imagen_etiquetada = bwareaopen(ibinaria,350);

        bw = bwlabel(imagen_etiquetada,8);
        cuenta = sum(sum(bw));

        estado = regionprops(bw,'BoundingBox','Centroid');

        try
           if cuenta == 0
             fprintf(s, '%c', '1');
             flushinput(s);
             end

             if cuenta > 0
                   ubi = estado.Centroid;
             if ubi(1) < 150
                   fprintf(s, '%c', '5');
                   flushinput(s);
             elseif ubi(1) >=150 && ubi(1) <210
                   disp('Centro')
                  fprintf(s, '%c', '2');
                   flushinput(s);
             else
                    fprintf(s, '%c', '4');
                    flushinput(s);
             end

                    cuenta = 0;

             end

        catch

      end


imshow(imagen_reducida, 'Parent', handles.axes1);
title('Color','FontName'...
,'Comfortaa Bold','FontSize',16, 'Parent', handles.axes1);

hold on

      for object = 1:length(estado)
```

```
            bb = estado(object).BoundingBox;
            rectangle('Position',bb,'EdgeColor','y','LineWidth',2,
            'Parent', handles.axes1)
        end

hold off

imshow(imagen_gris , 'Parent', handles.axes2);title('Escala de
grises','FontName'...
,'Comfortaa Bold','FontSize',16, 'Parent', handles.axes2);

hold on

        for object = 1:length(estado)
            bb = estado(object).BoundingBox;
            rectangle('Position',bb,'EdgeColor','y','LineWidth',2,
            'Parent', handles.axes2)
        end

hold off

imshow(imagen_filtrada, 'Parent', handles.axes3); title('Filtro de
ruido','FontName'...
,'Comfortaa Bold','FontSize',16, 'Parent', handles.axes3);

hold on

        for object = 1:length(estado)
            bb = estado(object).BoundingBox;
            rectangle('Position',bb,'EdgeColor','y','LineWidth',2,
            'Parent', handles.axes3)
        end

hold off

imshow(imagen_binaria, 'Parent', handles.axes4);
title('Binaria','FontName'...
,'Comfortaa Bold','FontSize',16, 'Parent', handles.axes4);

hold on

        for object = 1:length(estado)
            bb = estado(object).BoundingBox;
            rectangle('Position',bb,'EdgeColor','r','LineWidth',2,'LineSt
            yle','--', 'Parent', handles.axes4)
        end

