UNIVERSIDAD POLITECNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERIA ELECTRONICA

IMPLEMENTACION DE UN ROBOT HUMANOIDE CON RECONOCIMIENTO DE OBJETOS POR COLOR Y FORMA

Trabajo de titulación previo a la obtención del
Título de Ingeniero Electrónico

AUTORES:

César Arturo Valdiviezo Romero

Rurik Alfredo Valladares Yanqui

TUTOR:

Ing. Mónica María Miranda Ramos

Guayaquil-Ecuador

2022

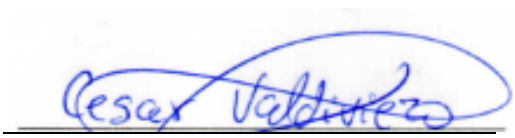
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, César Arturo Valdiviezo Romero con documento de identificación N° 0930095849 y Rurik Alfredo Valladares Yanqui con documento de identificación N° 0931423123; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 13 de septiembre del año 2022

Atentamente,



César Arturo Valdiviezo Romero
0930095849

Rurik Alfredo Valladares Yanqui
0931423123

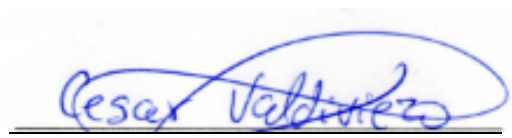
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, César Arturo Valdiviezo Romero con documento de identificación No. 0930095849 y Rurik Alfredo Valladares Yanqui con documento de identificación No. 0931423123, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de grado titulado: Implementación de un robot humanoide con reconocimiento de objetos por color y forma, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 13 de septiembre del año 2022

Atentamente,



César Arturo Valdiviezo Romero
0930095849

Rurik Alfredo Valladares Yanqui
0931423123

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Mónica María Miranda Ramos con documento de identificación N° 0917271785, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: Implementación de un robot humanoide con reconocimiento de objetos por color y forma, realizado por César Arturo Valdiviezo Romero con documento de identificación N° 0930095849 y por Rurik Alfredo Valladares Yanqui con documento de identificación N° 0931423123, obteniendo como resultado final el trabajo de titulación bajo la opción de trabajo de grado que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 13 de septiembre del año 2022

Atentamente,

Ing. Mónica María Miranda
Ramos
0917271785

DEDICATORIA

Dedico este proyecto en especial a Dios, a mis padres y a mis hermanos por el apoyo incondicional y la motivación que me daban siempre para seguir adelante y me han dado el valor y el coraje de poder llegar a ser un profesional. **Rurik Alfredo Valladares Yanqui.**

Les dedico mi tesis con todo mi cariño a mis padres por apoyarme en darme una carrera para mi futuro y por creer en mi capacidad, a pesar de los momentos difíciles que hemos pasado me han brindado su cariño y amor.

También dedico este trabajo a mis compañeros y amigos, quienes son esperar nada a cambio compartieron sus conocimientos, alegrías y a todas aquellas personas que durante estos años estuvieron apoyándome y lograron que este sueño se haga realidad. **César Arturo Valdiviezo Romero.**

AGRADECIMIENTO

Agradezco a mi familia por el apoyo incondicional, en la cual a mi papá me ayudó mucho en los primeros semestres pagándome las pensiones, a mi mamá por darme consejos que nunca me rinda y a mi tutora Ing. Mónica Miranda por sus consejos y su apoyo incondicional. **Rurik Alfredo Valladares Yanqui**.

Agradezco a mi familia, por haberme dado la oportunidad de formarme en esta prestigiosa universidad y haber sido mi apoyo durante todo este tiempo.

De manera especial a mi tutora de tesis, por haberme guiado, no sólo en la elaboración d este trabajo de titulación, sino a lo largo de mi carrera universitaria y haberme brindado el apoyo para desarrollarme profesionalmente y seguir cultivando mis valores. **César Arturo Valdiviezo Romero**

RESUMEN

AÑO	ALUMNOS	DIRECTOR DE PROYECTO TÉCNICO	TEMA DE PROYECTO TÉCNICO
2022	César Arturo Valdiviezo Romero Rurik Alfredo Valladares Yanqui		Implementación de un robot humanoide con reconocimiento de objetos de acuerdo a su color y forma

La robótica es la rama de la ingeniería mecánica, electrónica y ciencias informática que se encarga del diseño, operación, estructura y aplicación de los robots. La palabra robot se traduce como trabajo forzado o trabajador, nace de la palabra checa “robota”.

En el siguiente proyecto de titulación se diseñó un reconocimiento de objetos por su color y forma con cobertura de red Wifi de esta manera tendríamos una dirección IP fija para el software VNC View en lo cual nos ayuda a visualizar en nuestro dispositivo.

Por medio de un módulo Bluetooth nos podríamos comunicar a través de la Raspberry para controlar inalámbricamente los movimientos del robot Bioloid GP ya que por medio de cable tipo B nos impediría realizar las rutinas libremente.

En este trabajo se utilizaron las herramientas el Robot Motion y Robot Task, en la herramienta Robot Motion nos ayuda en guardar los parámetros de velocidad, el torque y el control de fuerza y el tiempo de ejecución que vamos a utilizar en la programación del robot Bioloid. Sobre en la herramienta Robot Task es donde vamos a realizar la programación operacional general del robot Bioloid.

Utilizamos un sensor de color RSC-10 el cual hace posible el reconocimiento de los colores con un rango de alcance de entre 6 a 18 mm, al momento de detectar los colores hicimos que el robot haga una rutina de aplausos, por cada color diferente que la herramienta sensa, el robot dará ciertas cantidades de aplausos.

Para el reconocimiento de los objetos se utilizó una cámara ESP32-CAM que a través de una programación en el software Arduino nos ayuda a conectamos de forma inalámbrica con la Raspberry PI4 por medio de una dirección IP fija que nos permite

transmitir imágenes con mayor resolución de 1600 x 1200 pixeles y resolución de video 1080p30, 720p60 y 6040x480p90.

Palabras claves: Robot Bioloid, Robot Task, Robot Motion, VNC View, Cámara ESP32-CAM, Sensor RSC-10, Raspberry PI4.

ABSTRACT

YEAR	STUDENTS	Technical Project Manager	Item of Project of titulation
2022	César Arturo Valdiviezo Romero Rurik Alfredo Valladares Yanqui		Implementation of a humanoid robot with object recognition by color and shape

Robotis is the branch of mechanical engineering, electronic and computer sciences, that is in charge of the design, structure and application of robots.

Word robot is traslate as forced work, and it is born from the czech word "robota".

In the following titling project, a recognition of objects by thin color and shape with wi-fi network coverage was designed, in this way we would have a fixed IP address for the VNC view software, with wich, it help us to visualize our device.

Through this bluetooth module we could communicate with the raspberry to wiredenly central the movements of bioloid GP robot, because by means of a type B wire, it would prevent us from perfoming the routines freely.

In this work, the robot motion and robot tools were used, Robot Motion tool help us to save the parameters of speed, torque and force control and the execution time, that we are going to use in the robot Bioloid programming. About the Robot task is where we are going to carry out the general operational programming of the Bioloid Robots.

We use an RSC-10 color sensor, this makes it possible to recognize colors with a range between 6 to 18 mm, at time to detecting the colors, we made the robot do a chaps routine, for each different color use by the tool, the robot with give certain amount of chapes.

For the recognition of objects it was used an ESP32-CAM camera, through a programming in the software Arduino, help us to connect wiresly with the fix Raspberry PL4 address, that allow us to transmit images with higher resolution of 1600 x 1200 pixeles and a video resolution of 1080 p 30, 720 p 60 and 6040 x 480 p 90.

Keywords: Robot Bioloid, Robot Task, Robot Motion, VNC View, ESP32-CAM Camera, RSC-10 Sensor, Raspberry PI4.

ÍNDICE GENERAL

ÍNDICE DE IMAGENES

ÍNDICE DE GRÁFICOS

INTRODUCCIÓN

El desarrollo paulatino de la tecnología hace necesaria que los conocimientos se desarrollen a la par, para logarlo es fundamental contar con los beneficios que se pueden obtener de las nuevas tecnologías. La incursión de los robots hace necesario la capacidad para controlarlos de forma eficiente.

Este estudio tiene la finalidad de ensamblar un robot humanoide para el reconocimiento de objetos de acuerdo a su color y forma, el cual puede ser de gran utilidad para aplicar los conocimientos y la estructura necesaria para su posterior implementación, puede ser una base para futuros desarrollos que permitan que estos mejoren algunos aspectos de nuestra vida cotidiana.

El desarrollo de la robótica en la actualidad no contempla límites y continuamente el desarrollo de la tecnología se supera de forma continua, esto hace necesario que

- Ensamblar un robot humanoide para el reconocimiento de objetos de acuerdo a su color y forma

1. El problema

1.1. Antecedentes

El presente proyecto se desarrolla gracias a los conocimientos adquiridos en el área de robótica y automatización, este se encuentra enfocado en los avances del campo de la robótica, con el objetivo de obtener avances de la robótica en la Universidad Politécnica Salesiana, para así motivar a los estudiantes de la institución a seguir realizando avances tecnológicos en los robots humanoides, con distintas finalidades u objetivos que se tengan en mente, los mismos que serán donados a la institución para que los estudiantes del área de robótica realicen prácticas y programación de rutinas para efectuar mejoras o avances tecnológicos a los robots.

El robot humanoide contará con la capacidad de realizar rutinas programadas previamente en lenguaje de programación Python, se realizarán simulaciones por medio de software para verificar la validación de dicha programación. El mismo contará con 16 servomotores los mismos que estarán conectados a un módulo controlador de servomotores.

El desarrollo del sistema se realiza en tres etapas:

- Diseño e implementación de una metodología de detección e identificación del objeto de interés usando procesamiento digital de imágenes y una cámara ESP.
- Identificación de colores por medio de un sensor de color RCS-10.
- Implementación del sistema de monitoreo en tiempo real.

En la primera etapa se analizan las imágenes en modo video en tiempo real, gracias al programa realizado en el software RoboTask y a la cámara ESP 31-Cam, el robot reconoce el objeto frente a él, esto se podrá visualizar en la pantalla del computador.

Asimismo, previo a la rutina programada en el VNC, se acerca a una distancia de 18mm un objeto de color al sensor RCS-10 y al detectar el color del prototipo hará una rutina

de aplausos. El proyecto tiene un total de 6 practicas, las cuales se dividen en 2 practicas comunes y 4 de practica diferencial.

1.2. Importancia y Alcances

En continuo avance de la tecnología, hace necesario que el diseño se adapte a las nuevas características, con la finalidad de que las nuevas versiones puedan ser aprovechadas y de esa forma obtener todos los beneficios de la tecnología. El presente proyecto busca implementar un robot humanoide con la capacidad de reconocer objetos acordes a su color y forma, de esta forma se busca contribuir al aprovechamiento de las tecnologías existentes, incentivando la evolución de los sistemas que lo controlan.

1.3. Delimitación

Campo: Robótica.

Área: Ciencias Aplicadas.

Aspecto: Implementación de un robot humanoide con capacidad de reconocer objetos acordes a su color y forma.

1.3.1. Temporal

El desarrollo y programación del robot humanoide se llevó a cabo en el año 2022.

1.3.2. Espacial

El presente trabajo se elaboró en la ciudad de Guayaquil, teniendo una aplicación específica en las instalaciones de la Universidad Politécnica Salesiana, en especial en los Laboratorios de Control que se encuentran ubicados en el bloque E de la institución con sede en Guayaquil.

1.3.3. Académica

Por medio del desarrollo de este proyecto será posible aplicar los conocimientos técnicos que han sido aprendidos a lo largo de la carrera. Estos conocimientos fueron adquiridos durante las clases regulares, seminarios, foros y educación autónoma, en especial en materias como informática Industrial.

El presente proyecto beneficiará a los estudiantes de la carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana – Sede Guayaquil, ya que, el mismo les permitirá poner en práctica los conocimientos sobre las tecnologías y robótica.

1.4. Innovación

Al implementar un robot humanoide con la capacidad de reconocer objetos de acuerdo a su color y forma, permitirá a los estudiantes una aplicación de todos los conocimientos adquiridos, además de permitir que cuenten con un mecanismo para realizar las practicas necesarias durante sus estudios.

1.5. Objetivos

1.5.1. Objetivo general

- Ensamblar un robot humanoide para el reconocimiento de objetos por color y forma.

1.5.2. Objetivos específicos

- Ensamblar la estructura de un robot humanoide.
- Planificar un manual de prácticas.
- Programar la rutina del robot la cual se basa en el reconocimiento de objetos de acuerdo con su color y forma usando el sensor del color RCS-10.

2. Fundamentos teóricos

Inteligencia artificial

Para (Silva Segovia, 2010) la inteligencia artificial y la robótica son áreas que tienen futuro de aplicación, ya que en el mercado actual se pueden observar la gran gama de robots que se manejan mediante la inteligencia artificial.

Según (Minsky, 2022) la inteligencia artificial es la ciencia de hacer que las maquinas realicen cosas que requerirían inteligencia como si las hubiera hecho un humano.

Por otro lado, para (Rouhiainen, 2018) la IA es la habilidad que tienen los ordenadores para realizar actividades que normalmente requieren inteligencia humana, además de eso el mismo autor menciona que la inteligencia artificial es la capacidad que tienen las maquinas para usar algoritmos, para así aprender datos y utilizarlos al momento de tomar una decisión.

Robot

Según Mikell Groover como se citó en (Garcia, Alvarez, & Cava, 2003) define al robot como una maquina programable de propósito general, que posee características antropomórficas.

(Barrientos, Peñin, Balaguer, & Aracil, 1997) afirman que para el mercado japones un robot es un dispositivo mecánico dotado de articulaciones movibles destinadas a la manipulación de objetos.

Es decir que para los autores un robot es una maquina o dispositivo programable diseñado con articulaciones antropomórficas, que están destinadas a la manipulación de objetos sin averiarlos y con una precisión superior a la de los humanos.

Al juntar a la robótica con la inteligencia artificial se puede lograr el manejo preciso de robots, incluyendo el movimiento de sus extremidades, esto para realizar el ensamble de piezas, actividades peligrosas para el humano o actividades de alta precisión. El

presente robot deberá ser capaz de reconocer los objetos por su color o forma usando el sensor de color RCS10 y una cámara que estará conectada al Raspberry Pi.

2.1. Propuesta de marco teórico

2.1.1. Phyton

En los últimos años el software libre ha evolucionado para convertirse en un sistema que permite a los distintos usuarios el aprovechar estas ventajas para desarrollar herramientas que les permitan aprovechar todas estas ventajas. En este sentido Challenger et at (2019), mencionan que este lenguaje puede garantizar el acceso a programas de código de fuente abierto, que puede emplearse en cualquier tipo de ambiente y permitir las modificaciones por parte de los distintos usuarios.

Entre las ventajas de este tipo de software libre se encuentran que el usuario puede modificarse en caso de ser necesario para garantizar que se adapte a sus necesidades. Dese su lanzamiento en el año de 1991 por parte de Guido van Rossum, este lenguaje se ha situado entre los principales lenguajes usados en todas partes del mundo.

Los lenguajes de programación son una herramienta básica en la construcción de todo tipo de programas informáticos, y Python ha ganado terreno sobre las comunidades de software libre, debido a las ventajas que ofrece sobre todo en la comunidad científica y en la educación quienes son sus principales usuarios debido a la facilidad para ser adaptado a las necesidades.

Imagen 1 Python Software

Entre las herramientas que contiene este programa se encuentran la siguientes:

Intérprete al vuelo

El programa Python cuenta con varias herramientas, que permitan que los usuarios puedan realizar las tareas por medio de la creación diferentes módulos

Librería estándar

Entre las características más importantes de Python se encuentra la librería estándar, que permite cubrir la mayoría de las necesidades para la elaboración de programas, cubriendo una gran cantidad de necesidades que pueden acceder a una gran cantidad de alternativas para facilitar la programación de los trabajos. Entre los principales se encuentra:

- Cadenas.
- Funciones numéricas y matemáticas.
- Estructura de datos.
- Compresión de datos.
- Formatos de archivo.
- Criptografía.
- Servicios de sistemas operativos.
- Comunicación entre procesos.
- Manejo de datos de internet.
- Servicios multimedia.
- Manejo de excepciones.

Todo este conjunto de datos permite a los usuarios de Python el gestionar las librerías convirtiéndola en una de las más completas que existen en la actualidad, que en muchas ocasiones pude ser comparable a las librerías de sus análogos como .NET o Java.

2.1.1.1. Rendimiento

En lo que se refiere al rendimiento de Python, mantiene el estado como uno de las mejores opciones para garantizar una fluidez adecuada, y a diferencia de sus principales competidores, dentro de las principales características de este lenguaje de programación se ha implementado en su totalidad en lenguaje C, de esta forma es posible que sus funciones sean mucho más eficientes además debido a la posibilidad para compilar el código en base a bytecodes que es una herramienta parecida (Challenger, Díaz, & Becerra, 2019).

2.1.1.2. Extensiones

entre las características mas importantes que posee Python se encuentra la capacidad de reutilizar códigos escritos en lenguajes C y C++. Asimismo, se encuentran mecanismos que realizan de forma eficiente las tareas de envolver funciones y clases hechas en estos lenguajes, entre estos podemos encontrar a Boost.Python, Sip y Shiboken. (MCkibre, 2020)

La importancia de esta integración es relevante, ya que las bases de código en lenguajes como C y C++ son las más grandes disponibles por el software libre hoy en día, y permiten no tener que duplicar código ya existente.

2.1.1.3. Licencia

El programa Python fue lanzado al mercado bajo una licencia propia denominada Python Licence, que ha sido certificada por Open Source, adema de ser compatible con GPL (por sus siglas en inglés, GNU Public Licence).

Una de las principales diferencias de esta licencia con respecto a la GPL, radica en que no existe una restricción copyleft, es decir que se poder elaborar, producir y distribuir programas de forma libre sin necesidad de entregar el código fuente, esto significa que puede usarse Python tanto para hacer software libre como software privativo (MCkibre, 2020)

2.1.1.4. Herramientas

Según (Challenger, Diaz, & Becerra, 2014) es vital que para un lenguaje de programación se cuenten con herramientas que la hagan más productivas, es por eso por lo que presentan a las herramientas más importantes:

Implementaciones

Python posee varios implementos, entre los mas importantes se encuentra CPython, esta se encuentra elaborada en lenguaje C y es considerada las más estable de todas. Al igual que esta, existen otras similares como IronPython que está escrita en lenguaje C# y fue diseñada por la plataforma .NET, Jython diseñada por Java, PyPy diseñada por Python para correr en diversas plataformas y Tinypy diseñada de forma minimalista para ocupar 64Kb de memoria.

Entornos de Desarrollo Integrado

Debido al éxito que ha tenido Python en las plataformas Unix, se ha elaborado una gran variedad de entornos de desarrollo integrado, entre los principales entronos se encuentra IDLE (Integrated Development Environment) desarrollado por Python y es un entorno de desarrollo simple, además cuenta con un interprete de vuelo desarrollado por Tkinter, este entorno es ideal para pequeñas aplicaciones, esto gracias a su sencillez y su distribución junto a los paquetes de Python para los sistemas operativos más comunes.

Servidores de Aplicaciones

Para (Rossum & Drake, 2010) el servicio web se ha convertido en el centro de la vida digital, por lo que menciona que es importante que el lenguaje cuente con un soporte

técnico en los distintos servidores de aplicaciones web, ya que al no contar con uno establecido el lenguaje puede ser considerado obsoleto.

2.2. **Robot Humanoide**

2.2.1. Características

La (Revista de robots, 2022) menciona que un robot humanoide está diseñado para reproducir la forma de los brazos y pies humanos, además estos pueden ser bípedos o desplazarse por medio de ruedas. Además, existen robots más sofisticados que buscan el hiperrealismo, intentando simular la piel, las expresiones y gestos de un humano

El diseño de los robots humanoides puede tener fines funcionales, como interactuar con herramientas y entornos humanos, con fines experimentales, como el estudio de la locomoción, o para otros fines.

Según el portal (Hisour, 2022) en la actualidad se investigan y desarrollan robots humanoides para realizar tareas humanas como asistencia personal, como por ejemplo la ayuda de ancianos, enfermos, trabajos de alto riesgo, etc. además de esto mencionan que los robots son adecuados para ciertas vocaciones como por ejemplo el ensamblaje automático de automóviles, esto gracias al diseño que le permite manejar herramientas y operar equipos diseñados para humanos, es otras palabras, un robot humanoide con un software adecuado puede elaborar cualquier actividad que un humano requiera.

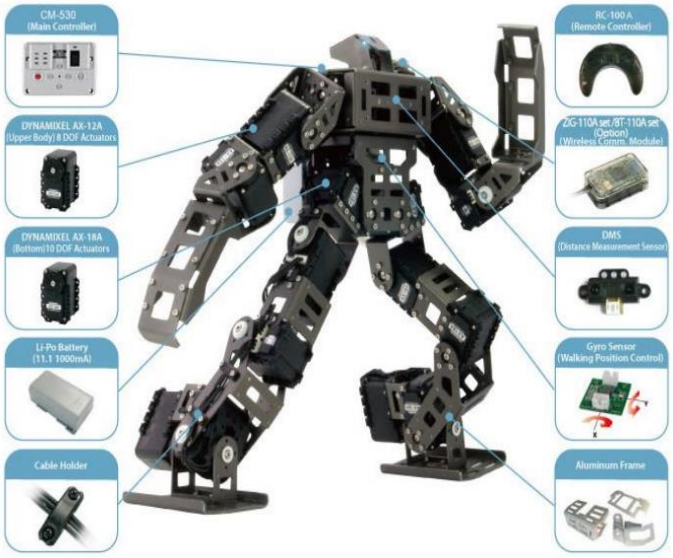


Imagen 2 Robot Bioloid GP y sus componentes

2.3. **Servomotores Dynamixel AX-12A y AX-18ª**

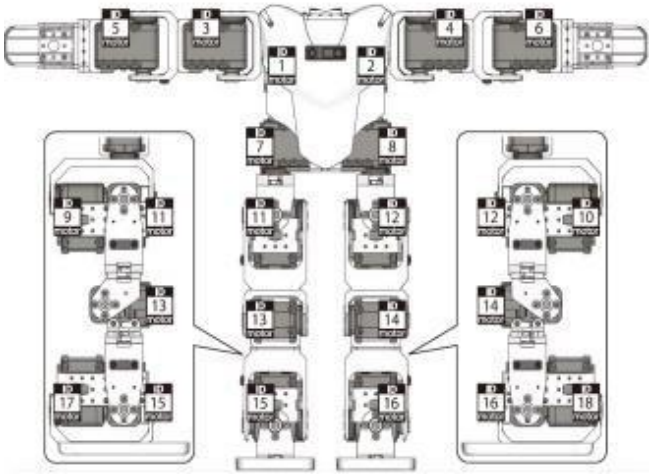


Imagen 3 Relación servomotores-articulación

2.3.1. Características

Es un nuevo actuador robótico del sistema ROBOTIS Bioloid, este sustituye al antiguo actuador robótico Dynamixel Ax12+ que, a pesar de contar con un mismo rendimiento, este cuenta con un diseño externo más avanzado.

Cada uno de los actuadores cuenta con un ID único, con lo que le permite ser identificado por el controlador, usando un rango de 0 a 252.

Ambos conectores de Dynamixel se conectan pin a pin, haciendo que el AX-12 pueda ser operado con un solo conector adjunto. (Sandorobotics, 2022)

2.3.2. Funciones

El servoactuador AX-18A de Robotis es el siguiente paso del popular servo AX-12A, que ofrece casi el doble de velocidad y un 15% más de par. El servo robot AX-18A conserva el factor de forma y todas las mismas características de su predecesor tale como la capacidad de rastrear la velocidad, temperatura, posición del eje, el voltaje y carga de este.

Además, el algoritmo de control que se utiliza para mantener la posición del eje en el actuador AX-18A puede ser ajustado individualmente, permitiendo controlar la velocidad y fuerza del motor.

La gestión del sensor y el control de posición es manejada por el microcontrolador incorporado en el servo, dejando a su controlador principal libre para realizar otras funciones.

2.4. **Raspberry**

Las herramientas Raspberry pi, es todo un sistema básicamente una computadora que permite comunicarse de forma eficiente con distintos dispositivos bajo una gran cantidad de sistemas operativos, por medio de memoria RAM, GPU (Unidad de procesamiento gráfico) un procesado, una gran cantidad de puertos pines de entrada tipo GPIO incluso en algunas versiones dispone de entradas para el almacenamiento externo como discos extraíbles o similares

En este sentido Castro (2019), menciona que por medio de este dispositivo es posible realizar una infinidad de proyectos, tanto para aplicaciones profesionales como para los aficionados, adaptándose a las diferentes necesidades y sobre todo debido a su pequeño tamaño lo hacen una gran opción para todas las aplicaciones disponibles.