hold off
end
        catch

        end
end
```

Principalmente se da la instrucción al PC para que adquiera las imágenes que se visualizan en la cámara inalámbrica. Y luego dentro de un ciclo repetitivo mientras que la cámara esté encendida se realiza el procesamiento, con el fin de que este se muestre en tiempo real.

El proceso dentro del ciclo es el siguiente: primero se captura una imagen, luego ésta es reducida con el fin de que el proceso se realice sin interrupciones por lentitud de procesamiento del PC, después de esto se extraen las matrices principales de la imagen (RGB), para los colores rojo, verde y azul. La imagen ya reducida es convertida a escala de grises, y luego se realiza una resta entre las matrices principales y la imagen a escala de grises. El resultado de esto es una imagen extraída la cual se enfoca en el color que arroja la matriz de la cual se sustrajo.

A la imagen extraída se le aplica un filtro, el cual consiste en reducir cualquier ruido que tenga de más la imagen, tal como las intersecciones de colores muy cercanos a las matrices principales.
Luego a la imagen filtrada se le aplica la conversión a blanco y negro, con un umbral determinado de 0.18. La función en Matlab permite umbrales comprendidos entre 0 y 1.

A la imagen binaria se le aplica la técnica de erosión con el fin de reducir lo más posible el ruido de la imagen ya final que en su totalidad es binaria (blanco y negro). Con esta técnica se reduce la región de interés del objeto detectado, pero dentro del algoritmo se tiene prioridad al objeto que más cerca se encuentre del plano izquierdo de la imagen.

Luego la imagen binaria es etiquetada para detectar cuantos objetos se encuentran en ella y se descartan aquellos espacios detectados como objetos que posean menos de 350 pixeles dentro de la imagen.

Para terminar el reconocimiento, se detectan los bordes y se dibuja un rectángulo, y se detecta el centro de cada uno de los espacios blancos de la imagen que cumplan con las condiciones anteriores (es del color de interés, mayor a 350 pixeles, etc.), los cuales serán etiquetados como objetos a recolectar.

Si algún objeto cumple con los requisitos antes mencionados, se procede a enviarle datos vía bluetooth al robot para que este se mueva. El robot tratara siempre de centrar el objeto para poder recogerlo.

1.6.4. Pruebas iniciales

Las siguientes capturas detallan el funcionamiento de los algoritmos utilizados en el proyecto:

Se enciende la cámara del robot

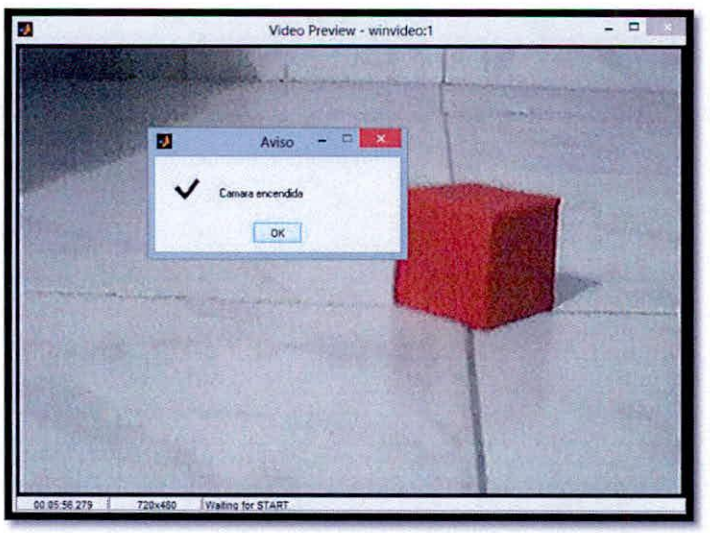


Figura 9 – Cámara encendida

Se detecta el objeto:

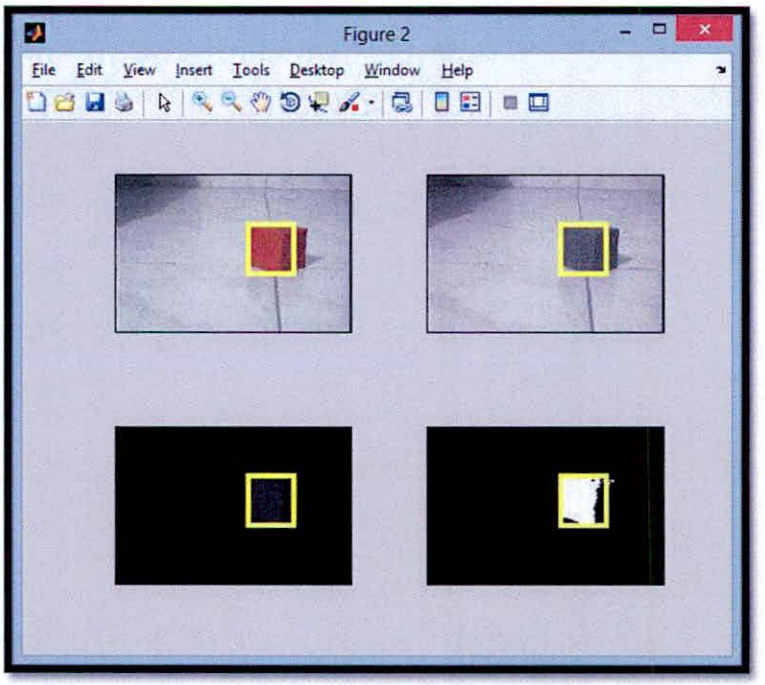


Figura 10 – Objeto detectado

Mirando el objeto de frente:

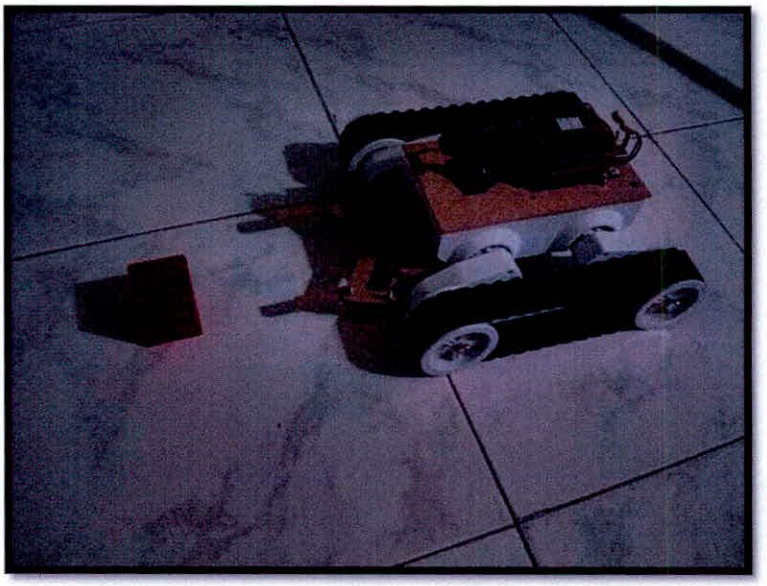


Figura 11 – Robot detectando el objeto de color rojo

1.6.5. Lógica Difusa

El sistema de lógica difusa Mercuryfuzzy (Figura 12) se desarrolló en Matlab por medio del Fuzzy Logic Toolbox empleando la interfaz de usuario Fuzzy Inference System (FIS); la cual es una herramienta gráfica que permite construir, editar y observar un sistema de inferencia difuso.

En el FIS se puede trabajar dos tipos de sistemas difusos Mamdani y Sugeno, se escogió el primero por ser intuitivo. Además, las funciones de membresía de la salida son conjuntos difusos mientras que en el segundo las mismas son solo funciones constantes y lineales.

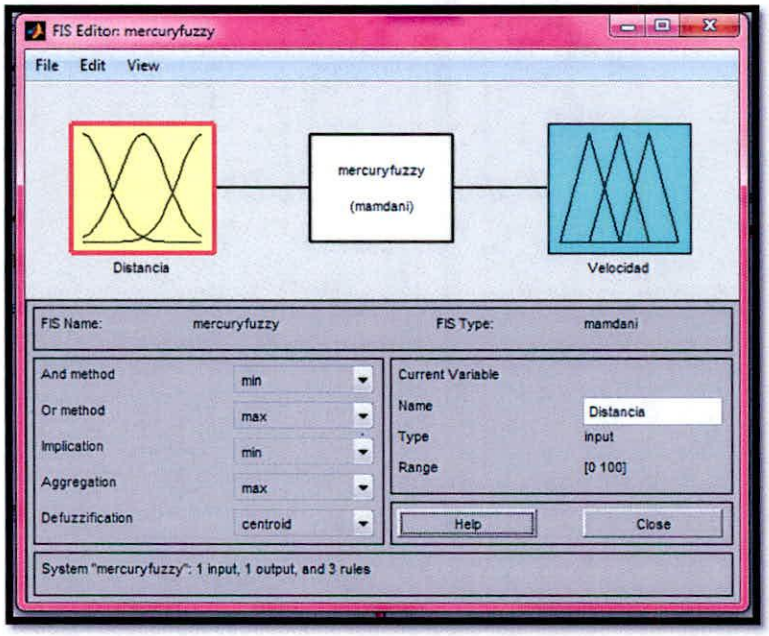


Figura 12 - Fuzzy Inference System

El Mercuryfuzzy cuenta con:

- Variables lingüísticas: Distancia y Velocidad. El rango de la primera es de 0 a 100, escogido así por la referencia del sensor de ultrasonido y el de la segunda es de 0 a 281 (debería ser de 0 a 255 que son los valores que acepta el Shield de Ardumoto, pero así no se llegaría al 100% de la velocidad).

- Funciones de membresía: Una función que especifica el grado en el que una entrada dada pertenece a un conjunto o se relaciona con un concepto.

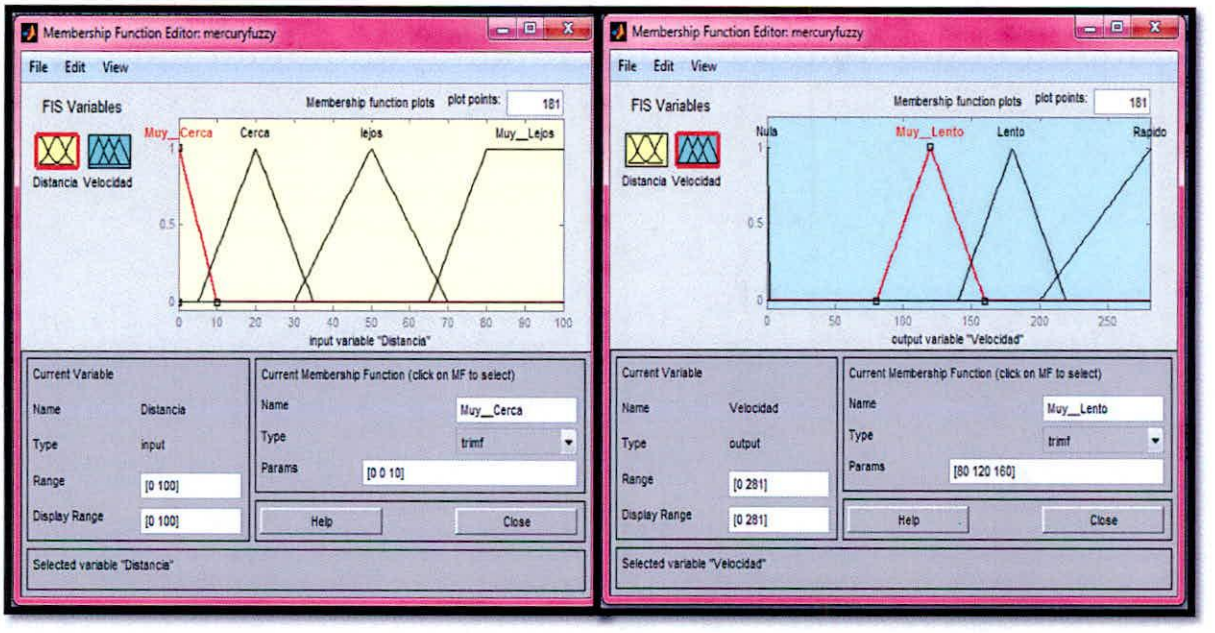


Figura 13 – Funciones de Membresía

- Distancia:
  - Muy Cerca: Es una función de tipo triangular. Con los parámetros de [0 0 10]. Este tipo de función en el parámetro del medio alcanza el 100% de pertenencia.
  - Cerca: Es una función de tipo triangular. Con los parámetros de [5 20 35].
  - Lejos: Es una función de tipo triangular. Con los parámetros de [30 50 70].
  - Muy Lejos: Es una función de tipo trapezoidal. Con los parámetros de [65 80 100 110]. Este tipo de función entre el intervalo conformado por los parámetros 2 y 3 alcanza y mantiene el 100% de pertenencia.

- Velocidad: Como se puede observar entre el intervalo abierto [0 80] no existe función de membresía; ya que cuando la variable adquiere uno de estos valores los motores se esfuerzan demasiado y no logran moverse, lo que los puede dañar.

- Nula: Es una función de tipo triangular. Con los parámetros de [0 0 0], esto se hace con el fin de poder detener los motores; según el Ardumoto tiene que ser cero la variable.
- Muy Lento: Es una función de tipo triangular. Con los parámetros de [80 120 160].
- Lento: Es una función de tipo triangular. Con los parámetros de [140 180 220].