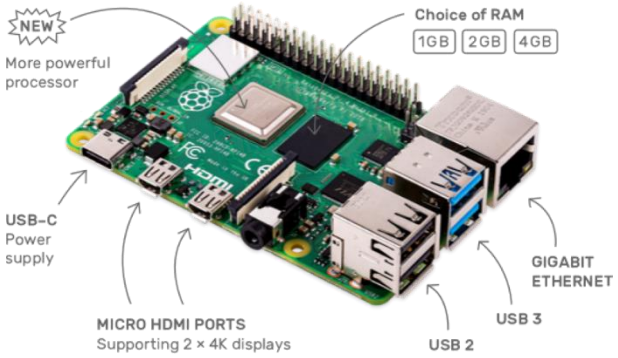


Imagen 4 Raspberry Pi 4b

2.4.1. Características

La Raspberry es muy similar a las placas base que poseen los teléfonos inteligentes. Tiene un procesador de la misma familia y muchos de los mismos periféricos (USB, audio, micrófono, conector de tarjeta SD). Las variantes más modernas incluso traen Bluetooth y WiFi. Sus interfaces la hacen interesante para controlar plataformas robóticas. (Escalante & Vargas, 2020)

Posee un sistema operativo Linux capaz de permitirles a las personas de todas las edades explorar la computación y aprender a programar lenguajes como Scratch y Python

2.4.2. Funciones

El funcionamiento de Raspberry, fue diseñado para facilitar el aprendizaje de los alumnos en las instituciones educativas, sin embargo Santos (2016), resalta que una vez fue comprobada su efectividad, sus usos se diversificaros hasta alcanzar casi cualquier uso o aplicación. Este sistema se confirma principalmente por un uso basado en Linux además de ofrecer la posibilidad de adaptar una gran cantidad de sistemas operativos diferentes, con una gama de usos prácticamente ilimitada.

Además, Lucas (2019), menciona que el éxito del Raspberry radica en su facilidad para ser programado en base a las necesidades, y sobre todo debido a que su código es

libre, haciendo que no existen las fronteras para asegurarse que su desarrollo sea constante.

Este pequeño instrumento ofrece la facilidad para la instalación y administración de una gran cantidad de herramientas, que van desde reproductores multimedia, circuitos de vigilancia, computadoras de mesa, dispositivos para la educación. Debido a su potencia de procesamiento que en la actualidad es equivalente a una computadora Pentium II de 300MHZ, y en lo que se refiere a la capacidad gráfica, es comprable a un iPhone 4s, además Adam (2019), menciona que para su correcto funcionamiento es necesario una tarjeta SD de 1GB, con una velocidad de escritura – lectura de clase 4 o superior

En general este dispositivo permite que los usuarios exploten el potencial por medio de varias herramientas que permiten el uso eficiente de todas las herramientas disponibles, que permiten mejorar el rendimiento de cualquier tipo de aplicaciones.

2.5. **RoboPlus**

Este es un software libre que permite trabajar con los diferentes modelos de robots tales como Ollo, Boiloid y Darwin Op, pertenecientes a la empresa Robotis.

Este es compatible con el controlador CM-530, servomotores Dynamixel y sensores de Robotis que son usados en el presente proyecto.

Imagen 5 RoboPlus

Roboplus contiene las siguientes herramientas

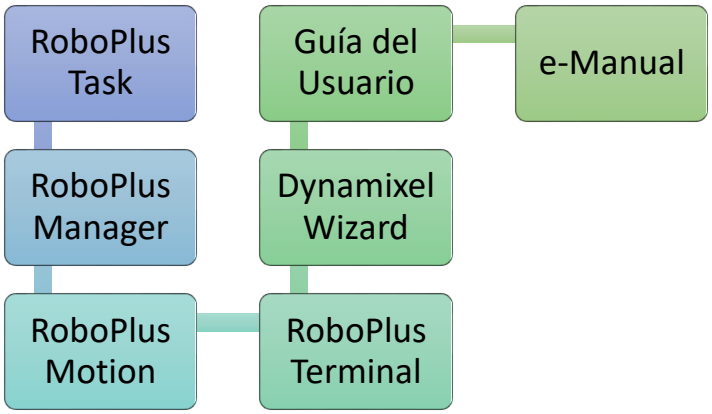


Gráfico 1 RoboPlus_Ro-Botica (2019)

2.5.1. RoboPlus Task

Task es un conjunto de movimientos para realizar ciertas acciones. RoboPlus se refiere al código fuente que especifica las tareas que debe ejecutar el robot como Task Code. El robot se mueve de acuerdo con sus códigos de tarea. El software de RoboPlus facilita la escritura de estos códigos de tareas.

Los archivos de códigos de tareas tienen una extensión de archivo *.tsk* y el icono que se muestra a la izquierda.

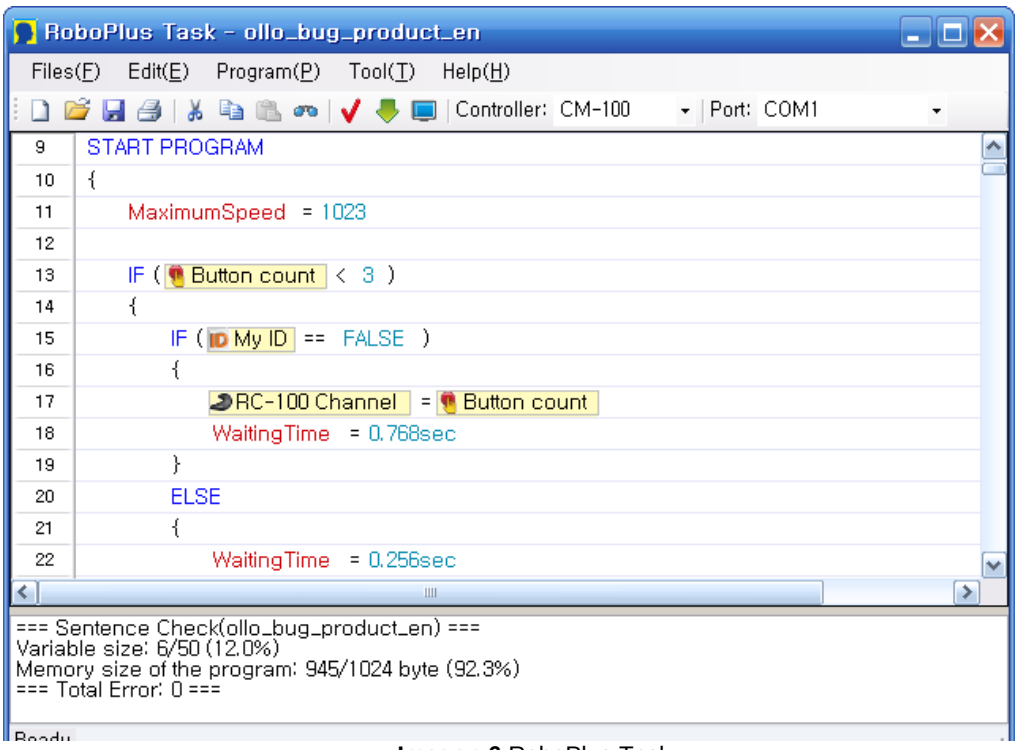


Imagen 6 RoboPlus Task

Los pasos a seguir para la instalación de RoboPlus Task son:

- Descarga el archivo .tsk
- Abra el archivo .tsk usando el software Roboplus.
- Conecte el CM 510 a la computadora con el kit provisto. (se explica más adelante en este manual)
- Elija el puerto COM correcto. (Microsoft actualiza dinámicamente sus controladores para un puerto en particular)
- Descargue el programa en el CM 510 usando el ícono de descarga en el software. (Asegúrese de que el CM 510 no esté en modo de reproducción)

2.5.2. RoboPlus Manager

RoboPlus Manager se utiliza para manejar dispositivos utilizados por un robot, entre las principales funciones que existen en el programa son administrar el firmware del controlador (actualizar y restaurar), inspeccionar el estado del controlador y los dispositivos periféricos (prueba) y configurar los modos requeridos (ajustes).

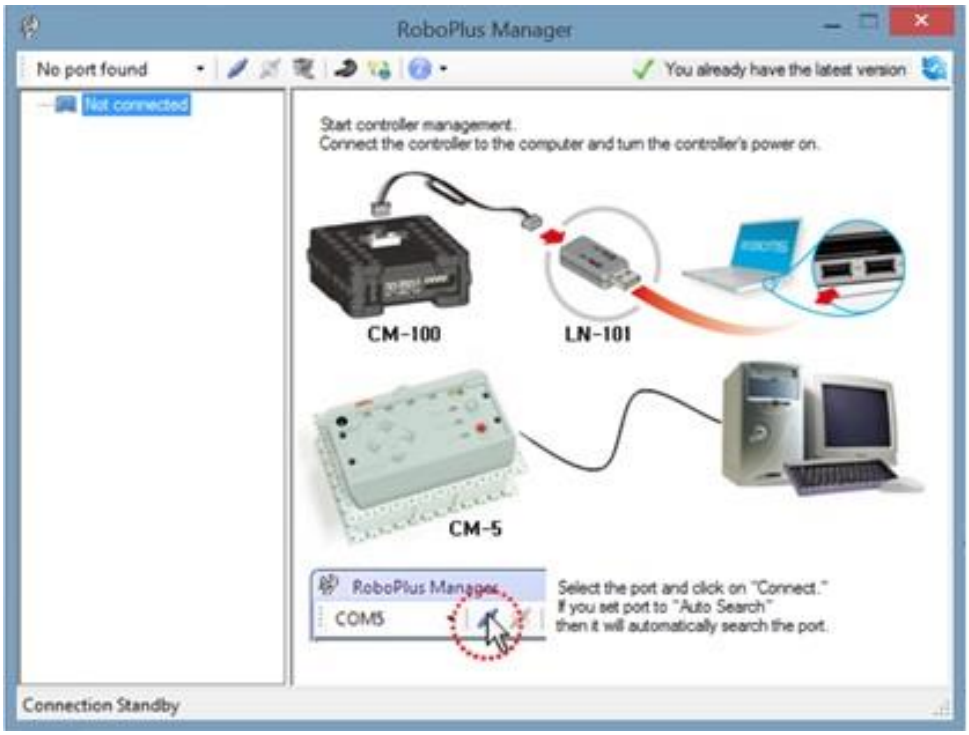


Imagen 7 RoboPlus Manager Conexiones

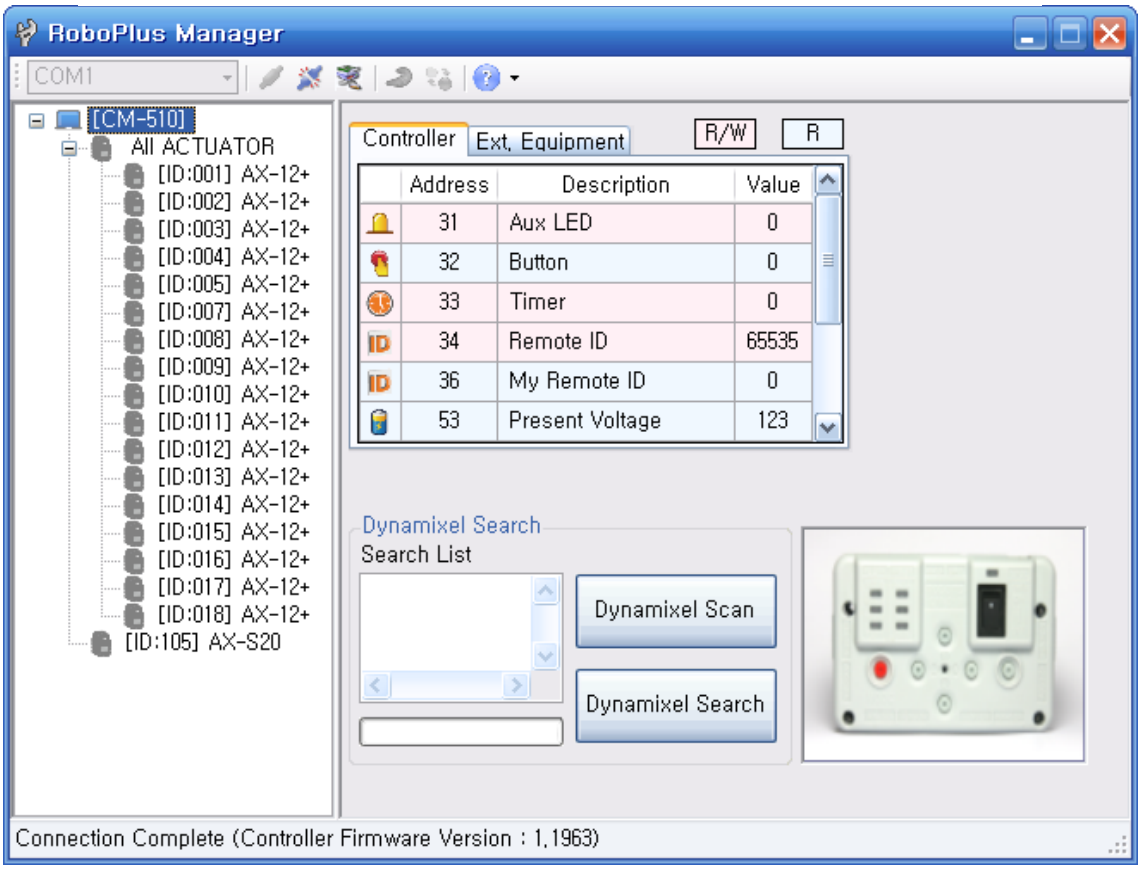


Imagen 8 RoboPlus Manager Controlador

2.5.3. RoboPlus Motion

Motion es una herramienta de programación animada que se utiliza para crear movimientos de robots (coreografías) editando la velocidad y la posición de Dynamixel. Para hacer que el robot se mueva se requiere un archivo de movimiento, cabe recalcar que el archivo de movimiento debe ser el adecuado para el robot que se vaya a ensamblar.

Un archivo de movimiento se identifica con el ícono a continuación y su extensión de archivo es .mtn.

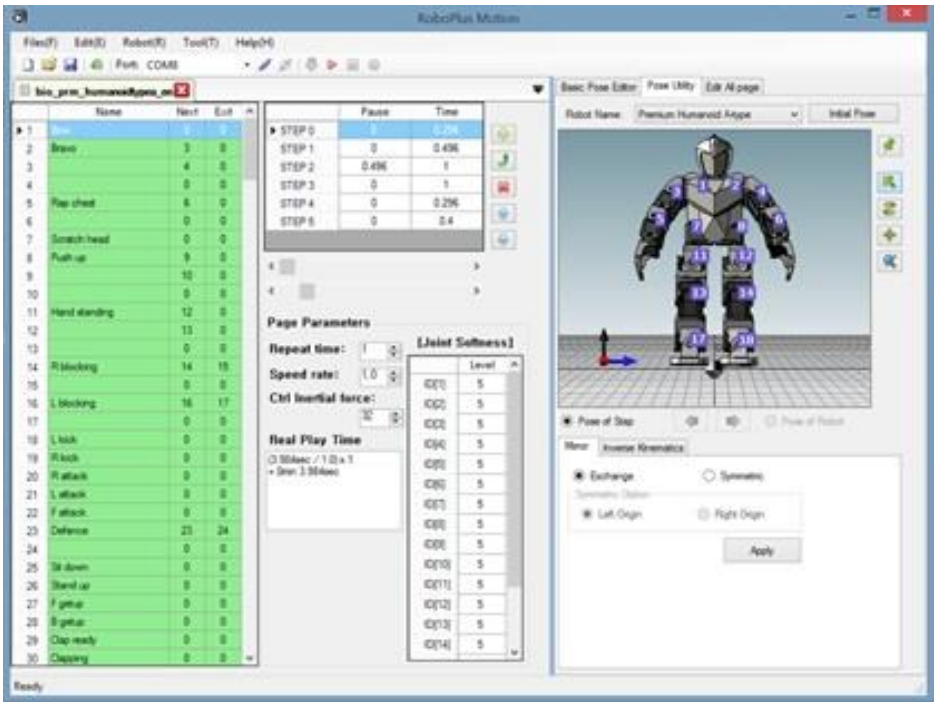


Imagen 9 RoboPlus Motion

2.5.4. RoboPlus Terminal

Terminal proporciona una manera fácil de administrar el firmware del controlador.

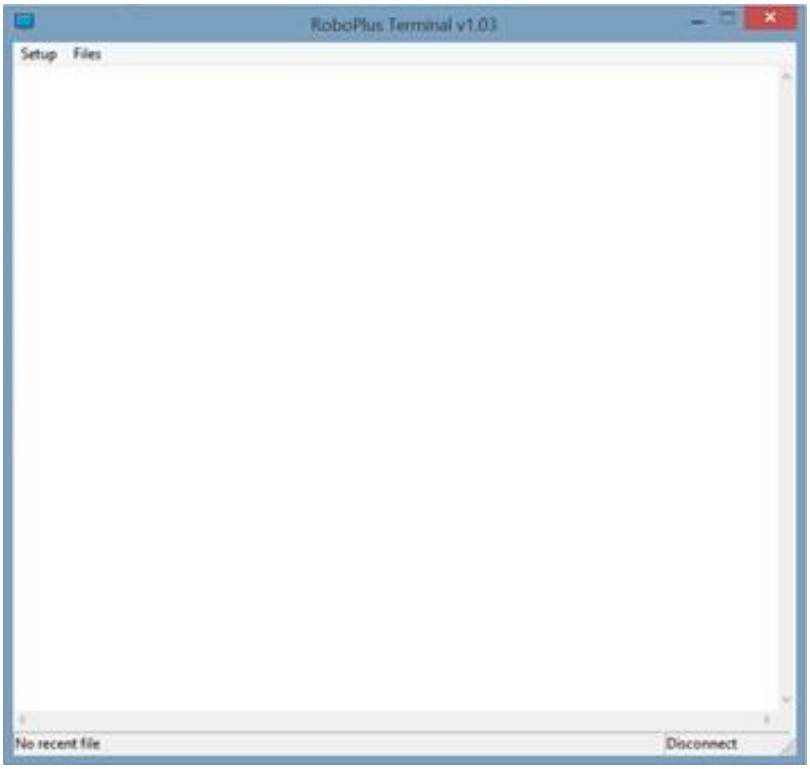


Imagen 10 RoboPlus Terminal

2.5.5. Dynamixel Wizard

Dynamixel Wizard ayuda a los usuarios a administrar Dynamixel más fácilmente. Las funciones principales del programa son las siguientes:

- Administrar el firmware de Dynamixel.
- Verifique el estado de Dynamixel.
- Configure los modos necesarios.

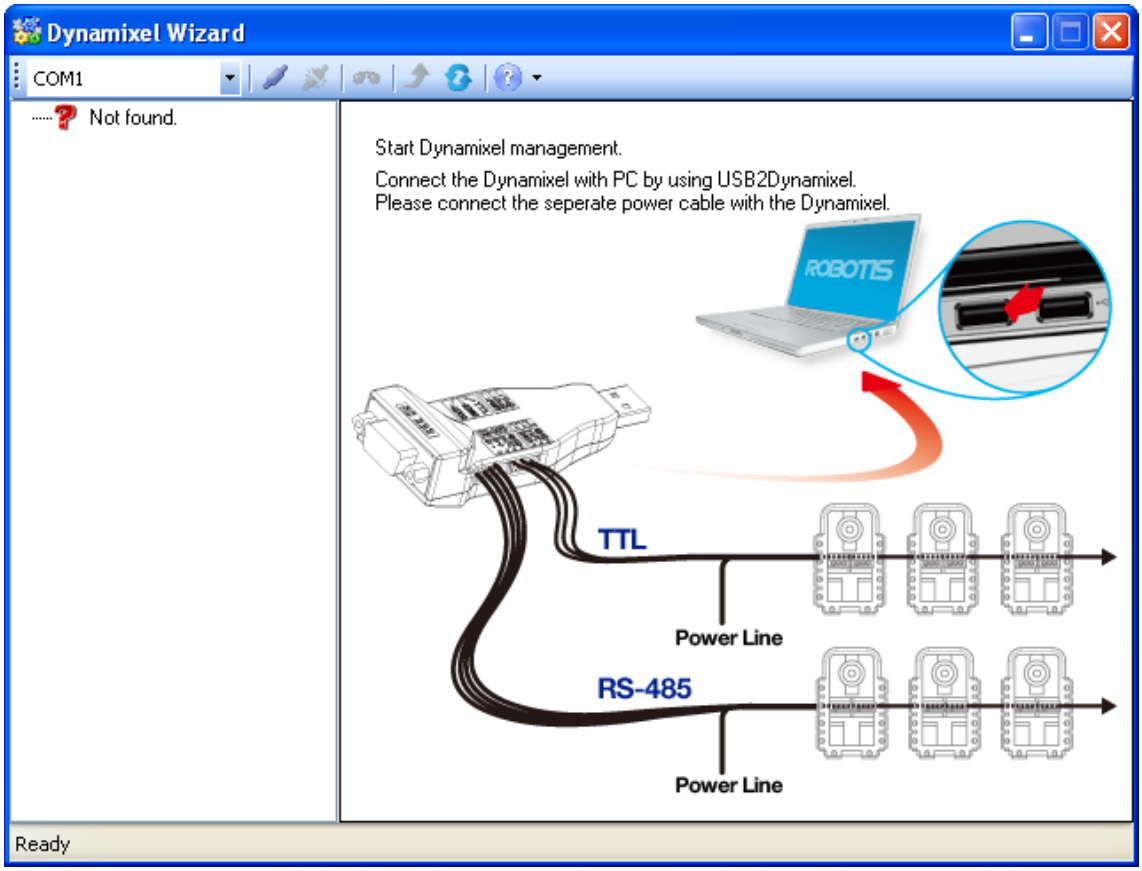


Imagen 11 Dynamixel Wizard Conexiones

2.6. **Sensor DMS-80**

Imagen 12 Sensor DMS-80

2.6.1. Características

El sensor DMS-80, también denominado sensor de distancia nos ayudará detectar objetos, paredes o imperfecciones en el terreno dentro de una distancia determinada, así se evita que el robot colisione con algún obstáculo y evitando que el prototipo tenga daños, por lo que este sensor no le afecta el color como lo sensores infrarrojos, nos ayuda a medir la distancia con precisión.

2.6.2. Funciones

Sensor de infrarrojos Sharp de gran precisión, no afectado por colores, para detectar paredes o objetos dentro de un rango de distancias.

2.7. **Sensor Gyro GS-12**

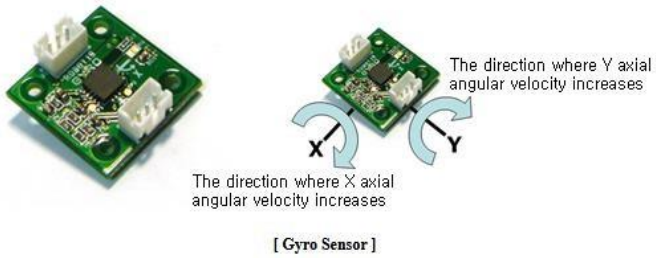


Imagen 13 Sensor Gyro GS-12

2.7.1. Características

Permitirá mantener el equilibrio del prototipo, ya que mide la velocidad angular y a su vez permite el uso de otras aplicaciones de movimiento. También, calcula que ángulo está inclinado el robot o está recibiendo más potencia. (Sensor de distancia Sharp DMS-80, 2020)

Las características físicas más importantes de este sensor son:

- Peso: 2,8 gr.
- Dimensiones: 23 mm x 23 mm x 10 mm
- Temperatura de operación: -40º ~ 85º
- Rango de velocidad angular medida: -300º/s ~ 300º/s
- Tensión de operación recomendada: 4,5 V ~ 5,5 V

2.7.2. Funciones

Entre las funciones se pueden encontrar que el sensor GS-12 Gyro es utilizado para calcular el movimiento, como la inclinación o la dirección de una fuerza aplicada sobre el robot.

Puede ser utilizado para medir la velocidad angular y permite equilibrar el robot por los movimientos de contra balance de programación, calculando hacia que lado se encuentra inclinado el robot o en que lado consume más energía.

Estas funciones permiten que las aplicaciones en las cuales es necesario este dispositivo se desarrollen de forma eficiente, y sobre todo que se garantice el rendimiento adecuado de los sistemas que lo componen.

2.8. **Receptor BT-410**

Este dispositivo cuenta con comunicación vía Bluetooth 4.0, siendo compatible con el control “Rc-100B” que está siendo usado para operar el prototipo de forma remota. Además, permite la comunicación en serie (UART) a través del Bluetooth, permitiendo el control a través de Smartphone, Tablet, PC, Laptop, etc.

Imagen 14 Receptor BT-410

2.8.1. Características

Este dispositivo permite garantizar una comunicación fluida con todos los componentes, debido a su facilidad para acoplarse a todos los dispositivos, como es el caso de Raspberry y similares.

2.8.2. Funciones

La función más importante que desempeña este dispositivo es de conectar todos los elementos para que funciones de forma fluida y eficiente. Este resulta de mucha utilidad en elementos que requieres de una buena velocidad y sobre todo de precisión para sus aplicaciones.

2.9. **Control RC-100B**

El control remoto RC – 100B nos ayudará a comunicarnos con el robot de manera remota ya que admite su comunicación con los módulos Bluetooth, ZIGbee o infrarrojo en caso de que no se adapte modulo Bluetooth o ZIGbee su comunicación predeterminada será por infrarrojo (RC-100B Remote Control, 2019).

Imagen 15 Control RC-100B

2.9.1. Características

Este tipo de control realiza las actividades de comunicar el movimiento a los distintos mecanismos, asegurando que las funciones del controlador sean comunicadas, hacia los distintos mecanismos.

3. Funcionamiento y Configuración de Programación del Robot Bioloid

3.1. Funcionamiento

3.1.1. Comunicación Raspberry Pi con controladora de Robot Bioloid CM-530

La comunicación con la CM-530 es un poco especial, ya que para enviar usa un método muy normal mientras que para recibir datos, el puerto USB está en paralelo con el receptor del mando a distancia, y usa la misma forma de comunicación que el mando a distancia.

Los parámetros de la comunicación serie son: 57600, 8 bits de datos, sin paridad y 1 bit de stop.