- Rápido: Es una función de tipo triangular. Con los parámetros de [200 282 292].

- Las reglas: Después de tener todas las entradas, salidas y sus respectivas funciones de membresías, se puede crear las reglas de inferencia del sistema difuso en el Rule Editor(Figura 14)

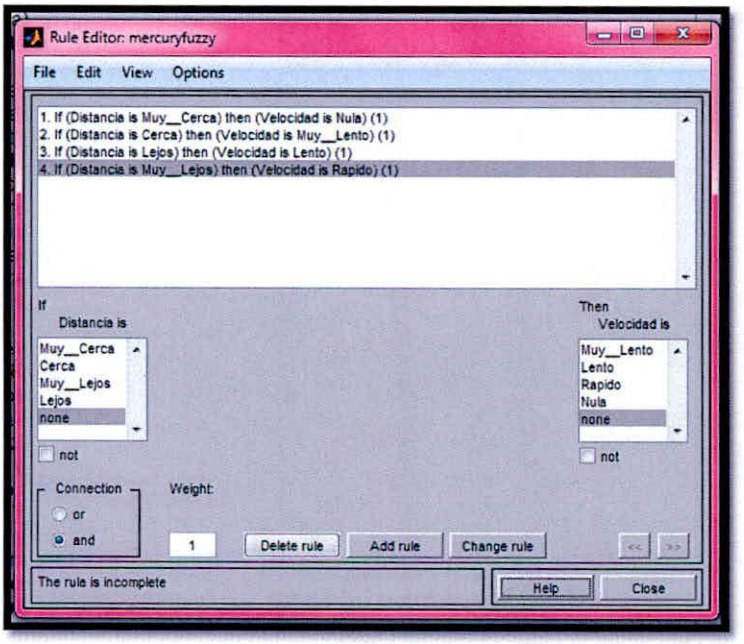


Figura 14 - Editor de Reglas.

Las reglas tienen la forma Si X es A (antecedente) Entonces Y es B (consecuente), donde X e Y son las variables lingüísticas y A y B son las respectivas funciones de pertenencia.

Se tienen 4 reglas:

- 1. If (Distancia is Muy_Cerca) then (Velocidad is Nula) (1)
- 2. If (Distancia is Cerca) then (Velocidad is Muy_Lento) (1)
- 3. If (Distancia is Lejos) then (Velocidad is Lento) (1)
- 4. If (Distancia is Muy_Lejos) then (Velocidad is Rapido) (1)

Se puede visualizar la aplicación de dichas reglas por medio del Rule Viewer. En la figura 15 se muestra el valor que toma la variable lingüística Velocidad según la correspondiente función de pertenencia dependiendo del valor que se le dé a Distancia.

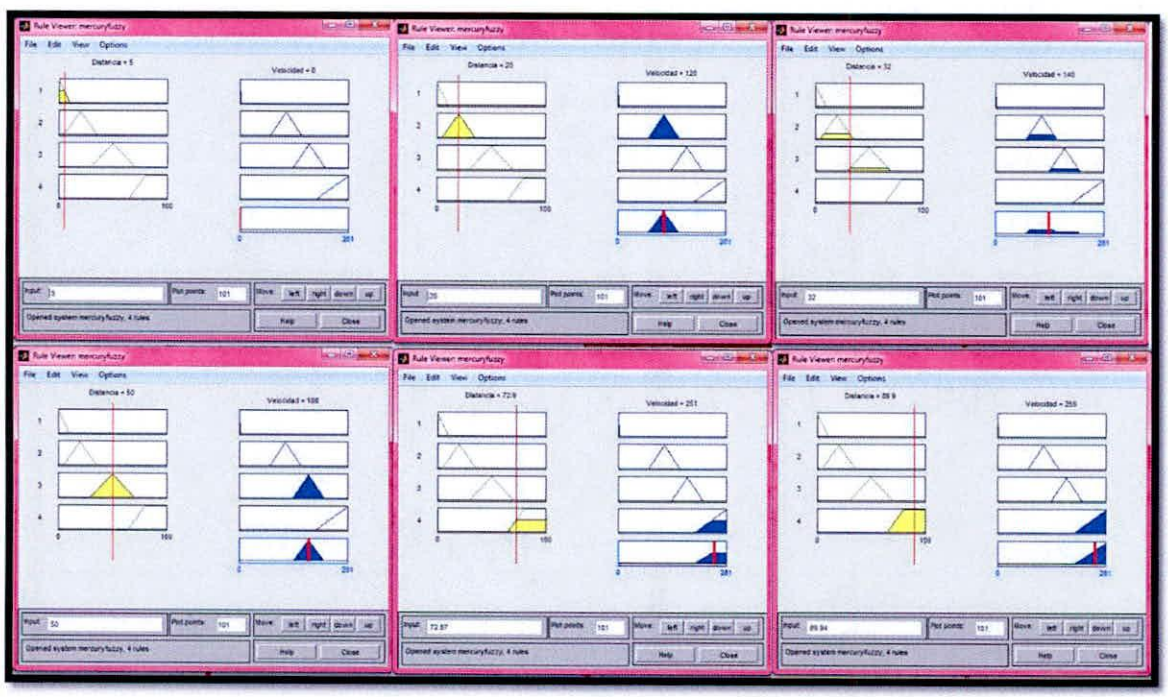


Figura 15 - Prueba de Mercuryfuzzy

1.6.6. Interfaz grafica

La interfaz gráfica del robot se desarrolló con ayuda del GUI de Matlab, en la cual se hace uso de algunas librerías de java para llamar icono e imágenes y se llaman directamente las funciones realizadas en Matlab. A continuación se detalla cada uno de sus componentes:

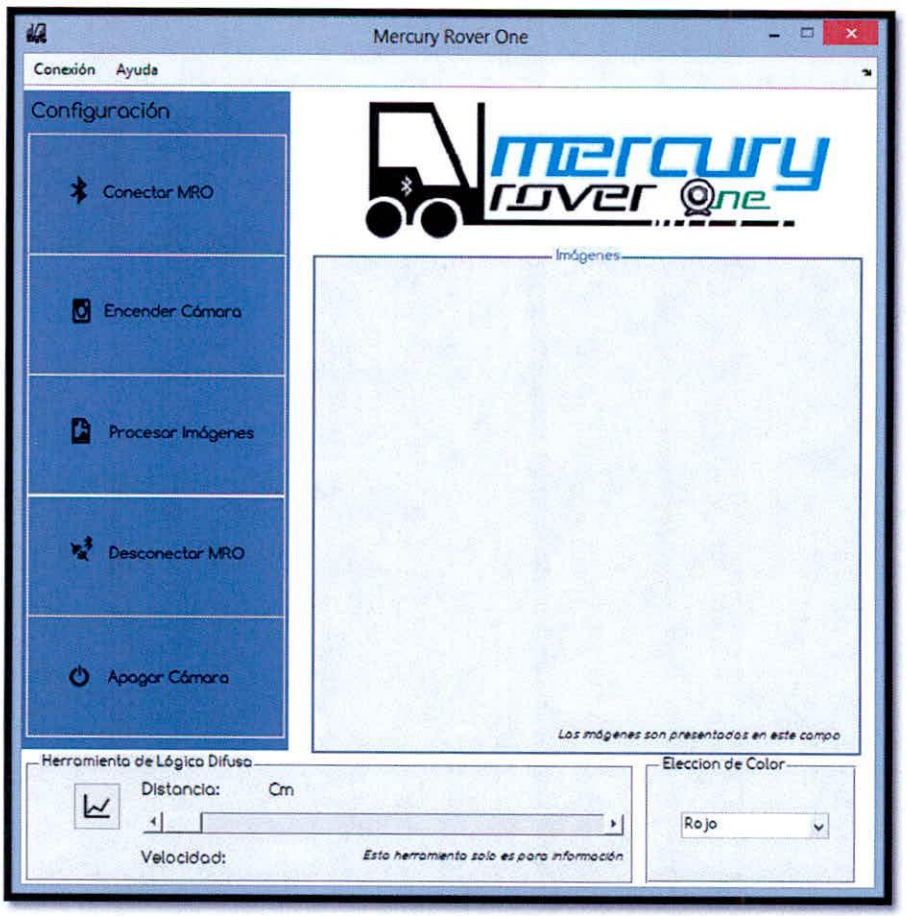


Figura 16 – Interfaz del robot MercuryOne

Conectar MRO: este botón se encarga de entablar la comunicación entre el PC y el robot.

Encender Cámara: enciende la cámara del robot.

Procesar Imágenes: con este botón se realiza el procesamiento de las imágenes y el software le envía los datos al robot para que este se mueva respecto al reconocimiento de los objetos a recolectar.

Desconectar MRO: realiza la desconexión del PC al robot.

Apagar Cámara: apaga la cámara del robot.

Imágenes: en este apartado son visualizadas las imágenes obtenidas por la cámara. (Imagen Original, Escalas de Grises, Imagen Filtrada e Imagen Binaria).

Herramienta de Lógica Difusa: esta herramienta provee información acerca de cómo se están manejando las entradas y salidas del sistema difuso utilizado por el robot.

Elección de color: este apartado permite al usuario escoger entre los colores rojo, azul y verde (RGB) para realizar el reconocimiento de objetos del color especificado.

Pruebas finales de la interfaz grafica y el prototipo en uso:

La interfaz grafica sirve como una guia visual para ver en tiempo real lo que ocurre con el robot y que es lo que éste observa.

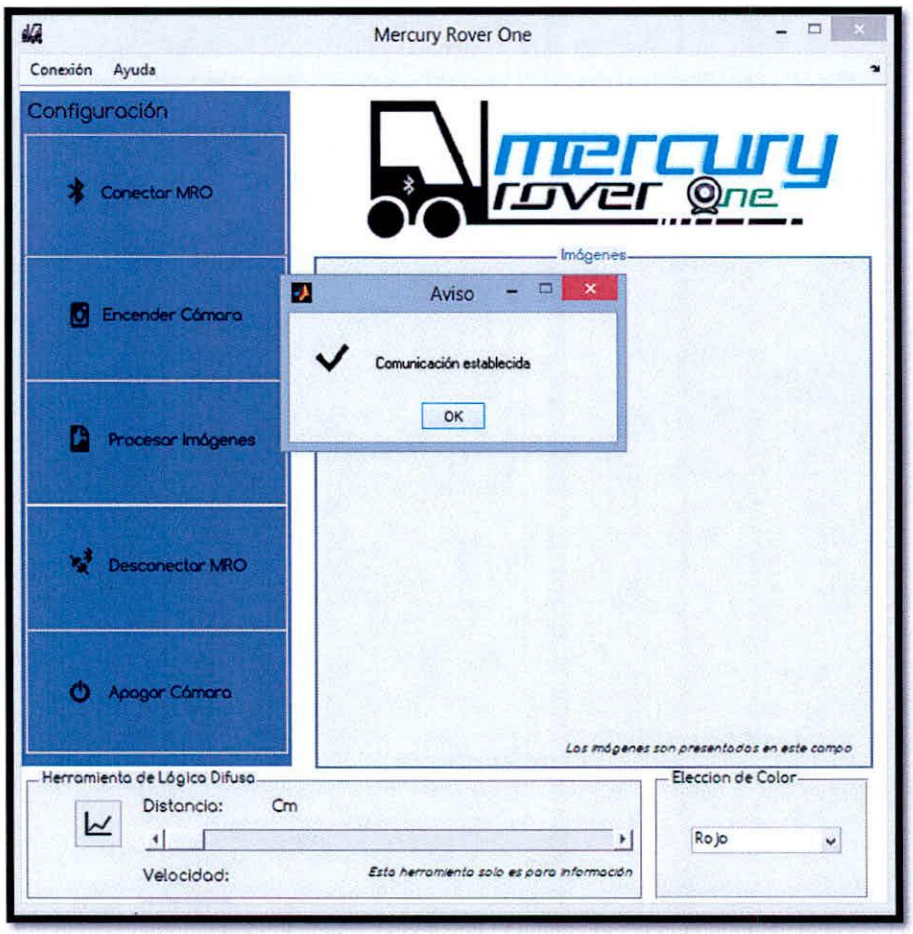


Figura 17 – Interfaz Gráfica, Comunicación establecida

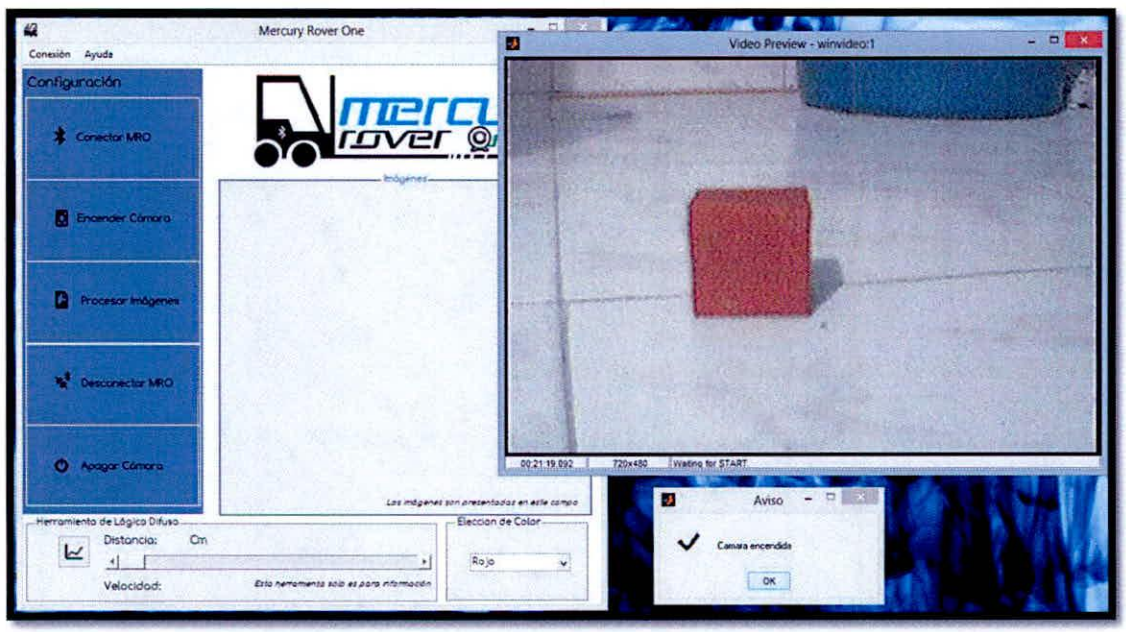


Figura 18 – Interfaz Gráfica, Cámara encendida

Figura 19 – Interfaz Gráfica, Imágenes procesadas y objeto detectado

Figura 20 – Desconectando el robot

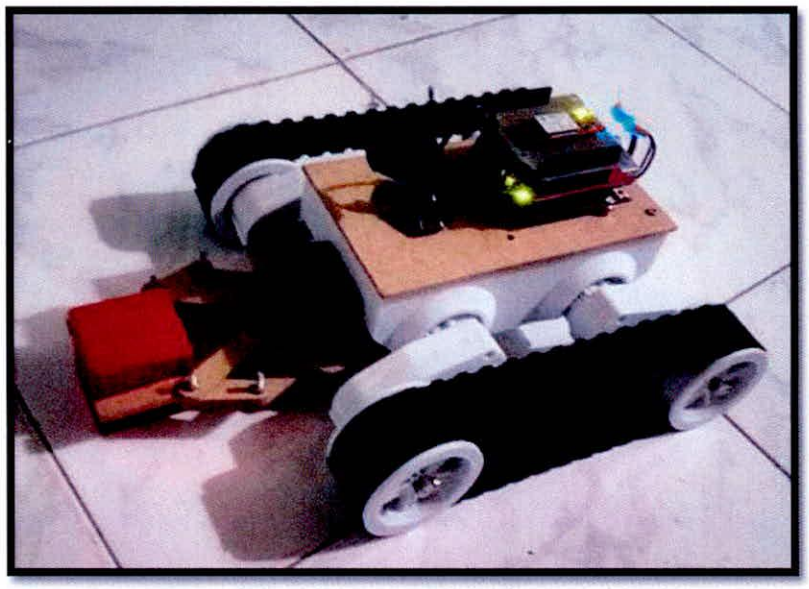


Figura 21 – Objeto recolectado por el robot

1.6.7. Plataforma Rover 5

La plataforma Rover 5 es una moderna solución para robots con tracción tipo oruga, a diferencia de los sistemas tradicionales de oruga la separación respecto al piso puede ser variada ajustando la rotación de las cajas reductoras en incrementos de 5 grados.

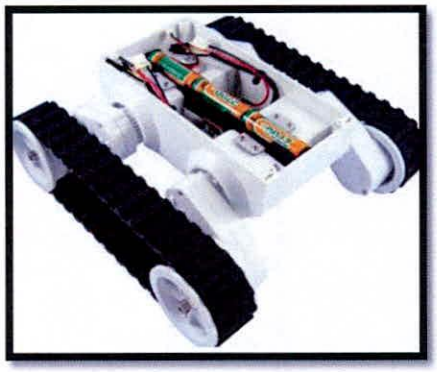


Figura 22 – Plataforma Rover 5

Especificaciones:

- Voltaje del motor: 7.2v
- Corriente del motor en reposo: 2.5A
- Torque de salida: 10Kg/cm
- Radio de la caja reductora: 86.8:1
- Velocidad: 1Km/hr
- Dos motores sin encoder

1.6.8. Firmware de Arduino

Arduino es un conjunto de tres elementos una plataforma de hardware o placas, el entorno y lenguaje de programación, creada con la finalidad de facilitar el desarrollo de prototipos electrónicos u objetos interactivos.

Actualmente es muy usada por sus grandes ventajas, entre las cuales están: la disponibilidad de sus componentes bajo licencias abiertas, la placa viene lista para usarse, solo se necesita conectar los sensores y actuadores, en entre otros, ocupan poco espacio,

el consumo de potencia es bajo y gracias al uso de las librerías, la programación es sencilla y fácil de aprender.

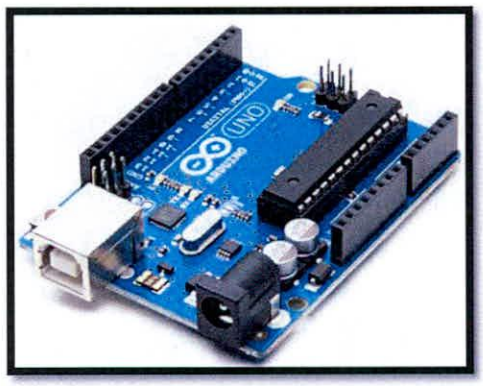


Figura 23 – Placa Arduino Uno

En el entorno de programación por lo general se trabaja con base a dos estructuras, el setup() es una función que solo se ejecuta una única vez al inicio del programa para inicializar las variables, establecer el estado de los pines e inicializar las librerías. Y el loop() es una función que se ejecuta consecutivamente, usada para controlar de forma activa al Arduino.

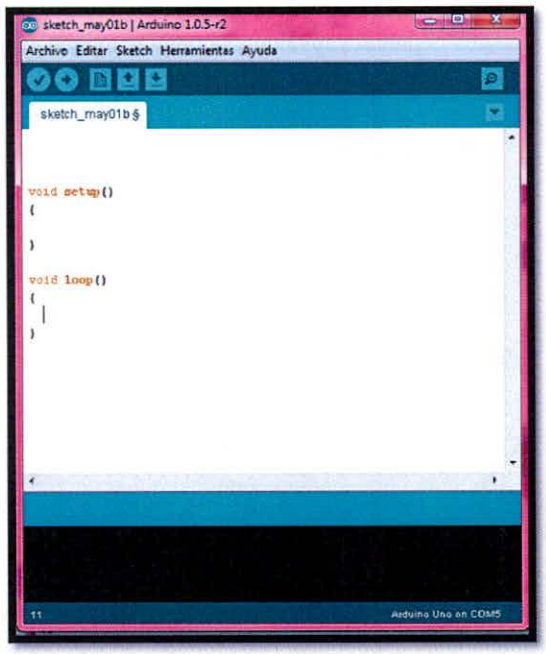


Figura 24 – Entorno Arduino (IDE)

Por medio de una placa de Arduino Uno en el robot Mercury One se controla las siguientes actividades:

- Comunicación serial : Para realizar la comunicación primero se incluye la librería Serial y se inicializa el puerto cuando se requiere leer un dato recibido o enviar uno por el puerto serial se usan las funciones que aparecen dentro del loop en el siguiente código.

```
#include <SoftwareSerial.h>
void setup()
{
Serial.begin(9600);
}
void loop() // run over and over
{
  while(Serial.available() > 0)
  {
     char recieved = Serial.read();
  }
  Serial.flush();
  Serial.print(R);
}
```

- El movimiento de la plataforma: Se comienza con la creación de los pines para controlar la Shield del Ardumoto y luego se inicializan como salidas en el setup, en el loop las líneas de código determinan la dirección de los motores y la velocidad dependiendo de la rutina escogida por el Matlab según los resultados del procesamiento de la imagen.

  - Exploración: En esta el robot no ha detectado ningún objeto, por lo que girará sobre su propio eje para encontrarlo, con una velocidad alta.

```
void loop(){
if(recibido == '1'){
        exploracion();}}
void exploracion(){
  digitalWrite (dir_a, LOW);
```

```
digitalWrite (dir_b, HIGH);
analogWrite (pwm_a, 200);
analogWrite (pwm_b, 200); }
```

- Seguimiento: Luego de reconocer el objeto el robot debe centrarlo y acercarce según los comandos de recibidos del Matlab los cuales direccionan al programa para que vaya a las siguientes subrutinas, ejecute su contenido, regrese y borre el registro.

```
void derecha() {
  digitalWrite (dir_a, LOW);
  digitalWrite (dir_b, HIGH);
  analogWrite (pwm_a, 250);
  analogWrite (pwm_b, 250); }

void izquierda() {
  digitalWrite (dir_a, HIGH);
  digitalWrite (dir_b, LOW);
  analogWrite (pwm_a, 250);
  analogWrite (pwm_b, 250); }

void adelante() {
  digitalWrite (dir_a, HIGH);
  digitalWrite (dir_b, HIGH);
  analogWrite (pwm_a, 250);
  analogWrite (pwm_b, 250); }

void atras() {
  digitalWrite (dir_a, LOW);
  digitalWrite (dir_b, LOW);
  analogWrite (pwm_a, 250);
  analogWrite (pwm_b, 250); }

void detenido() {
  digitalWrite (dir_a, LOW);
  digitalWrite (dir_b, LOW);
  analogWrite (pwm_a, 0);
  analogWrite (pwm_b, 0); }
```

- Calcular la distancia: La distancia se mide por medio del sensor ultrasonido. Para determinarla primero creamos las variables y el pin por donde se comunicara con el

sensor y se creó una subrutina para convertir el ancho de pulso que se recibe en centímetros.

```
//ultrasonido
 unsigned long pulso; //variable para medir el pulso
 float cm;          //calcular la distancia
 int pin = 8;       //pin

void dist()
{
  pinMode(pin, OUTPUT);     // ponemos el pin como salida
  digitalWrite(pin, HIGH);  // lo activamos
  delayMicroseconds(10);    // esperamos 10 microsegundos
  digitalWrite(pin, LOW);   // lo desactivamos
  pinMode(pin, INPUT);      // cambiamos el pin como entrada
  pulso = pulseIn(pin, HIGH);  // medimos el pulso de salida del sensor
  cm = ((float(pulso/1000.0))*34.32)/2;
}
```

- La recolección de los objetos: Cuando el robot tiene centrado al objeto y se encuentra lo suficientemente cerca para recogerlo se necesita controlar el servomotor que mueve a la pinza, para lo cual se incluye la librería Servo, se crea un objeto y una variable, a esta última le asignamos un ángulo, inicializamos un pin, y se cambia el ángulo del servomotor según el dato recibido.