3.1.2. Envío de datos desde el Bioloid a la Raspberry:

El envío datos de la CM-530 a la Raspberry es muy sencillo, ya que usamos la función de envío al terminal "Print" es la que se usa para ver en pantalla variables del estado del robot, de este modo podemos enviar números, por ejemplo:

Para decirle a la Raspberry que ya puede leer un código QR enviamos un «1»

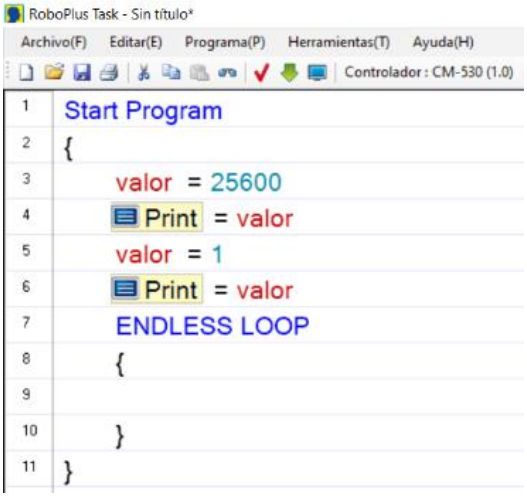


Imagen 16 Comunicación desde Robot Bioloid al Raspberry

En el Raspberry tras recibir esos cinco caracteres seguidos se pone en modo de aceptar ya lo que le llegue. Esto no lo hace la Raspberry porque si, sino que el programa que nosotros realizamos se encarga de detectar esa secuencia.

De este modo, el siguiente 1 ya sería aceptado por el software de la Raspberry como una orden lo que vemos que se ha recibido en ASCII en la Raspberry al ejecutar este programa en el Bioloid. 21 21 08 08 21 21 08 08 0D 0A 0A 20 7B 5B 43 4D 2D 35 33 30 3A 31 2E 31 39 38 32 5D 7D 0D 0A 20 7B 5B 50 43 3A 35 37 31 34 32 28 42 50 53 29 5D 5B 44 58 4C 3A 31 30 30 30 30 30 30 28 42 50 53 29 5D 7D 0D 0A 20 7B 0D 0A 0D 0A 20 7D 0D 0A 20 7B 5B 44 58 4C 3A 30 30 30 28 50 43 53 29 5D 7D 0D 0A **32 35 36 30 30** 20 20 20 20 31

La parte en negrita es lo que recibimos al enviar el 25600 y en la parte subrayada es lo que recibimos al enviar un 1.

El carácter 20 es el código ASCII del espacio en blanco, el 32 es el código ASCII del 2, el 35 del 5, el 36 del 6 y el 30 del 0 La cadena que deberíamos comprobar, escrita ya con caracteres sería: « 25600 » Y al recibir el 1 recibiremos 4 espacios en blanco y el 1. En los números se reciben como si tuviesen 5 cifras).

3.1.3. Comunicación desde la Raspberry al Bioloid

Para enviar datos al Bioloid es más extraño, si queremos enviar al robot el byte 07, el mensaje que hay que enviar desde la Raspberry es: FF 55 07 F8 00 FF Los primeros 2 bytes y los dos últimos son fijos. El tercer byte es el que queremos enviar El cuarto byte es el complemento a 1 del byte que queremos enviar. Para recibir este byte en el Bioloid debemos mirar si hay algún byte recibido por el mando a distancia y si es así, leerlo. Al leerlo nos devolverá solo el byte 07.

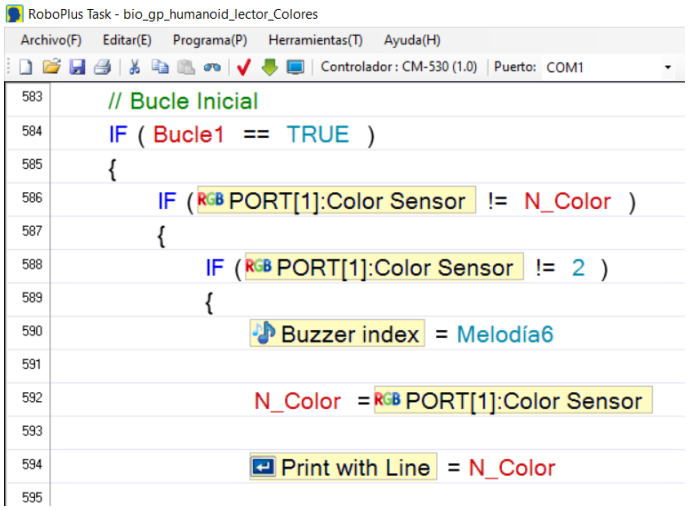


Imagen 17 Comunicación desde Raspberry al Robot Bioloid

Este programa está esperando que llegue algún dato al Bioloid, aunque parezca que recibe órdenes del mando a distancia, también recibe lo que le llega por el USB.

3.2. **Funcionamiento de Reconocimiento de Objetos de acuerdo a su color**

Tesis_Colores.py

home > Tesis > Desktop > Tesis > Pyton > Tesis_Colores.py > ...

```python
from email import message_from_binary_file
from re import T
import time
import sys
import cv2
from numpy import False_
import serial

#Funcion
def send_data():
    global TxData, TxD_packet, ser
    lowbyte = TxData & 0xff
    TxD_packet[2] = lowbyte & 0xff
    TxD_packet[3] = ~lowbyte & 0xff
    highbyte = (TxData >> 8) & 0xff
    TxD_packet[4] = highbyte & 0xff
    TxD_packet[5] = ~highbyte & 0xff
    if(ser.write (TxD_packet)!= 6):
        print("Error de transmision")
    time.sleep(0.100)
    #asegura el contenido de ~highbyte
    ser= serial.Serial("/dev/ttyUSB0",57600, timeout=0.1, write_timeout=0.1)

```

Imagen 18 Comunicación Serial desde Raspberry al Robot Bioloid

En esta imagen (18) de la Función se indica la comunicación serial de envío de información desde el Raspberry al Robot Bioloid ya sea alámbrico o por Bluetooth.

```
#
#Variables
#
#PySerial section
highbyte = 0
lowbyte = 0
TxD_packet = bytearray(6)
TxD_packet[0] = 0xff
TxD_packet[1] = 0x55
TxD_packet[2] = lowbyte
TxD_packet[3] = ~lowbyte & 0xff #asegura el contenido de ~lowbyte
TxD_packet[4] = highbyte
TxD_packet[5] = ~highbyte & 0xff #asegura el contenido de ~highbyte
ser = serial.Serial("/dev/ttyUSB0",57600, timeout=0.1, write_timeout=0.1)
ser.reset_output_buffer() #Limmpia el buffer usado para el envio de dato al CM-530

```

Tesis_Colores.py

home > Tesis > Desktop > Tesis > Pyton > Tesis_Colores.py > ...

```
#Seccion de interfaz de teclado
ESC=27
FLECHA_DERECHA= 65361
FLECHA_IZQUIERDA= 65363
FLECHA_ARRIBA= 65362
FLECHA_ABAJO= 65364
LEVANTARSE_FRENTE_No_3= 51
LEVANTARSE_ESPALDA_No_4= 52
STOP_No_2= 50
Color_No_5= 53
Color_No_6= 54

No_1 = 49
message1 = "use  el teclado para varias operaciones"
message2 = "Esc para salir del programa"
message3 = "Alternar 1 para CM-530"
message4 = "Tecla derecha o Tecla Izquierda para rotar"
```

Imagen 19 Interfaz de Teclado en valores ASCII

La imagen (19) hace referencia a la Sección de Interfaz de Teclado en donde desde el segmento del 43 al 52 se indica valores ASCII en cada de una de las variables para controlar el Robot Bioloid.

Tesis_Colores.py

home > Tesis > Desktop > Tesis > Pyton > Tesis_Colores.py > ...

```
#Seccion de OpenCV
robot_1 = False
robot_2 = False
im_CM530_ON = "/home/Tesis/Desktop/Tesis/Encendido.png"
im_CM530_OFF = "/home/Tesis/Desktop/Tesis/Apagado1.png"
Negro = "/home/Tesis/Desktop/Tesis/Negro1.png"

window_1="CM-530"
window_2="CM-530_Color"
img_1 = cv2.imread(im_CM530_OFF)
img_2 = cv2.imread(im_CM530_OFF)
img_1S = cv2.resize(img_1, (150, 150))
img_2S = cv2.resize(img_2, (150, 150))
cv2.imshow(window_1,img_1S)
cv2.moveWindow(window_1,10,50)
cv2.imshow(window_2,img_2S)
cv2.moveWindow(window_2,310,50)

```

Imagen 20 Sección de OpenCV

En la imagen (20) hace referencia a la Sección Open CV se mostrará ventanas a través del Python.

En las variables del segmento 61 y 62 son variables booleanas se indicará si esta encendido o apagado.

En las variables del segmento 63 y 64 son direcciones se indicará dónde están las imágenes que va a representar si esta encendido o apagado que se va utilizar.

En la variable del segmento 67 se mostrará la ventana que se va a utilizar en este caso es el control del robot Bioloid.

En la variable del segmento 68 se mostrará la ventana que se va a utilizar en este caso Reconocimiento de objetos de acuerdo a su color.

En las variables del segmento 71 y 72 se indicará la ventana a escala en formato de pixeles que se va a mostrar o utilizar.

En las variables del segmento 74 y 76 se indicará la ventana con la escala definida anteriormente en la posición que se desea.

Tesis_Colores.py

home > Tesis > Desktop > Tesis > Pyton > Tesis_Colores.py > ...

```
81  #Programa Inicial comienza aqui
82  #
83  print(message1)
84  print(message2)
85  print(message3)
86  #print(message4)
87
88  while True:
89      #Seccion de Teclado
90      key = cv2.waitKeyEx(33)
91      if(key == ESC):
92          print("usuario presionó Esc")
93          break
94      if(key == No_1): # alternar en tecla 1
95          if(robot_1 == False):
96              img_1 = cv2.imread(im_CM530_ON)
97              img_1S = cv2.resize(img_1, (150, 150))
98              cv2.imshow(window_1,img_1S)
99              robot_1 = True
100             print("Encendido")
101             ser.reset_output_buffer()
102         else:
103             TxData+= 0 #Valor STOP TxData
104             send_data()
105             img_1 = cv2.imread(im_CM530_OFF)
106             img_1S = cv2.resize(img_1, (150, 150))
107             cv2.imshow(window_1,img_1S)
108             robot_1 = False
109             print("Apagado")
110         cv2.waitKey(1)
```

Imagen 21 Sección de Teclado

En esta imagen (21) que se muestra en los segmentos del 83 al 85 se indicará los mensajes dependiendo la función que realizará, se muestran en los segmentos 55 al 57.

3.3. **Funcionamiento de Robot Bioloid**

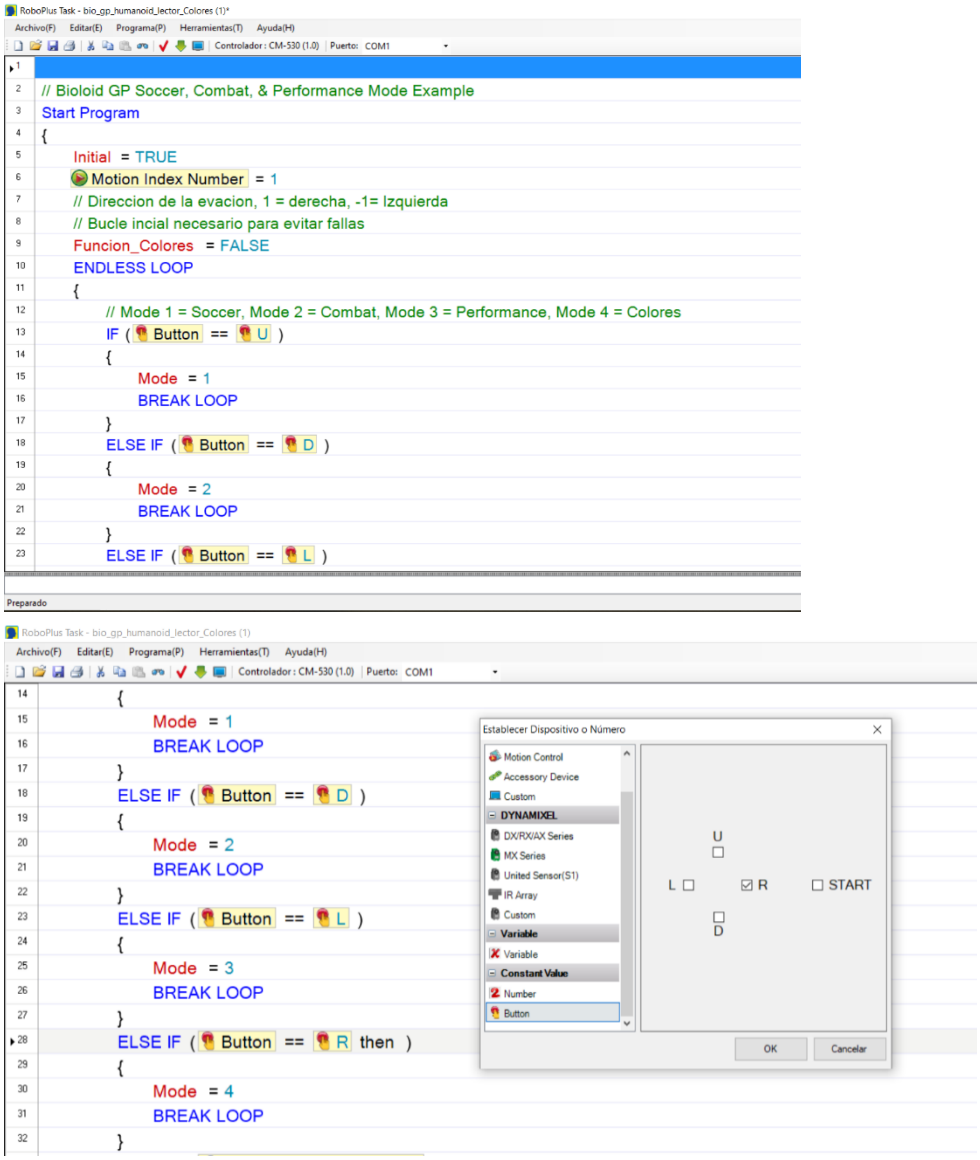


Imagen 22 Comunicación de modos por medio de botón en el CM-530.