```
#include <Servo.h>

Servo miServo;
int angulo = 90;

void setup() {
miServo.attach(9); }

void loop() {
      if (recieved == 'a'){
        angulo = 70;            //abrir la pinza
        Serial.flush(); }
      else if (recieved == 'z') {
        angulo = 130;           //cerrar la pinza
        Serial.flush(); }
```

1.6.9. Board control de los motores

Esta board es un Shield para Arduino el cual puede ser adaptado fácilmente a la placa base Arduino UNO y se encarga de controlar motores DC. Está basado en el circuito integrado puente-H L298, el cual puede manejar hasta 2 amperios por canal. Lo suficiente para mover con soltura la base del robot. La board puede ser alimentada hasta con 6 baterías AA y puede proporcionarle energía a la placa base.

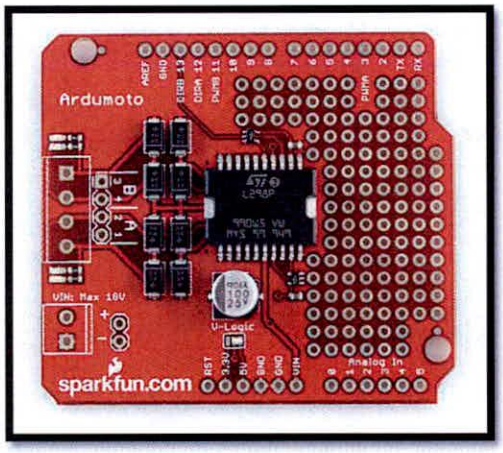


Figura 25 – Shield Ardumoto para Arduino

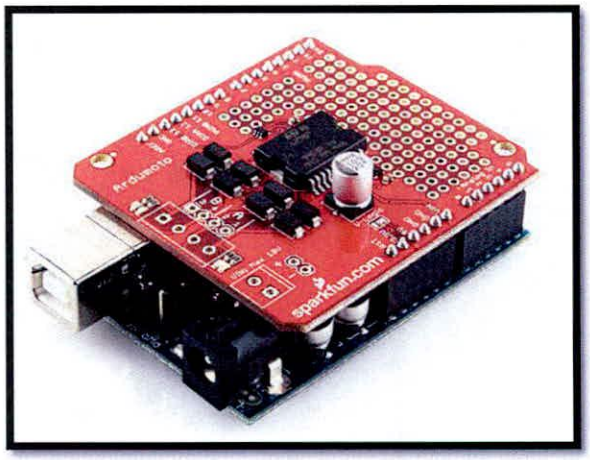


Figura 26 – Shield conectado al Arduino

El esquema de la board y el datasheet del puente-H L298, pueden encontrarse en los anexos de este documento.

1.6.10. Sensor de distancia

El sensor de distancia cumple un papel muy importante a la hora de recolectar los objetos por parte del robot. Este le indica a la placa base a que distancia se encuentra el objeto del robot y la placa se comunica con el PC para darle las instrucciones obtenidas por medio de la lógica difusa del robot y mediante estas el robot se moverá a diferentes velocidades y tratará de utilizar la pinza para agarrar el objeto.

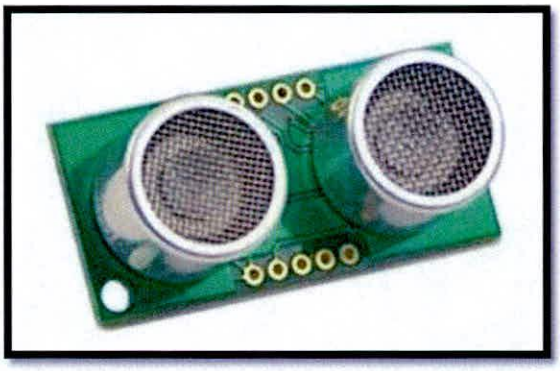


Figura 27 – Sensor de distancia SRF05

Este sensor es un módulo ultrasónico o de ultrasonido que integra un receptor y emisor en una sola board, debido a su baja zona muerta (1,7cm) es ampliamente usado en robots de sumo, así como en medición de distancia, control de nivel, detector de objetos, etc.

Características

- Rango - 1,7 cm a 4m.
- Voltaje - 5v, 4mA Typ.
- Frecuencia - 40KHz.
- Tamaño 43mm x 20mm x 17mm.

1.6.11. Pinza para recolectar los objetos

Para recolectar los objetos el robot posee una pinza con la cual los puede agarrar dependiendo a la distancia que detecte el sensor de ultrasonido. Esta pinza ha sido fabricada manualmente a partir de un modelo 3D encontrado en la página MiniServo Gripper.

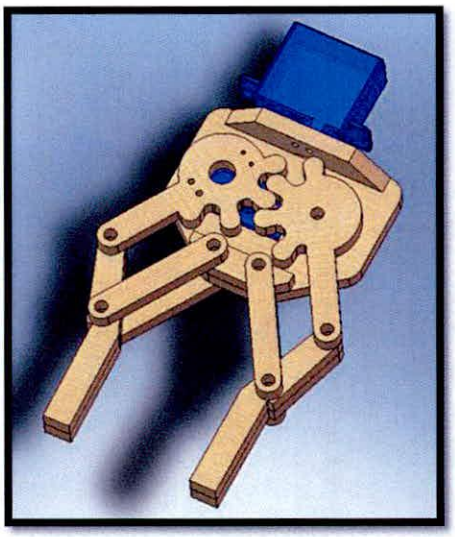


Figura 28 – Modelo 3D de la pinza

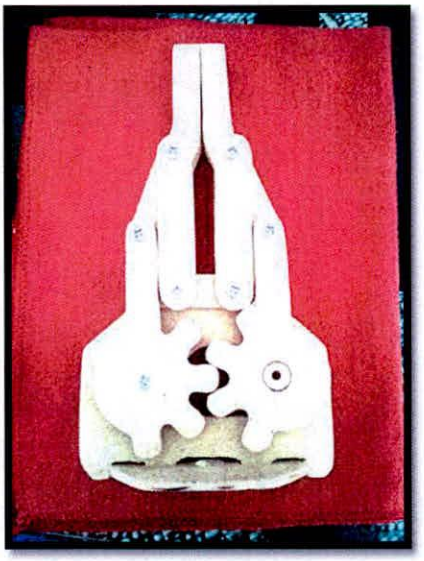


Figura 29 – Pinza fabricada

1.6.12. Alimentación del prototipo

El robot cuenta con una alimentación independiente. La base de los motores posee un rack de baterías AA que alberga 6 de estas, con la cual se energizan los motores DC, la placa base de Arduino UNO, el shield Ardumoto y el modulo bluetooth.

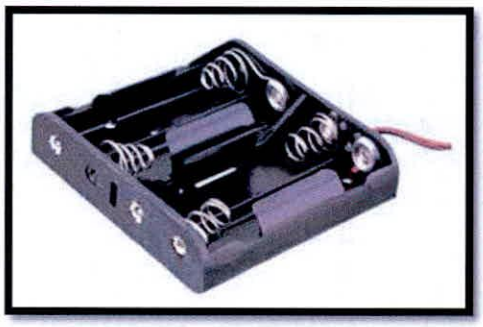


Figura 30 – Arreglo de baterías AA

Figura 31 – Batería AA

La cámara inalámbrica es alimentada con una batería de 9V, esta es independiente dado a que al estar alimentados con el rack de baterías AA los motores al arrancar distorsionan la imagen de la cámara.

Figura 32 – Batería 9V

1.7. Diseño Metodológico

1.7.1. Tipo de investigación

Existen varios tipos de investigación los cuales a su vez obedecen a diversidad de criterios propuestos por distintos autores. De todas maneras, fuera cual se la clasificación adoptada, esta no supone fronteras estrictas entre las clases de investigación pues en muchos casos es muy difícil definir límites entre una y otra [2].

A continuación se muestra una clasificación de los tipos de investigación:

Histórica: Busca reconstruir el pasado de manera objetiva, con base en evidencias documentales confiables. Se aplica no sólo a la historia sino también a las ciencias de la naturaleza, al derecho, la medicina o cualquier otra disciplina científica y o social.

Descriptiva: Comprende la descripción, es una forma de estudio para saber quién, dónde, cuándo, cómo y porqué del sujeto del estudio o área de interés. Explica perfectamente las características de cualquier conjunto de sujetos.

Experimental: Es aquella que permite con más seguridad establecer relaciones de causa y efecto. Se presenta mediante la manipulación de una variable experimental no comprobada, en condiciones rigurosamente comprobadas, con el fin de describir de qué modo o por qué causa se produce una situación o acontecimiento particular.

Exploratoria: Su objetivo es examinar un tema o problema de investigación poco estudiado o que no ha sido abordado antes. Nos sirve para aumentar el grado de familiaridad con fenómenos relativamente desconocidos.

Aplicada: Aquella que incluye cualquier esfuerzo sistemático y socializado por resolver problemas o intervenir situaciones, aunque no sea programático, es decir, aunque no

[2] (Castillo, 2004)

pertenezca a una trayectoria de investigaciones descriptivas y teóricas. En ese sentido se concibe como *investigación aplicada* tanto la innovación técnica, como la propiamente científica.

También es aquella que considera los estudios que explotan teorías científicas previamente validadas para la solución de problemas prácticos y el control de situaciones de la vida cotidiana. [3]

Se considera ésta investigación del tipo aplicada, dado a que el proyecto trata de dotar a un sistema robótico con una capacidad que para muchos seres humanos es cotidiana, es el adquirir e interpretar la información visual del medio para poder tomar decisiones, y dado a que la investigación aplicativa hace referencia al uso de estudios científicos orientados a controlar situaciones prácticas, los cuales en éste caso son teorías del procesamiento de señales digitales y la robótica, de manera especial los temas del tratamiento de imágenes, la lógica difusa y la construcción física de robots.

1.7.2. Métodos de investigación

Para López José, Hernández Roberto y Fernández Carlos existen varios métodos de investigación, y se mencionaran a continuación:

El método histórico: Está vinculado al conocimiento de las distintas etapas de los objetos en su sucesión cronológica, para conocer la evolución y desarrollo del objeto o fenómeno de investigación se hace necesario revelar su historia, las etapas principales de su desenvolvimiento y las conexiones históricas fundamentales. [4]

Método sintético: Es un proceso mediante el cual se relacionan hechos aparentemente aislados y se formula una teoría que unifica los diversos elementos. Consiste en la reunión racional de varios elementos dispersos en una nueva totalidad, este se presenta más en el

[3] (Padro, 2012)
[4] (Lopez, 1984)

planteamiento de la hipótesis. El investigador sintetiza las superaciones en la imaginación para establecer una explicación tentativa que someterá a prueba. [5]

Método analítico: Se distinguen los elementos de un fenómeno y se procede a revisar ordenadamente cada uno de ellos por separado. La física, la química y la biología utilizan este método; a partir de la experimentación y el análisis de gran número de casos se establecen leyes universales. Consiste en la extracción de las partes de un todo, con el objeto de estudiarlas y examinarlas por separado, para ver, por ejemplo las relaciones entre las mismas. [6]

El método científico: El método científico o experimental es una manera de recopilar información y comprobar ideas. Es la forma en que un científico trata de hallar respuestas a sus interrogantes sobre la naturaleza. A pesar de que el procedimiento puede variar, el método científico consta de los siguientes pasos generales: hacer observaciones; formular hipótesis; someter a prueba las hipótesis y llegar a conclusiones. El método científico es lo que distingue a la ciencia de los otros campos de estudio. [7]

El método más apropiado para este trabajo es el científico, se desarrollará el diseño del prototipo en actividades semanales y se realizarán pruebas experimentales con los códigos y los mecanismos, teniendo en cuenta las variables que se puedan presentar en las distintas situaciones a las que el prototipo se enfrente, y a partir de esto se sacarán las conclusiones de las preguntas que se plantearon al inicio de ésta investigación.

1.7.3. Recolección de la información

La recolección de la información debe realizarse utilizando un proceso planeado paso a paso, para que de forma coherente se puedan obtener resultados que contribuyan favorablemente al logro de los objetivos propuestos.

[5] (Lopez, 1984)
[6](Lopez, 1984)
[7](Métodos de Investigación)

Si en el proceso investigativo, la obtención y recolección de la información no se realiza sistemáticamente, siguiendo un proceso ordenado y coherente, que a su vez permita evaluar la confiabilidad y validez tanto del proceso mismo como de la información recolectada, ésta no será relevante y por lo tanto no podrá reflejar la situación que se pretende describir.

Se hace necesario que el investigador y los responsables de estas acciones tengan un dominio conceptual y teórico tanto del tema objeto de investigación, como de la población a estudiar, para minimizar la posibilidad de que se presenten sesgos en esta etapa. Una vez identificadas las necesidades de información se pueden realizar tres actividades estrechamente relacionadas entre sí:

La primera se refiere a la selección de los instrumentos de medición y/o técnicas de recolección de información.
La segunda se relaciona con la aplicación de estos instrumentos.
La tercera concierne a la preparación o codificación de la información obtenida en busca de facilitar su análisis.

Existen dos tipos de información:

Información primaria. Es aquella que el investigador recoge directamente a través de un contacto inmediato con su objeto de análisis.

Información secundaria. Es aquella que el investigador recoge a partir de investigaciones ya hechas por otros investigadores con propósitos diferentes. La información secundaria existe antes de que el investigador plantee su hipótesis, y por lo general, nunca se entra en contacto directo con el objeto de estudio.

Por lo tanto este trabajo se llevará a cabo haciendo una recolección de la información de tipo secundaria; ya que se irá probando teorías y conclusiones de otras personas que han trabajado en los campos del procesamiento de señales digitales y la robótica.

1.8. Cronograma

N°	ACTIVIDADES	SEMANAS																			
		DICIEMBRE				ENERO				FEBRERO				MARZO				ABRIL			
1	Adquisición de los componentes	X	X	X																	
2	Realización de script en Matlab (procesamiento de imágenes y Lógica Difusa)	X	X	X	X	X	X	X	X	X	X		X	X	X						
3	Diseño de la plataforma y del mecanismo recolector			X	X	X															
4	Construcción de plataforma y mecanismo recolector					X	X	X													
5	Pruebas con imágenes en tiempo real adquiridas con cámara inalámbrica									X	X	X									
6	Desarrollo de sistema de control con Arduino (Lógica difusa)						X	X	X		X	X									
7	Establecer una comunicación inalámbrica entre el computador y el Arduino										X	X									
8	Pruebas de reconocimiento												X	X							
9	Pruebas de rastreo													X	X	X					
10	Pruebas de recolección															X	X				
11	Diseñar una interfaz visual amigable para que el usuario pueda interactuar con el sistema fácilmente.																X	X			
12	Prueba final del sistema																	X	X	X	
13	Elaborar un manual técnico con las especificaciones del prototipo.																			X	X

Tabla 1 - Cronograma

1.9. Resultados

El Mercury One es capaz de reconocer, realizar seguimiento y recolectar objetos (cubos de cartón) de colores pre-seleccionados, en la interfaz gráfica por el usuario. Así mismo, sirve como herramienta para el desarrollo de nuevos algoritmos de procesamientos de imágenes y visión artificial por parte de los estudiantes de pregrado de la carrera de Ingeniería Electrónica.

1.9.1. Productos y Logros

Se espera que con este proyecto, se incentive a los estudiantes a desarrollar investigaciones que combinen distintas teorías de la carrera y que estas posean un contenido donde se plasme el conocimiento que se ha adquirido en pregrado. La combinación del desarrollo de Software, Firmware y Hardware hace de un estudiante de Ingeniería Electrónica alguien capaz de afrontar distintas clases de roles en su vida laboral.

1.9.2. Efectos (Impacto)

Este proyecto de investigación tiene como finalidad la aplicación de teorías que profundizan y complementan los temas de las cátedras de procesamiento digital de señales y robótica, enriqueciendo a los autores del mismo, pero sobre todo fortaleciendo a los estudiantes de Ingeniería Electrónica de la Universidad del Magdalena, los cuales desde las cátedras ya mencionadas, puedan hacer uso del prototipo y del sistema completo para desarrollar laboratorios en los temas de tratamiento digital de imágenes, lógica difusa y robótica.

Por otro lado, en las actividades para fomentar el estudio de la ingeniería electrónica se espera que el prototipo sea una muestra de creativa e ingenio que motivé a los estudiantes de básica secundaria ser ingenieros electrónicos.

1.9.3. Recomendaciones

Existen detalles del sistema en general que pueden ser mejorados por los desarrolladores para lograr brindar una experiencia de usuario de mayor satisfacción, tales como: agregar mayor cantidad de colores a escoger para recolectar los distintos objetos, mejorar los acabados del prototipo o cambiar las baterías comunes por baterías recargables.

2. Presupuesto

2.1. Recursos Humanos

Para la realización del proyecto se contará con la dirección del M.SC. ING. José Ramón Iglesias Gamarra, quien orientó en la elaboración de éste documento, supervisará y asesorará en las demás etapas del proyecto de grado, para lo cual brindará 2 horas semanales durante los 5 meses. El valor estimado es:

Valor director:	$ 2'700.000
Total de Recursos Humanos:	$ 2'700.000

2.2. Recursos Institucionales

Se dispondrá en los momentos que estén disponibles y sea necesario el uso de las instalaciones, equipos y software con sus respectivas licencias, de los laboratorios de Ingeniería Electrónica pertenecientes a la Universidad del Magdalena.

Los valores estimados para ello son:

Valor Computadores:	$ 3'600,000
Valor Matlab:	$ 1'800,000
Valor Otros:	$ 1'200,000
Total Recursos Institucionales:	$ 5'600,000

2.3. Recursos Adicionales

COMPONENTE	CANT	VALOR UNITARIO	VALOR TOTAL
RF 2.4G cámara inalámbrica de PC	1	155.500	155.500
Envío de la cámara	1	30.000	30.000
Base tipo oruga Rover 5	1	150.000	150.000
Shield Ardumoto:	1	23.000	23.000
Arduino UNO	1	45.000	45.000
Baterías AA	6	3.500	21.000
Modulo Bluetooth BlueSmirf Gold	1	90.000	90.000
Receptor bluetooth para PC	1	10.000	10.000