En esta imagen (22) se indica la sección de Modos del Robot en lo cual se presiona en el Botón R en el CM530 para dar el inicio a la comunicación de Colores en la variable Modo 4.

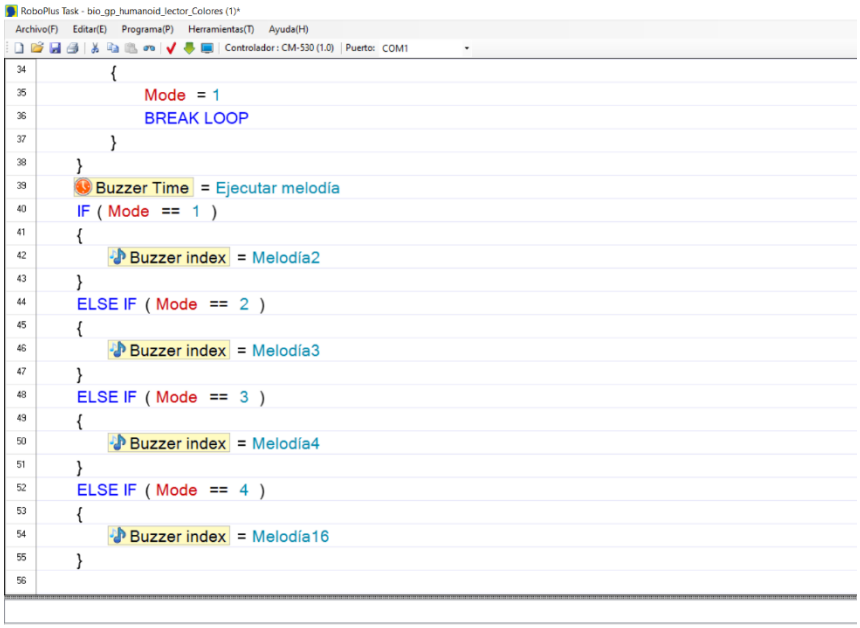


Imagen 24 Melodías de Modos

En esta imagen (24) se indica las melodías de cada uno de los modos que desea seleccionar, en este caso se escucharía del modo 4.

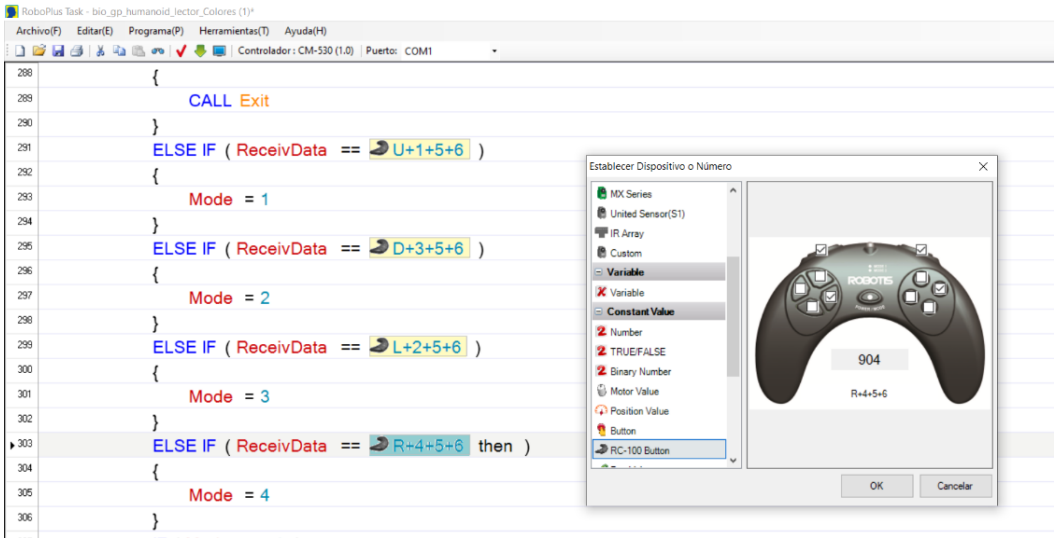


Imagen 23 Comunicación de Modos por medio del Control RC-100B

En esta imagen (23) se indica otra comunicación de modos por medio del Control RC-100B que desea seleccionar, en este caso sería el modo 4.

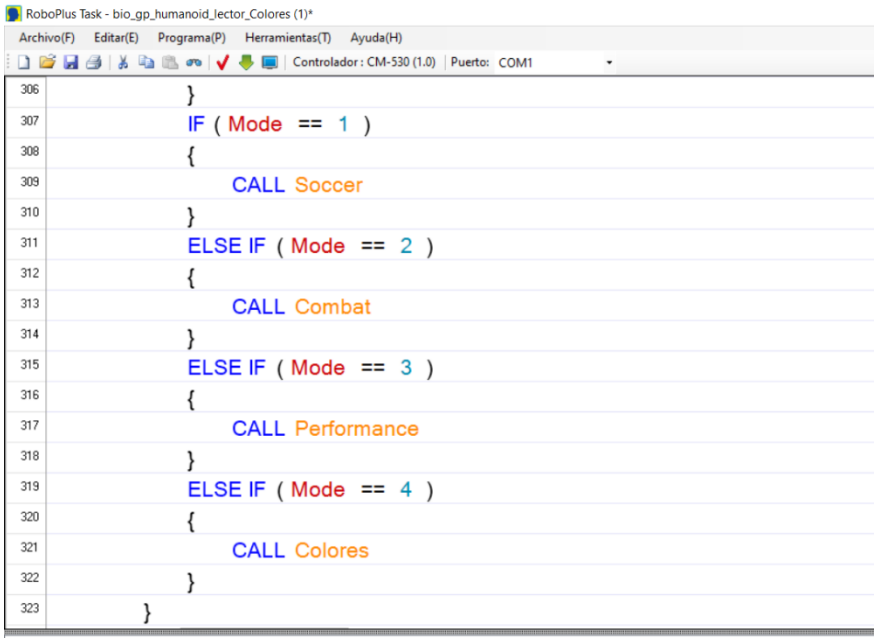


Imagen 26 Sección de Modos

En esta imagen (26) se indica la sección de Modos que se desea llamar.

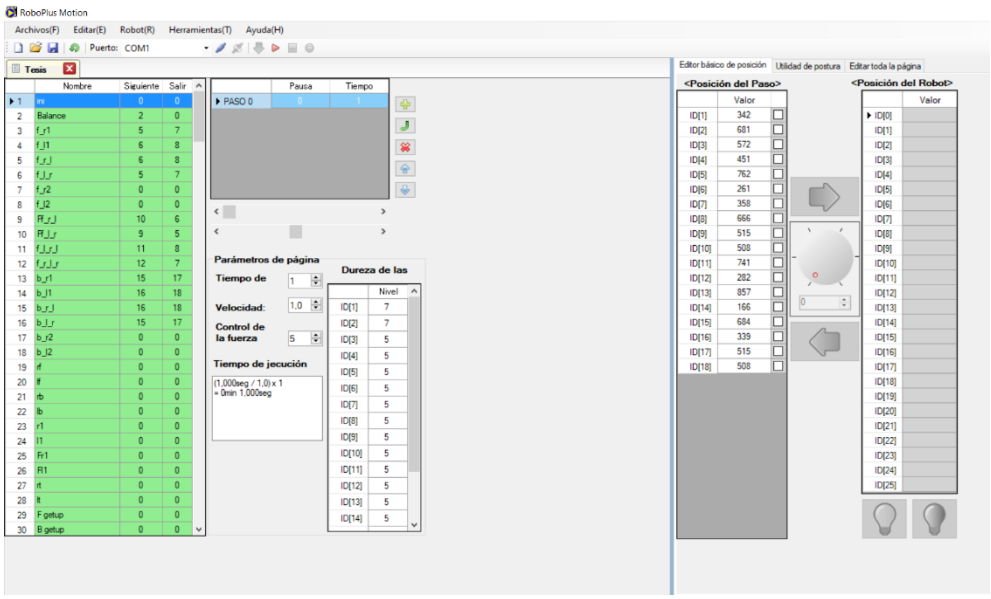


Imagen 25 Sección de Movimientos en RoboPlus Motion

En esta imagen (25) se indica la programación de los movimientos en tiempo, velocidad y el control de la fuerza para el control del robot Bioloid.

3.4. **Funcionamiento de Reconocimiento de Objetos de acuerdo a su forma**

Tesis-CAM.py

```
import cv2
import matplotlib.pyplot as plt
import cvlib as cv
import urllib.request
import numpy as np
from cvlib.object_detection import draw_bbox
import concurrent.futures

url='http://192.168.1.110/cam-hi.jpg'
im=None
```

Imagen 27 Detección de Objetos

En esta imagen (27) se indica la programación de detección de objetos mediante una dirección IP de la cámara.

Tesis-CAM.py

```
def run1():
    #cv2.namedWindow("live transmission", cv2.WINDOW_AUTOSIZE)
    cv2.namedWindow("live transmission", cv2.WINDOW_NORMAL)
    while True:
        img_resp=urllib.request.urlopen(url)
        imgnp=np.array(bytearray(img_resp.read()),dtype=np.uint8)
        im = cv2.imdecode(imgnp,-1)
        #live_transmission = cv2.resize('live transmission', (150, 150))
        cv2.moveWindow('live transmission',0,200)
        cv2.resizeWindow('live transmission', 600, 500)
        cv2.imshow('live transmission',im)
        key=cv2.waitKey(5)
        if key==ord('q'):
            break

    cv2.destroyAllWindows()
```

Imagen 28 Transmisión en vivo

En esta imagen (28) se mostrará en una ventana la transmision en vivo por medio de la cámara ESP32-CAM.

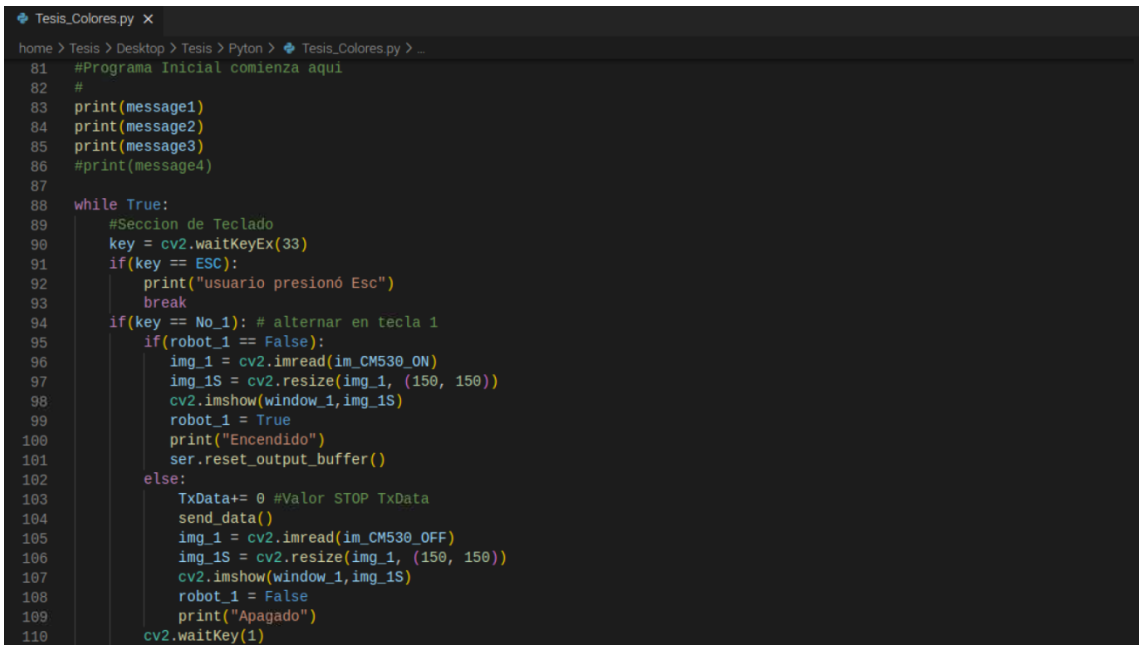
Tesis-CAM.py

```
def run2():
    #cv2.namedWindow("detection", cv2.WINDOW_AUTOSIZE)
    cv2.namedWindow("detection", cv2.WINDOW_NORMAL)
    while True:
        img_resp=urllib.request.urlopen(url)
        imgnp=np.array(bytearray(img_resp.read()),dtype=np.uint8)

        im = cv2.imdecode(imgnp,-1)

        bbox, label, conf = cv.detect_common_objects(im)
        im = draw_bbox(im, bbox, label, conf)
        cv2.moveWindow('detection',600,200)
        cv2.resizeWindow('detection', 600, 500)

        cv2.imshow('detection',im)
        key=cv2.waitKey(5)
        if key==ord('q'):
            break

    cv2.destroyAllWindows()
```

Imagen 29 Detección

En esta imagen (29) se mostrará en una ventana la detección y recononocimiento de objetos por medio de la cámara ESP32-CAM.