PC portátil Acer Aspire 4750, Procesador i5, 8Gb	1	1'400.000	1'400.000
PC portátil SONY Vaio, Procesador i3, 3Gb	1	1'200.000	1'200.000
Conectores CN904	1	500	500
Headers para shields de Arduino	1	3.500	3.500
Cable Vehicular 22AWG	9	1.000	9.000
Sensor ultrasónico SRF05	1	58.500	58.500
Batería 9V	1	10.000	10.000
Mototool Bauker	1	99.000	99.000
Servomotor Mediano	1	51.000	51.000
Lamina de madera MDF	1	8.000	8.000
Diseño del logo de la interfaz	1	50.000	50.000
Alcohol isopropilico	1	9.000	9.000
Tornillos	10	300	3.000
TOTAL		3.397.800	**3.426.000**

Tabla 2 – Costos de los Componentes

Valor Componentes:	$ 3.426.000
Valor Internet:	$ 600.000
Valor Transporte:	$ 1´000.000
Valor Otros:	$ 800.000
Total Recursos Adicionales:	$ 5'826.000

2.4. Costo Total del Proyecto

RECURSOS	COSTO
HUMANOS	$ 2.700.000
INSTITUCIONALES	$ 5.600.000
ADICIONAL	$ 5.826.000
TOTAL	**$ 14.126.000**

Tabla 3 – Costo total del proyecto

3. Referencia

3.1. Bibliografía y Webgrafía

CASTILLO, M. (2004). *Guía para la formulación de proyectos de investigación.* Bogotá, Colombia: Alma Mater Magisterio.

ESQUEDA, José, L. E. (2004). *Fundamentos de procesamiento de imágenes.* California: Departamento Editorisal Universitaria

ETXEBERRIA, Jon. (2010). *Algoritmo de reconocimiento de forma y color para una plataforma robótica.* Donostia – San Sebastian. Obtenido de http://goo.gl/pruFha

HAYKIN, Simon, B. V. (2001). *Señales y Sistemas.* Mexico D.F: Limusa.

LOPEZ, J. L. (1984). *Métodos e hipótesis científicas.* México D.F: Trillas.

Métodos de Investigación. (s.f.). Recuperado el 15 de Marzo de 2012, Obtenido de http://www.doschivos.com/trabajos/biologia/116.htm

OPPENHEIM, Alan, A. S. (2004). *Señales y Sistemas.* Mexico D.F: Ingramex.

PALACIO, Enrique, F. R, L. L (2004). *Microcontrolador 16F84 - Desarrollo De Proyectos.* México DF: Alfa Omega.

PADRO, J. (Marzo de 2012). *Bases del concepto de "investigación aplicada"*. Obtenido de http://padron.entretemas.com/InvAplicada/index.htm

QUIRÓS, Fernando. (2006). *Navegación robótica basada en aprendizaje evolutivo de acciones mediante lógica difusa*. Valparaiso: Universidad Tecnica Federico. Obtenido de http://profesores.elo.utfsm.cl/~tarredondo/memorias/2005-memoria-fquiros.pdf

4. Anexos

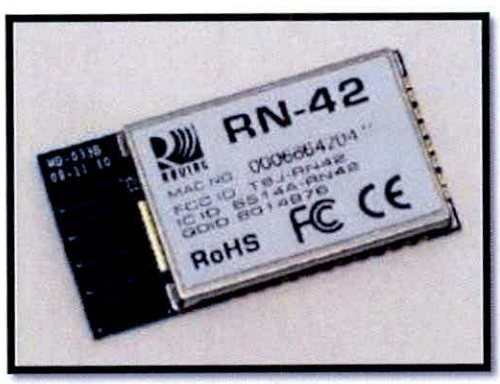
4.1. Anexo A: Datasheet Cámara inalámbrica

Parámetros de la Cámara:
Sensor de Imagen: 1/3" CMOS Sensor
Sistema: PAL/NTSC
resolución horizontal: 380TVL
Lente: 3.6mm Board Lens
Angulo de visión: 62°
Iluminación mínima: 1.5Lux/F1.5
Frecuencias de transmisión: ISM
2,400MHz~2,483MHz
Tipo de modulación: FM
Tensión de Salida: 50mW
Canales de transmisión: 4Ch
Rango de transmisión con línea de vista: 100m
Alimentación: DC8V, 500mA
Dimensiones: 23x23x23mm
Peso: 100g

Receptor:
Frecuencia del receptor: 2.4GHz
Canales del receptor: 4Ch
Tasa de grabación de video: 25/30fps
Formato de video: AVI
Resolución de grabación: 720 x 480 pixeles
Grabación de audio: Si
Tasa de transmisión de datos con el PC: 480bit/s con interfaz USB2.0
Compatibilidad con SO: Windows XP, VISTA 64bit, WINDOWS 7 64bit
Dimensiones: 102x28x12mm
Peso: 85g

4.2. Anexo B: Datasheet bluetooth

ROVING networks
WIRELESS FOR LESS RN-42/RN-42-N Data Sheet



Dimensiones: 42x16.5x5.6mm

Características:

Firmware v6.15

Clase Aprobado FCC 1 Radio Bluetooth Modem

Extremadamente pequeña radio - 0.15x0.6x1.9 "

Vínculo muy robusta, tanto en la integridad y la distancia de transmisión (100 m) - no más desbordamientos de búfer!

Bajo consumo de energía: 25 mA promedio

Frecuencia Hardy esquema de salto - funciona en ambientes hostiles de RF como WiFi, 802.11g, y Zigbee

Conexión cifrada

Frecuencia: 2.4 ~ 2.524 GHz

Voltaje de funcionamiento: 3.3V-6V

Comunicaciones de serie: 2400-115200bps

Temperatura de funcionamiento: -40 ~ +70 C

Antena incorporada

Digital I/O Characteristics

2.7V ≤ VDD ≤ 3.0V	Min	Typ.	Max.	Unit
Input logic level LOW	-0.4	-	+0.8	V
Input logic level HIGH	0.7VDD	-	VDD+0.4	V
Output logc level LOW	-	-	0.2	V
Output logic level HIGH	VDD-0.2	-	-	V
All I/O's (except reset) default to weakpull down	+0.2	+1.0	+5.0	uA

4.3. Anexo C: Especificaciones de Arduino Uno

Arduino Uno posee:

14 entrada/salida digitales, 6 pueden ser usados como salidas PWM

Los pin 0 y 1 pueden funcionar como RX y TX serial.

Posee 6 entradas analógicas

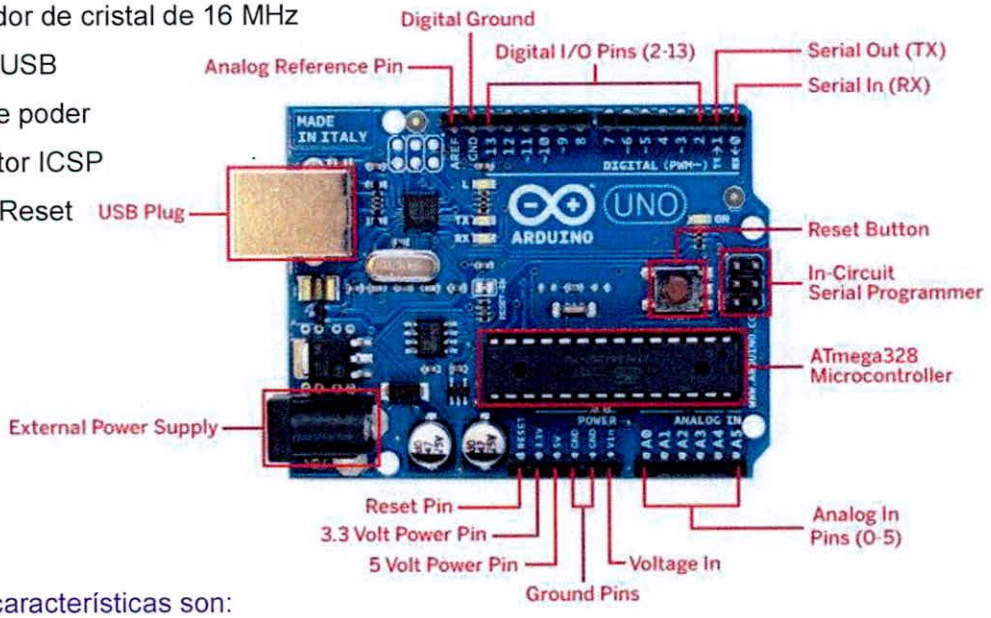
Un oscilador de cristal de 16 MHz

Conector USB

Un jack de poder

Un conector ICSP

Botón de Reset



Algunas características son:

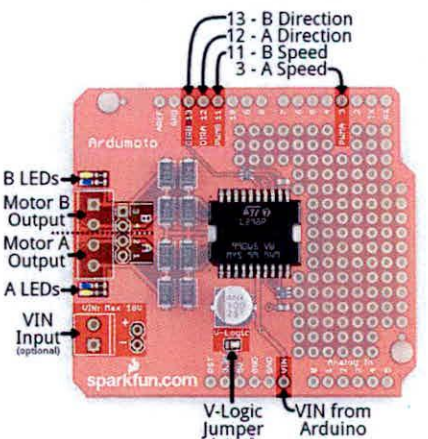
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (of which 0.5 KB used by bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

4.4. Anexo D: Datasheet Ardumoto

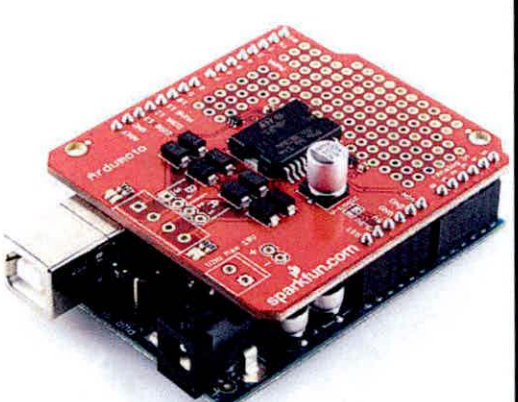
L298

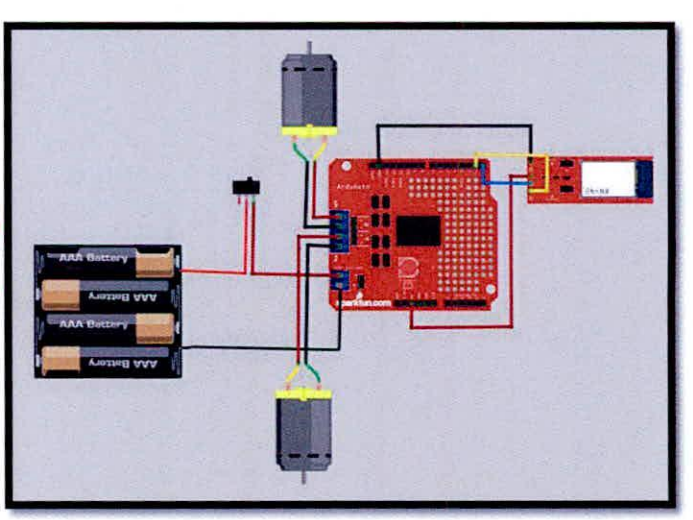
DUAL FULL-BRIDGE DRIVER

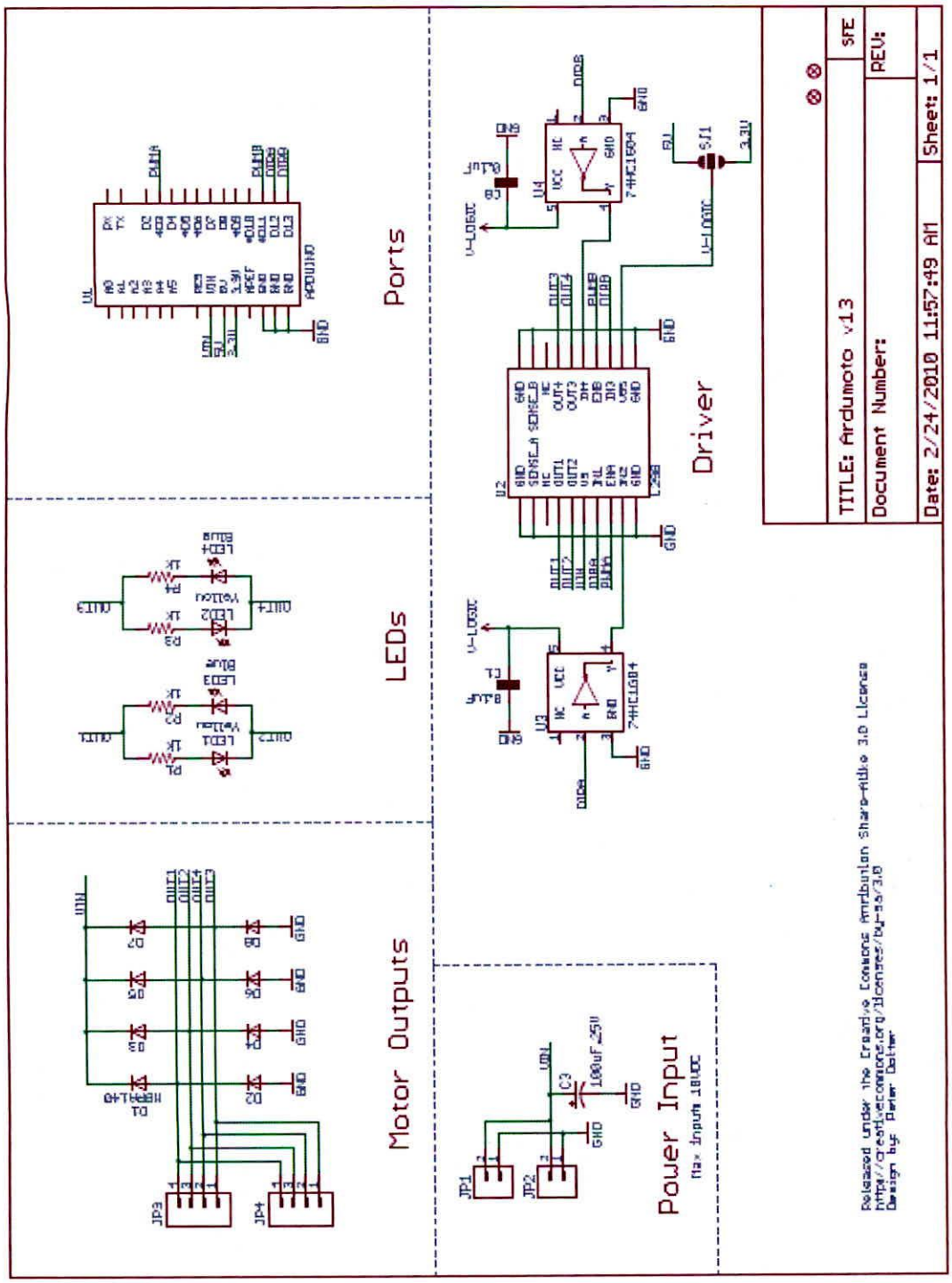
Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	−0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100µs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	−1 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



Arduino Pin	Ardumoto Shield Pin Label	Notes
3	PWM A	A PWM signal to control the speed of motor A. 0=off, 255=max speed.
11	PWM B	A PWM signal to control the speed of motor B. 0=off, 255=max speed.
12	DIR A	A digital signal to control the rotation direction of motor A (e.g. HIGH=CW, LOW=CCW).
13	DIR B	A digital signal to control the rotation direction of motor B (e.g. HIGH=CW, LOW=CCW).



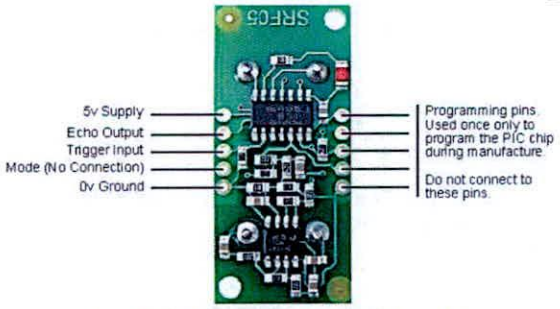



Ports
LEDs
Motor Outputs
Driver
Power Input
Max Inputs 18VDC
TITLE: Arduino v13
Document Number:
REV:
Date: 2/24/2010 11:57:49 AM
Sheet: 1/1

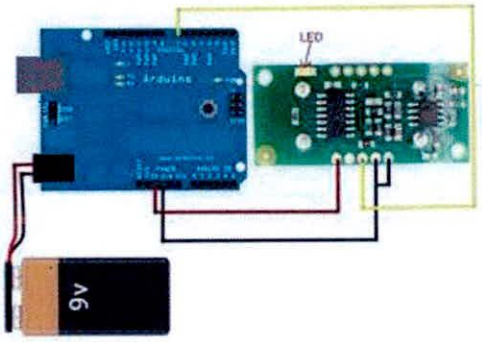
4.5. Anexo E: Datasheet Sensor SRF05

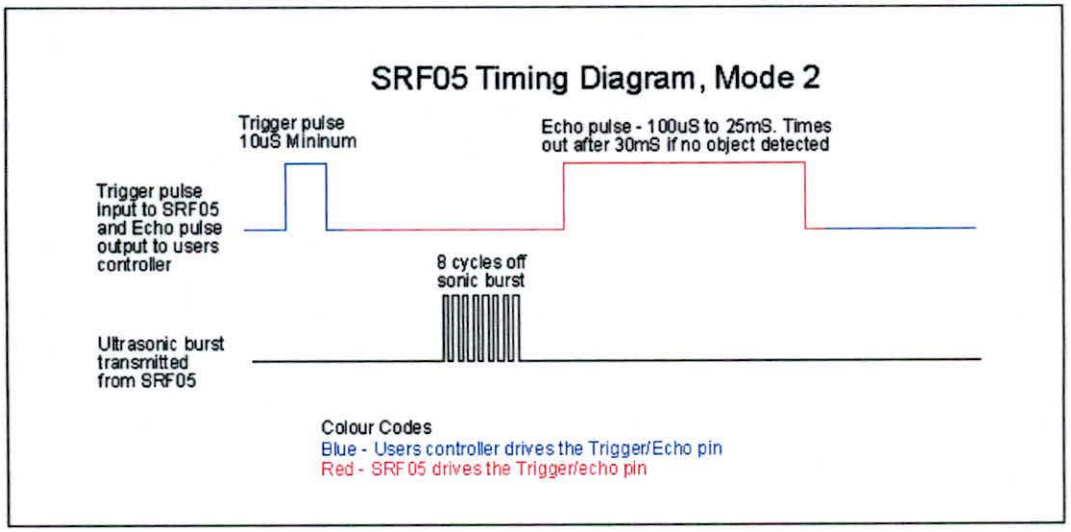
Características:

Rango - 1cm a 4m.
Voltaje - 5v, 4mA Típico.
Frecuencia - 40KHz. Tamaño - 43mm x 20mm x 17mm.
Cada pulso es una señal TTL positiva, con un acho proporcional al tamaño del objeto.
Tamaño: 43mm x 20mm x 17mm



Connections for 2-pin Trigger/Echo Mode (SRF04 compatible)

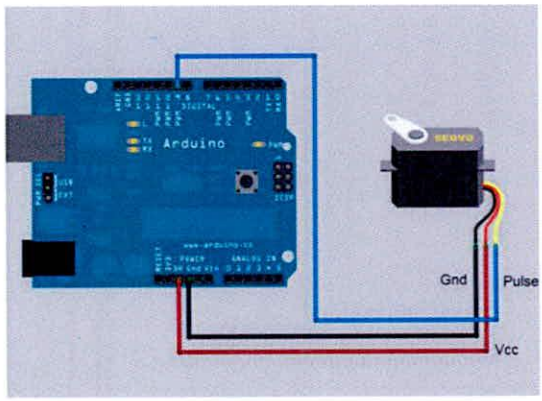




4.6. Anexo F: Datasheet Servomotor

Basic Information

Modulation:	Analog
Torque:	4.8V: 215.3 oz-in (15.50 kg-cm) 6.0V: 236.1 oz-in (17.00 kg-cm) 7.2V: 284.0 oz-in (20.45 kg-cm)
Speed:	4.8V: 0.16 sec/60° 6.0V: 0.14 sec/60°
Weight:	2.12 oz (60.0 g)
Dimensions:	**Length:** 1.65 in (41.9 mm) **Width:** 0.81 in (20.6 mm) **Height:** 1.56 in (39.6 mm)
Gear Type:	Metal
Rotation/Support:	Dual Bearings



Additional Specifications

Pulse Cycle:	20 ms
Pulse Width:	500-2100 µs

MANUAL

PROTOTIPO ROBÓTICO RECOLECTOR DE OBJETOS BASADO EN EL TRATAMIENTO DE IMÁGENES Y LÓGICA DIFUSA EN MATLAB

POR:

LEONARDO ANDRES PERDOMO DURAN
GERALDINE ESTHER STAN HERNANDEZ

DIRECTOR: JOSE RAMON IGLESIAS

MAYO 2014

PROTOTIPO ROBÓTICO RECOLECTOR DE OBJETOS BASADO EN EL TRATAMIENTO DE IMÁGENES Y LÓGICA DIFUSA EN MATLAB

LEONARDO ANDRES PERDOMO DURÁN
2008119041
GERALDINE ESTHER STAN HERNÁNDEZ
2008119057

DIRECTOR:
M.SC. ING. JOSÉ RAMÓN IGLESIAS GAMARRA PH.D.(C)

UNIVERSIDAD DEL MAGDALENA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA

SANTA MARTA D.T.C.H.
MAYO 2, 2014

Índice General

Índice General..... 3

1. Aspectos técnicos..... 4
   1.1. Datasheet Cámara inalámbrica..... 4