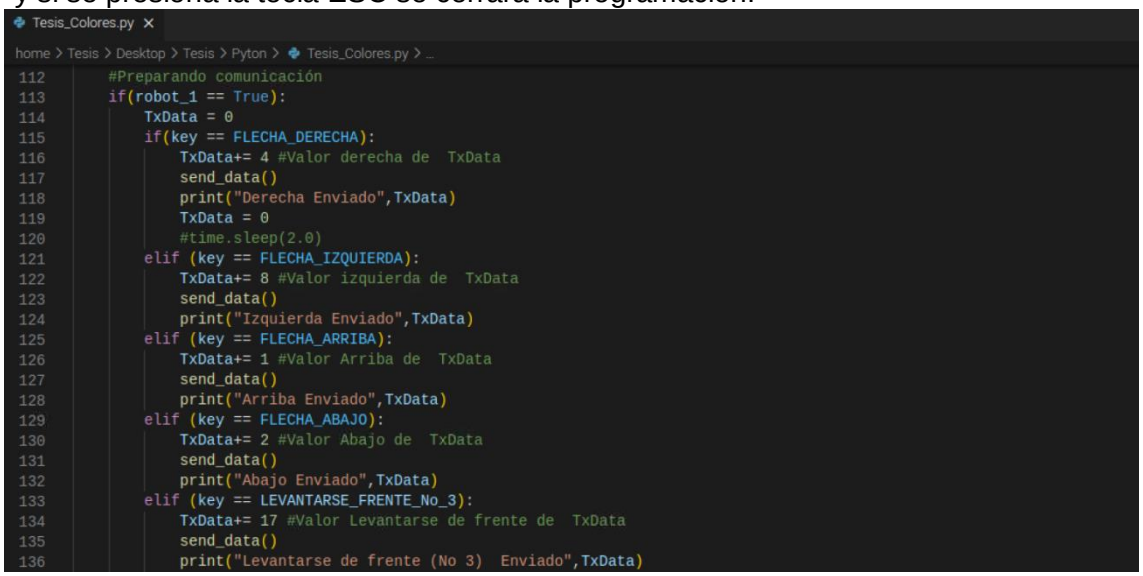
3.5. **Configuración de Reconocimiento de objetos de acuerdo a su color**



```
Tesis_Colores.py
home > Tesis > Desktop > Tesis > Pyton > Tesis_Colores.py > ...
#Programa Inicial comienza aqui
#
print(message1)
print(message2)
print(message3)
#print(message4)

while True:
    #Seccion de Teclado
    key = cv2.waitKeyEx(33)
    if(key == ESC):
        print("usuario presionó Esc")
        break
    if(key == No_1): # alternar en tecla 1
        if(robot_1 == False):
           img_1 = cv2.imread(im_CM530_ON)
           img_1S = cv2.resize(img_1, (150, 150))
           cv2.imshow(window_1,img_1S)
           robot_1 = True
           print("Encendido")
           ser.reset_output_buffer()
        else:
            TxData+= 0 #Valor STOP TxData
            send_data()
            img_1 = cv2.imread(im_CM530_OFF)
            img_1S = cv2.resize(img_1, (150, 150))
            cv2.imshow(window_1,img_1S)
            robot_1 = False
            print("Apagado")
        cv2.waitKey(1)
```

Imagen 30 Activación y Desactivación del Robot Bioloid

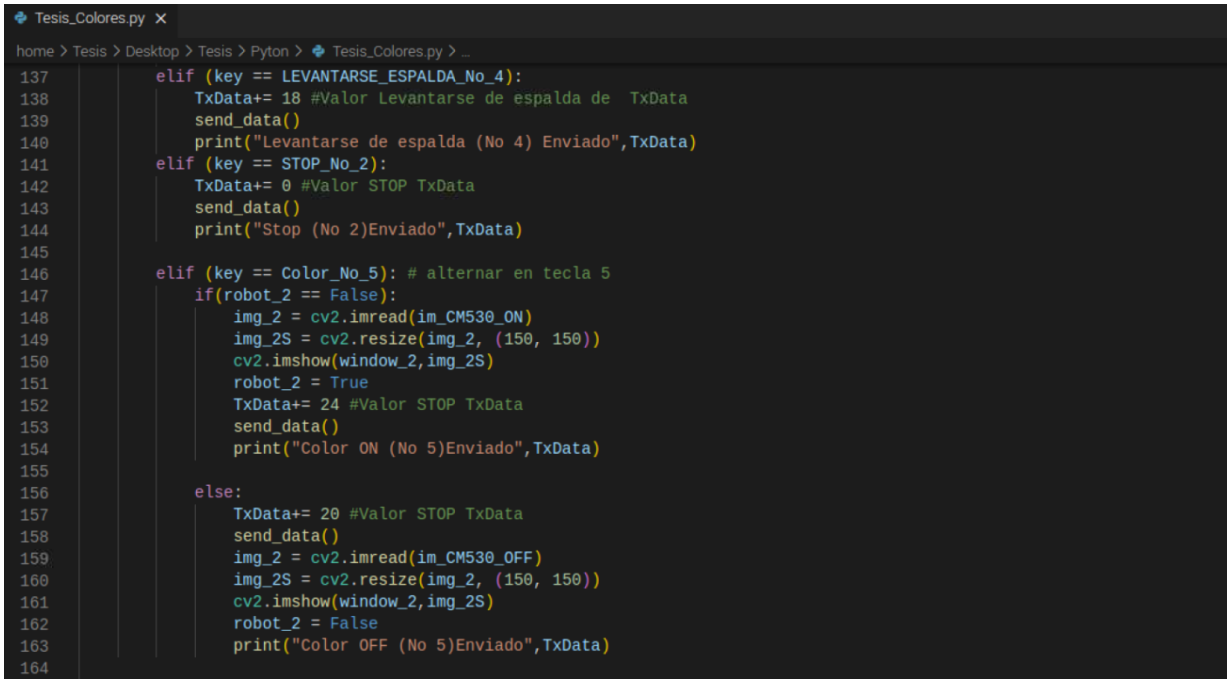
En esta imagen (30) se indica la activación y desactivación del programa mostrada la imagen escalada, posicionada y comunica si está encendido o apagado el Robot Bioloid y si se presiona la tecla ESC se cerrará la programación.



```
Tesis_Colores.py
home > Tesis > Desktop > Tesis > Pyton > Tesis_Colores.py > ...
    #Preparando comunicación
    if(robot_1 == True):
        TxData = 0
        if(key == FLECHA_DERECHA):
            TxData+= 4 #Valor derecha de  TxData
            send_data()
            print("Derecha Enviado",TxData)
            TxData = 0
            #time.sleep(2.0)
        elif (key == FLECHA_IZQUIERDA):
            TxData+= 8 #Valor izquierda de  TxData
            send_data()
            print("Izquierda Enviado",TxData)
        elif (key == FLECHA_ARRIBA):
            TxData+= 1 #Valor Arriba de  TxData
            send_data()
            print("Arriba Enviado",TxData)
        elif (key == FLECHA_ABAJO):
            TxData+= 2 #Valor Abajo de  TxData
            send_data()
            print("Abajo Enviado",TxData)
        elif (key == LEVANTARSE_FRENTE_No_3):
            TxData+= 17 #Valor Levantarse de frente de  TxData
            send_data()
            print("Levantarse de frente (No 3)  Enviado",TxData)
```

Imagen 31 Comunicación de Movimientos del Robot Bioloid

En esta imagen (31) nos indica la comunicación cuando el robot esta encendido si se presionan las teclas que se han asignado anteriormente y mostrará el mensaje de la función que se realice.



```
Tesis_Colores.py
home > Tesis > Desktop > Tesis > Pyton > Tesis_Colores.py > ...
137         elif (key == LEVANTARSE_ESPALDA_No_4):
138             TxData+= 18 #Valor Levantarse de espalda de  TxData
139             send_data()
140             print("Levantarse de espalda (No 4) Enviado",TxData)
141         elif (key == STOP_No_2):
142             TxData+= 0 #Valor STOP TxData
143             send_data()
144             print("Stop (No 2)Enviado",TxData)
145
146         elif (key == Color_No_5): # alternar en tecla 5
147             if(robot_2 == False):
148                 img_2 = cv2.imread(im_CM530_ON)
149                 img_2S = cv2.resize(img_2, (150, 150))
150                 cv2.imshow(window_2,img_2S)
151                 robot_2 = True
152                 TxData+= 24 #Valor STOP TxData
153                 send_data()
154                 print("Color ON (No 5)Enviado",TxData)
155
156             else:
157                 TxData+= 20 #Valor STOP TxData
158                 send_data()
159                 img_2 = cv2.imread(im_CM530_OFF)
160                 img_2S = cv2.resize(img_2, (150, 150))
161                 cv2.imshow(window_2,img_2S)
162                 robot_2 = False
163                 print("Color OFF (No 5)Enviado",TxData)
164
```

Imagen 32 Activación y Desactivación de Sección Colores por medio del teclado

En esta imagen (32) se indica la activación y desactivación del programa mostrada la imagen escalada, posicionada y comunica si está encendido o apagado la Sección de Colores.

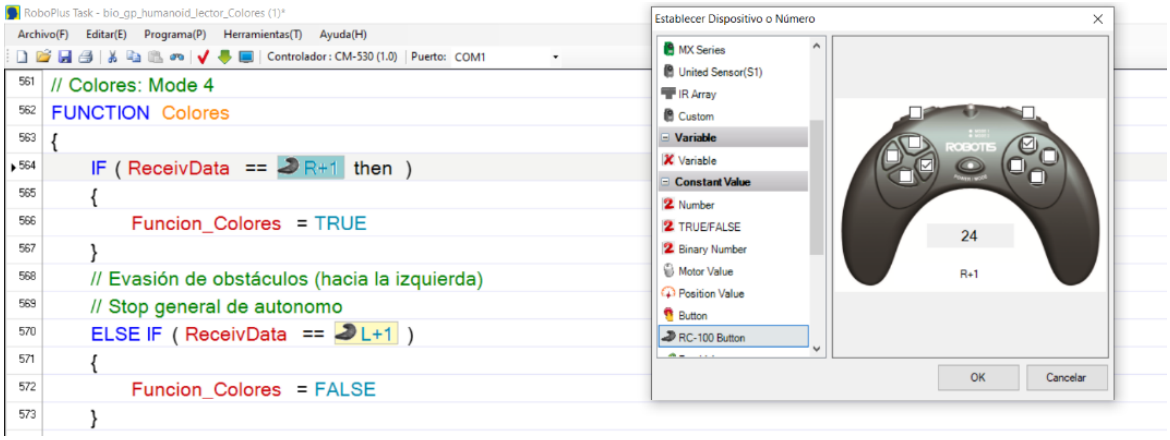


Imagen 33 Activación de Sección Colores por medio del Control RC-100B

En esta imagen (33) se indica la activación del programa mostrada la imagen escalada, posicionada y comunica si está encendido la Sección de Colores por medio del Control RC-100B.