   1.2. Datasheet bluetooth..... 5
   1.3. Especificaciones de Arduino Uno..... 6
   1.4. Datasheet Ardumoto..... 7
   1.5. Datasheet Sensor SRF05..... 9
   1.6. Datasheet Servomotor..... 10
2. Requisitos para funcionamiento..... 11
3. Configuración del bluetooth..... 12
4. Interfaz de Usuario..... 15

1. Aspectos técnicos

1.1. Datasheet Cámara inalámbrica

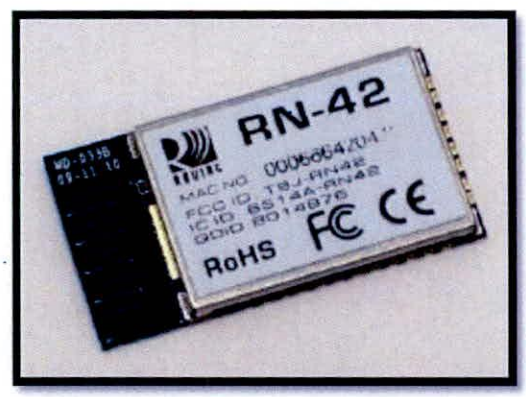
Parámetros de la Cámara:
Sensor de Imagen: 1/3" CMOS Sensor
Sistema: PAL/NTSC
resolución horizontal: 380TVL
Lente: 3.6mm Board Lens
Angulo de visión: 62°
Iluminación mínima: 1.5Lux/F1.5
Frecuencias de transmisión: ISM
2,400MHz~2,483MHz
Tipo de modulación: FM
Tensión de Salida: 50mW
Canales de transmisión: 4Ch
Rango de transmisión con línea de vista: 100m
Alimentación: DC8V, 500mA
Dimensiones: 23x23x23mm
Peso: 100g

Receptor:
Frecuencia del receptor: 2.4GHz
Canales del receptor: 4Ch
Tasa de grabación de video: 25/30fps
Formato de video: AVI
Resolución de grabación: 720 x 480 pixeles
Grabación de audio: Si
Tasa de transmisión de datos con el PC: 480bit/s con interfaz USB2.0
Compatibilidad con SO: Windows XP, VISTA 64bit, WINDOWS 7 64bit
Dimensiones: 102x28x12mm
Peso: 85g

1.2. Datasheet bluetooth

ROVING networks
WIRELESS FOR LESS RN-42/RN-42-N Data Sheet

Dimensiones: 42x16.5x5.6mm



Características:

Firmware v6.15

Clase Aprobado FCC 1 Radio Bluetooth Modem

Extremadamente pequeña radio - 0.15x0.6x1.9 "

Vínculo muy robusta, tanto en la integridad y la distancia de transmisión (100 m) - no más desbordamientos de búfer!

Bajo consumo de energía: 25 mA promedio

Frecuencia Hardy esquema de salto - funciona en ambientes hostiles de RF como WiFi, 802.11g, y Zigbee

Conexión cifrada

Frecuencia: 2.4 ~ 2.524 GHz

Voltaje de funcionamiento: 3.3V-6V

Comunicaciones de serie: 2400-115200bps

Temperatura de funcionamiento: -40 ~ +70 C

Antena incorporada

Digital I/O Characteristics

2.7V ≤ VDD ≤ 3.0V	Min	Typ.	Max.	Unit
Input logic level LOW	-0.4	-	+0.8	V
Input logic level HIGH	0.7VDD	-	VDD+0.4	V
Output logc level LOW	-	-	0.2	V
Output logic level HIGH	VDD-0.2	-	-	V
All I/O's (except reset) default to weakpull down	+0.2	+1.0	+5.0	uA

1.3. Especificaciones de Arduino Uno

Arduino Uno posee:

14 entrada/salida digitales, 6 pueden ser usados como salidas PWM

Los pin 0 y 1 pueden funcionar como RX y TX serial.

Posee 6 entradas analógicas

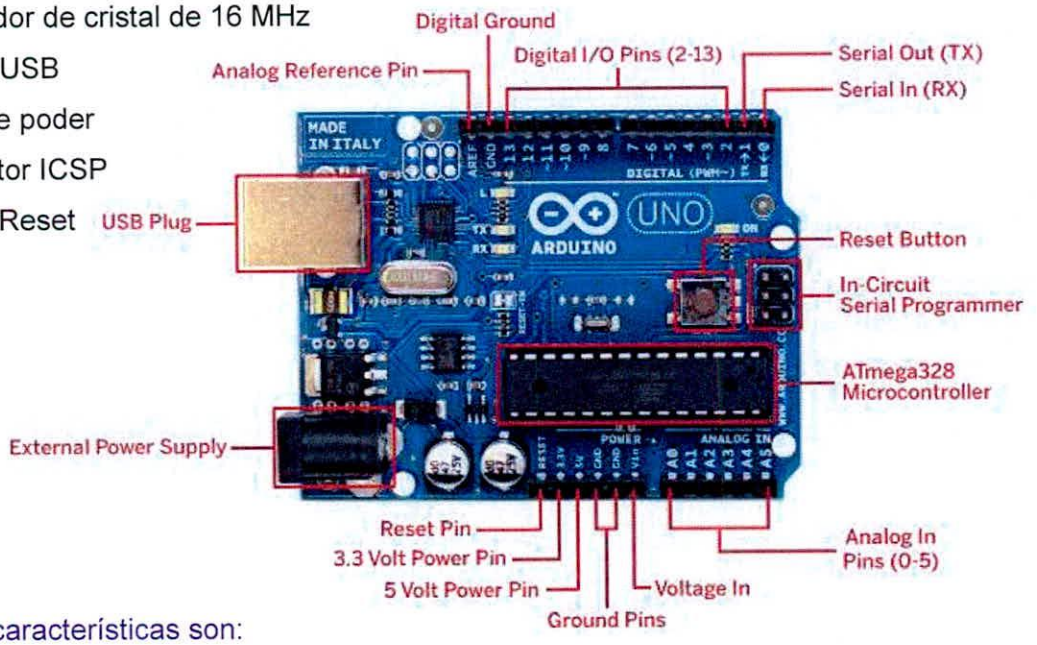
Un oscilador de cristal de 16 MHz

Conector USB

Un jack de poder

Un conector ICSP

Botón de Reset



Algunas características son:

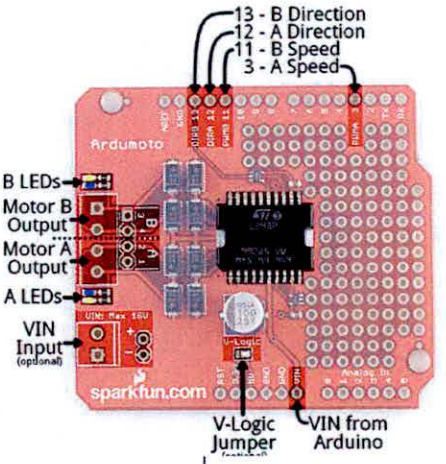
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (of which 0.5 KB used by bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

1.4. Datasheet Ardumoto

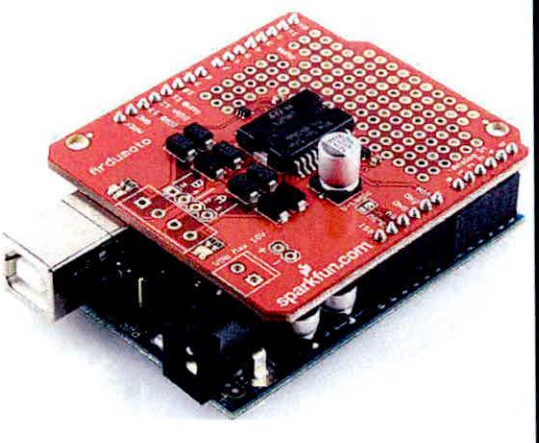
L298

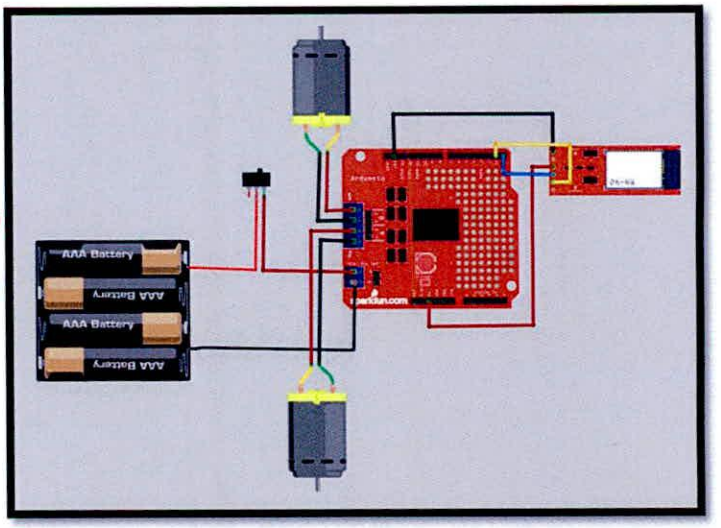
DUAL FULL-BRIDGE DRIVER

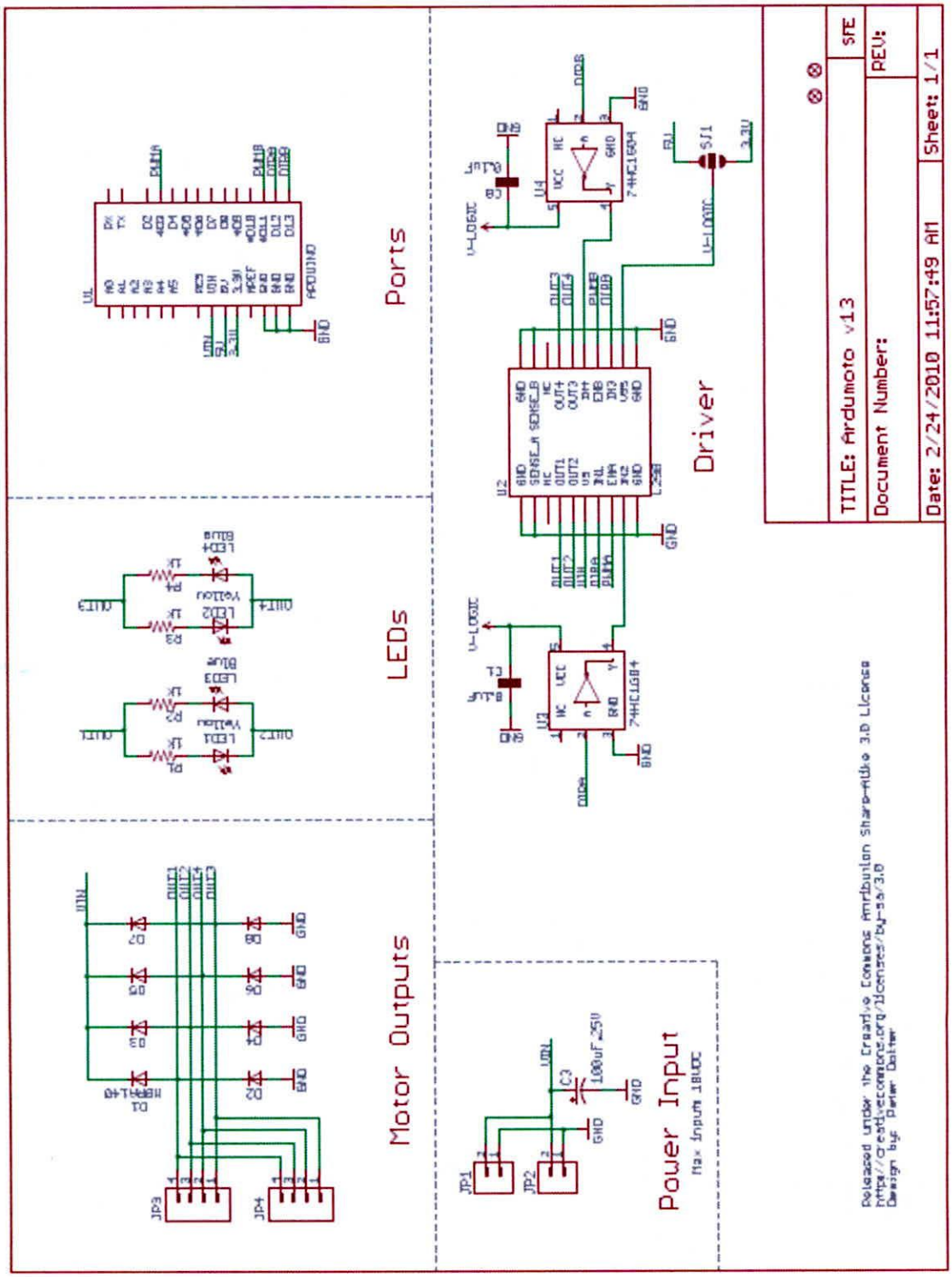
Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	−0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100µs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	−1 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



Arduino Pin	Ardumoto Shield Pin Label	Notes
3	PWM A	A PWM signal to control the speed of motor A. 0=off, 255=max speed.
11	PWM B	A PWM signal to control the speed of motor B. 0=off, 255=max speed.
12	DIR A	A digital signal to control the rotation direction of motor A (e.g. HIGH=CW, LOW=CCW).
13	DIR B	A digital signal to control the rotation direction of motor B (e.g. HIGH=CW, LOW=CCW).



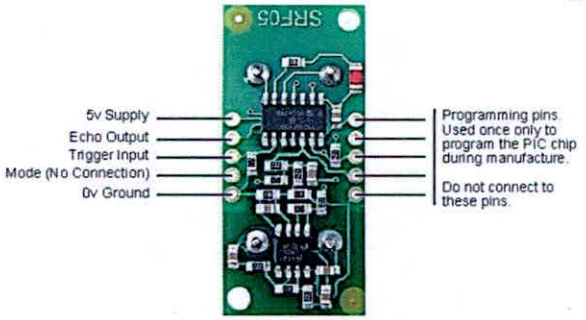



Ports
LEDs
Motor Outputs
Driver
Power Input
Max Input 18VDC
TITLE: Ardumoto v13
Document Number:
Date: 2/24/2010 11:57:49 AM
Sheet: 1/1

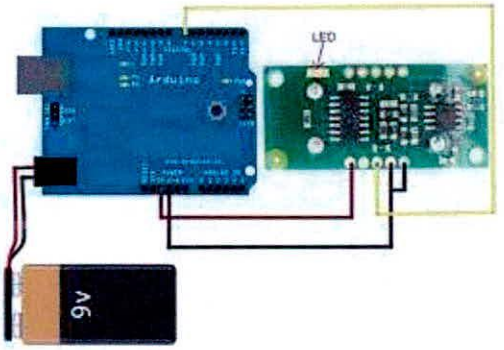
1.5. Datasheet Sensor SRF05

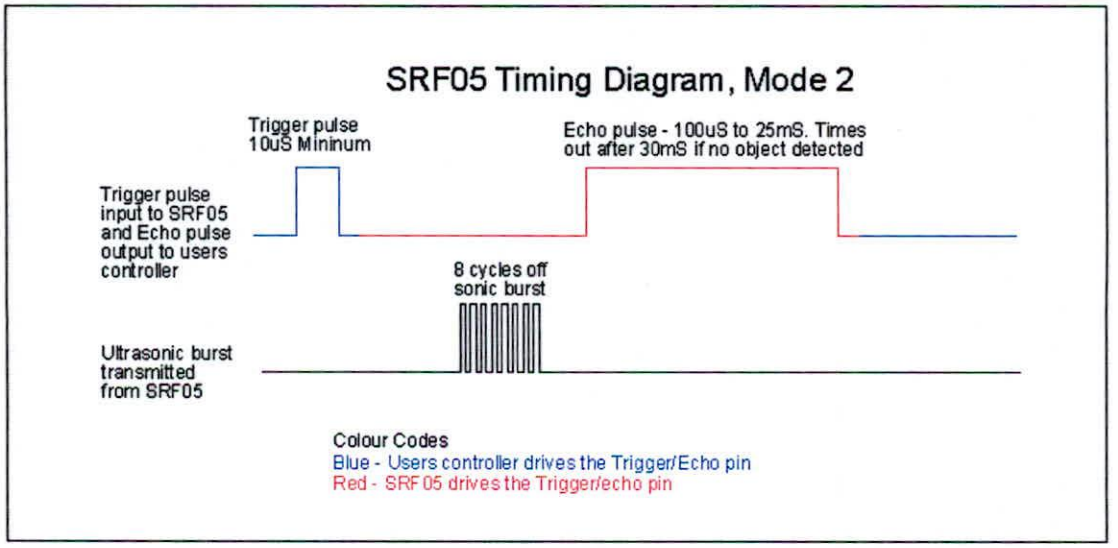
Características:

Rango - 1cm a 4m.
Voltaje - 5v, 4mA Típico.
Frecuencia - 40KHz. Tamaño - 43mm x 20mm x 17mm.
Cada pulso es una señal TTL positiva, con un acho proporcional al tamaño del objeto.
Tamaño: 43mm x 20mm x 17mm



Connections for 2-pin Trigger/Echo Mode (SRF04 compatible)

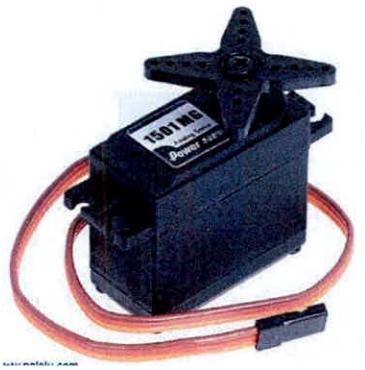


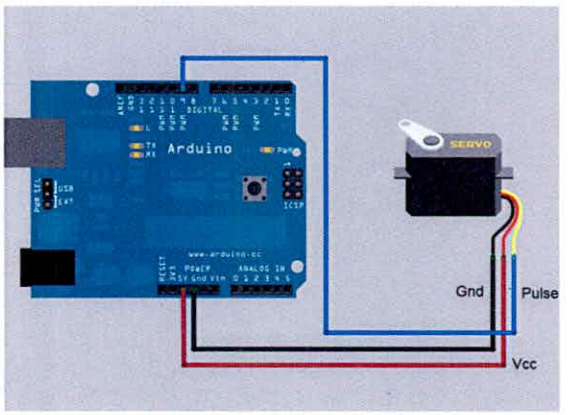


1.6. Datasheet Servomotor

Basic Information

Modulation:	Analog
Torque:	4.8V: 215.3 oz-in (15.50 kg-cm) 6.0V: 236.1 oz-in (17.00 kg-cm) 7.2V: 284.0 oz-in (20.45 kg-cm)
Speed:	4.8V: 0.16 sec/60° 6.0V: 0.14 sec/60°
Weight:	2.12 oz (60.0 g)
Dimensions:	**Length:** 1.65 in (41.9 mm) **Width:** 0.81 in (20.6 mm) **Height:** 1.56 in (39.6 mm)
Gear Type:	Metal
Rotation/Support:	Dual Bearings





Additional Specifications

Pulse Cycle:	20 ms
Pulse Width:	500-2100 µs

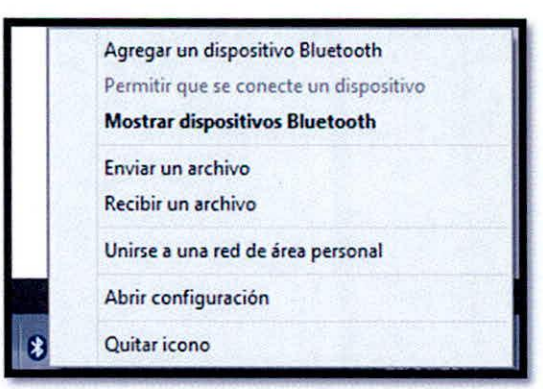
2. Requisitos para funcionamiento

I.	Sistema Operativo:	Windows 7 – Windows 8, 64 Bits.
II.	Memoria RAM:	2 Gb
III.	Dispositivo Bluetooth:	2.0 (incluido)
IV.	Puertos USB 2.0:	1
V.	X	
VI.	Matlab Version:	R 2013 b
VII.	CD de instalación de Driver para cámara inalámbrica	Si (incluido)
VIII.	Baterias:	6 de 1.5 V (incluido) 1 de 9 V (incluido)

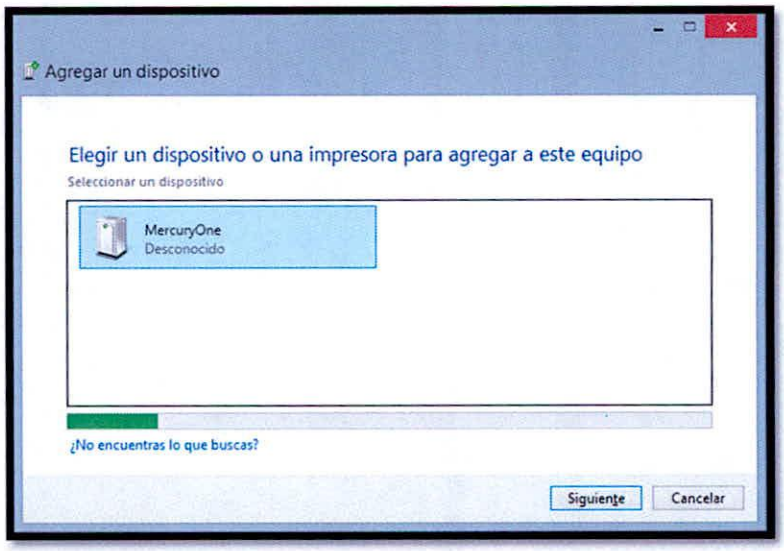
3. Configuración del bluetooth

Para establecer la comunicación inalámbrica por primera vez con el modulo bluetooth desde el PC, se procede de la siguiente manera:

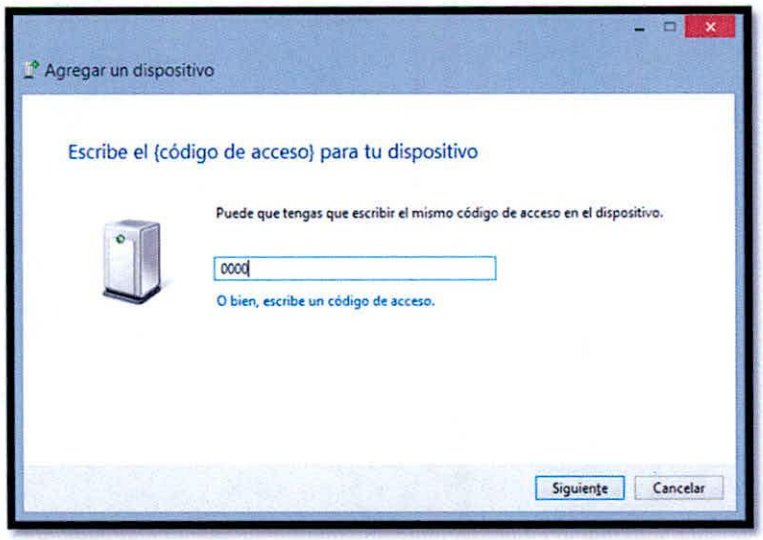
Dar click derecho en dispositivos bluetooth que se encuentra en la barra de herramientas de Windows y luego en Agregar un dispositivo Bluetooth.



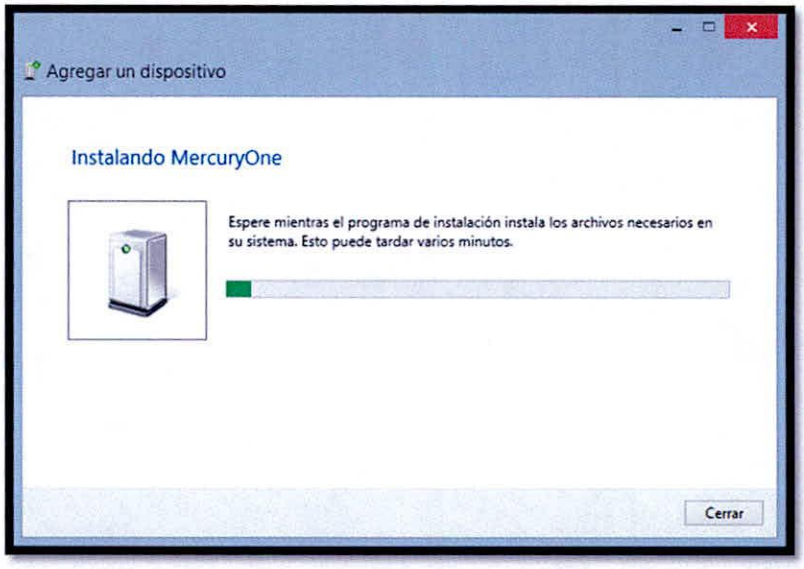
Aparecerá la siguiente ventana indicándonos el dispositivo bluetooth encontrado, en este caso MercuryOne es el nombre del módulo bluetooth del robot.



Luego la instalación pedirá el código para acceder al dispositivo, éste es: 0000.



Luego que reconozca el código de acceso instalará satisfactoriamente el dispositivo para ser utilizado con el Matlab.


Agregar un dispositivo
Instalando MercuryOne
Espere mientras el programa de instalación instala los archivos necesarios en su sistema. Esto puede tardar varios minutos.
Cerrar

4. Interfaz de Usuario

Instrucciones:

I. Guardar la carpeta Mercury Rover One en una carpeta local del computador a utilizar

II. Desde el Matlab seleccionar la carpeta Mercury Rover One y dar click en el archivo mro_gui_v04.fig

Se abrirá la siguiente ventana.

III. Dando click en Conectar MRO, se procede a establecer la comunicación por bluetooth con el robot; el cual debe estar encendido, de igual forma la cámara.

IV. Conectar el módulo receptor de la cámara al computador e instalar el driver desde el CD si no se instaló de forma automática.

V. Dar click en Encender Cámara.

VI. Escoger el color a reconocer, se puede cambiar el color mientras se procesa, pero no es recomendable; ya que esto podría causar errores en el procesamiento.

VII. Dar click en Procesar Imágenes

VIII. Para terminar el procesamiento o cambiar el color se desconecta MRO y se apaga la Cámara.

IX. Mediante se vayan procesando las imágenes y presentando en la interfaz, se puede acceder a la información de la lógica difusa del sistema, para eso es necesario dar click en el botón en forma de gráfica. Se puede tantear la velocidad que se le va a suministrar a los motores por medio del slider o simplemente visualizar como se realiza de forma automática mediante el robot vaya detectando el objeto.