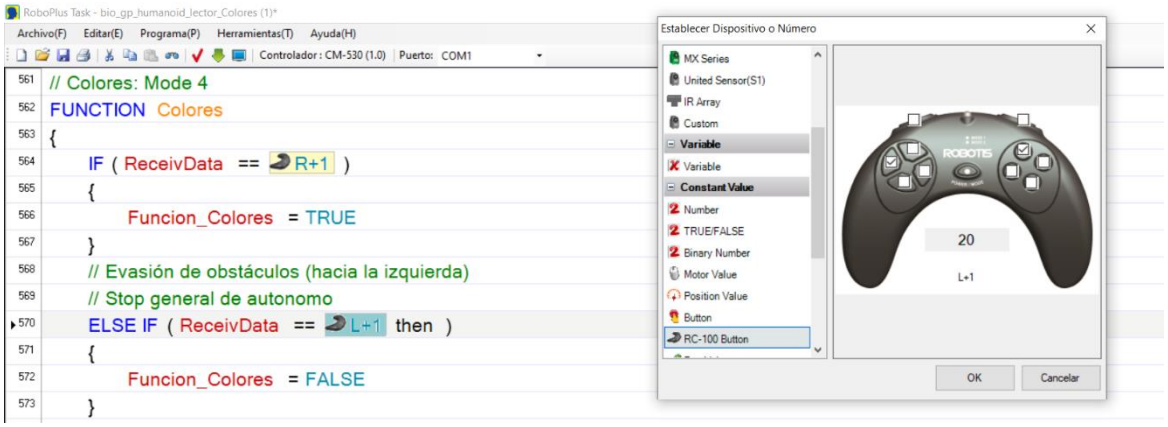


Imagen 36 Desactivación de Sección Colores por medio del Control RC-100B

En esta imagen (36) se indica la desactivación del programa mostrada la imagen escalada, posicionada y comunica si está apagada la Sección de Colores por medio del Control RC-100B.

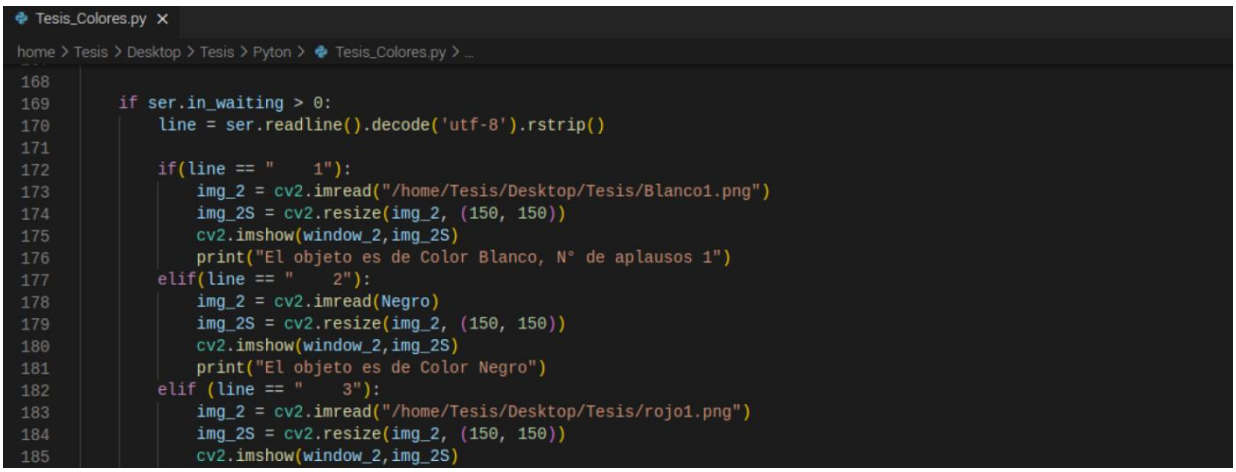


Imagen 35 Comunicación Serial

En esta imagen (35) se indica la espera de la comunicación serial y la variable guarda el mensaje que se recibe en un formato de códigos ASCII y lo codifica con un lenguaje que se puede leerlo.

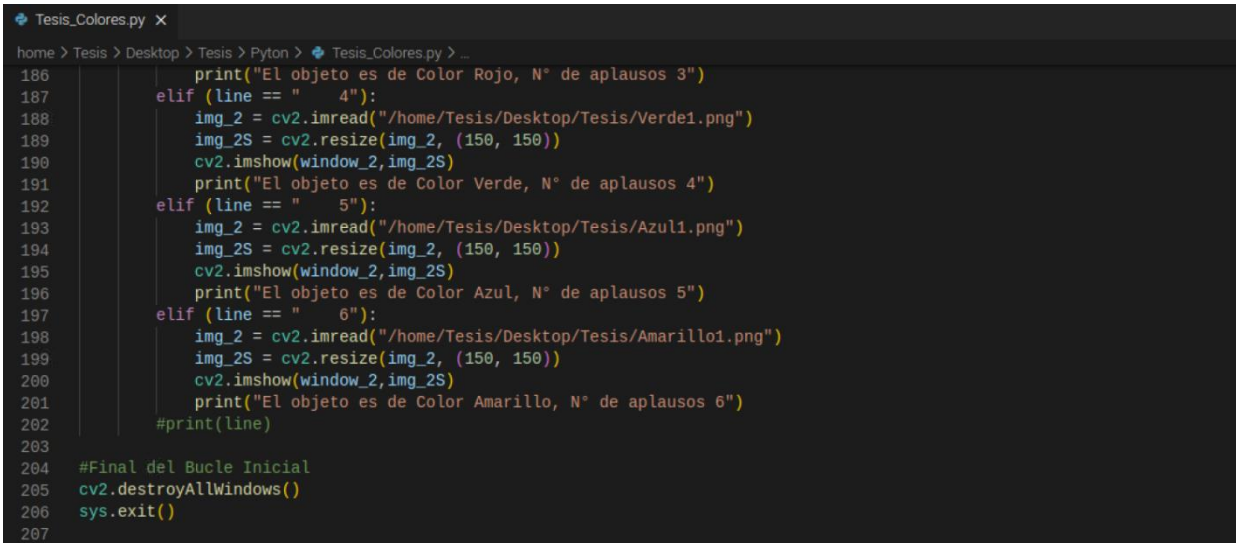


Imagen 37 Función de Colores mediante por el sensor RCS-10

En esta imagen (37) proyectará el color en la ventana escalada y posicionada que está recibiendo mediante por el sensor RCS-10 y realizará una función por cada color que se reciba.

3.6. **Configuración de Reconocimiento de objetos de acuerdo a su forma**

Tesis-CAM.py

```python
import cv2
import matplotlib.pyplot as plt
import cvlib as cv
import urllib.request
import numpy as np
from cvlib.object_detection import draw_bbox
import concurrent.futures

url='http://192.168.1.110/cam-hi.jpg'
im=None
```

Imagen 39 Detección de Objetos mediante una dirección IP

Imagen 38 Programación Arduino de la ESP32-CAM

En esta imagen (39) se indica la configuración de una dirección IP mediante por un router que compartirá la señal de WiFi para la conexión a la cámara ESP32-CAM que receptará las imágenes captadas y muestra en una ventana.

4. Resultados

4.1. Reconocimiento de objetos por su color

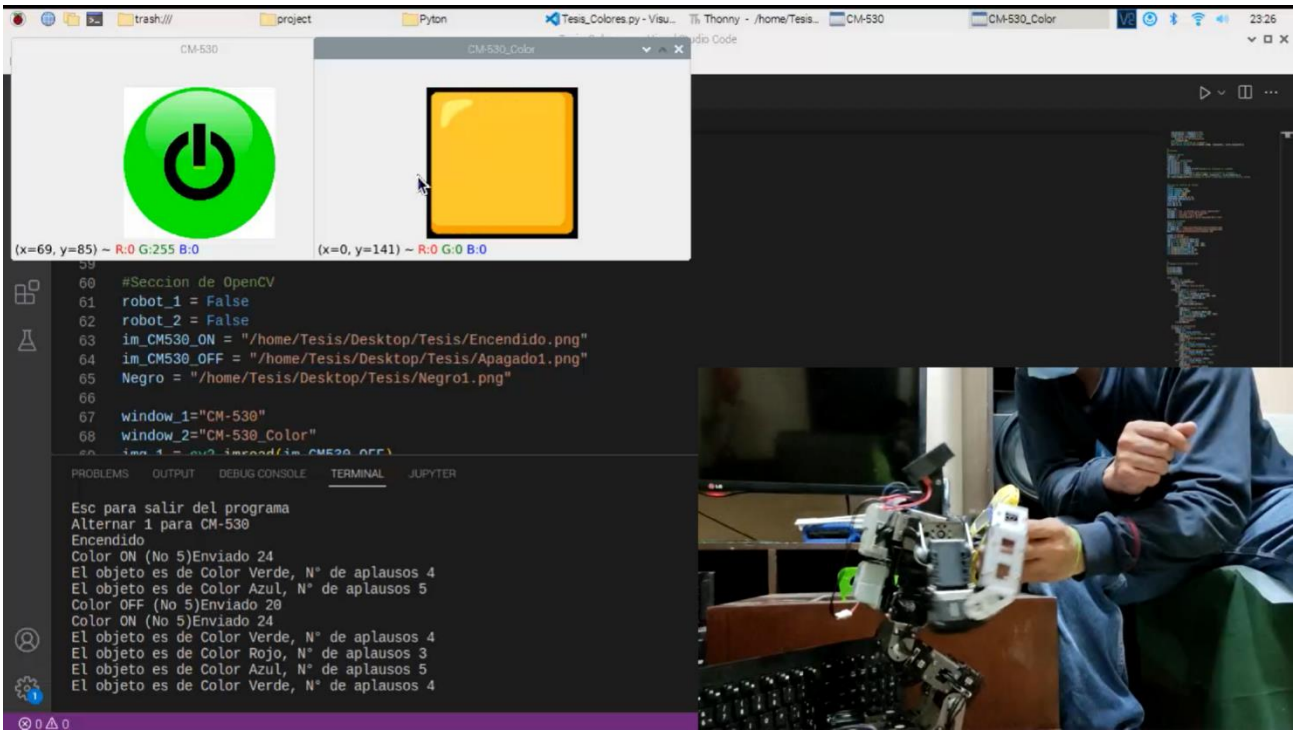


Imagen 40 Reconocimiento de Objetos por su color

En esta imagen (40) se muestra que el Robot Bioloid mediante por el sensor RCS-10 reconoce el objeto por su color, lo muestra en la ventana escalada y comunica el color detectado y realiza la función designada.

4.2. **Reconocimiento de Objetos por su forma**

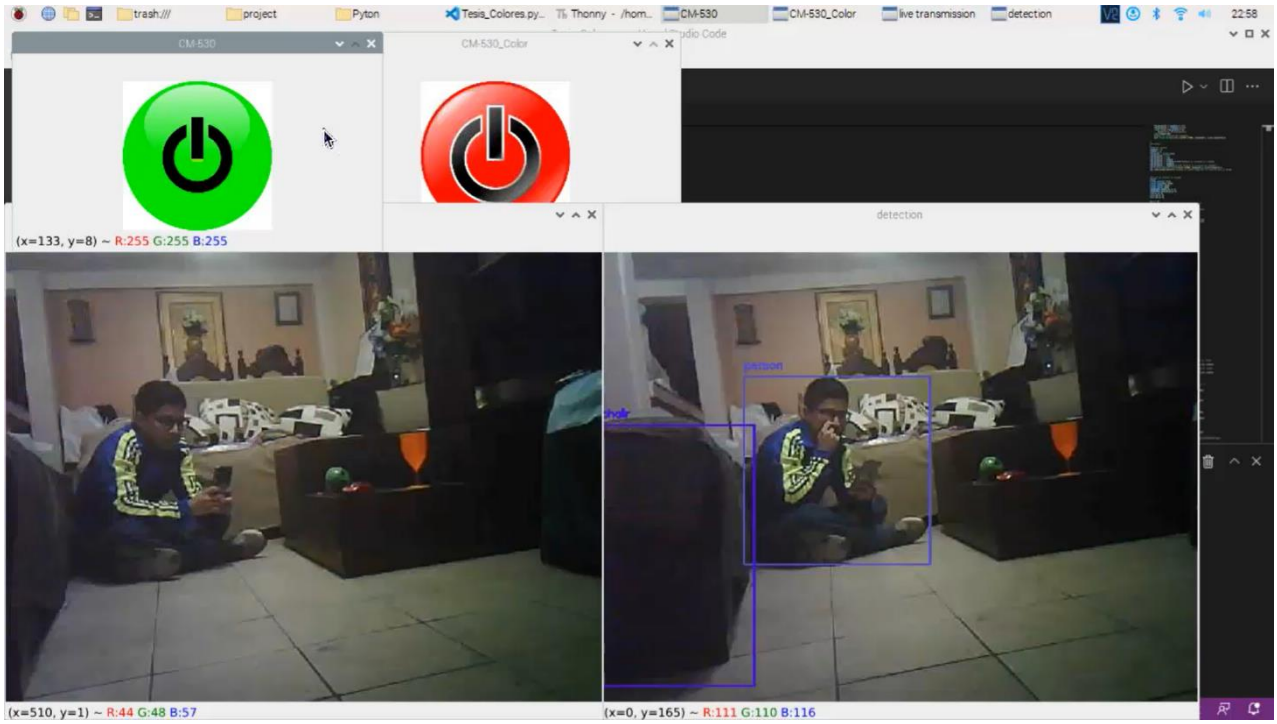


Imagen 41 Reconocimiento de Objetos por su forma

En esta imagen (41) se muestra que el Robot Bioloid mediante por la cámara ESP32-CAM reconoce el objeto por su forma, mostrará en la ventana escalada, el objeto detectado.

4.3. **Comunicación por medio del módulo Bluetooth**

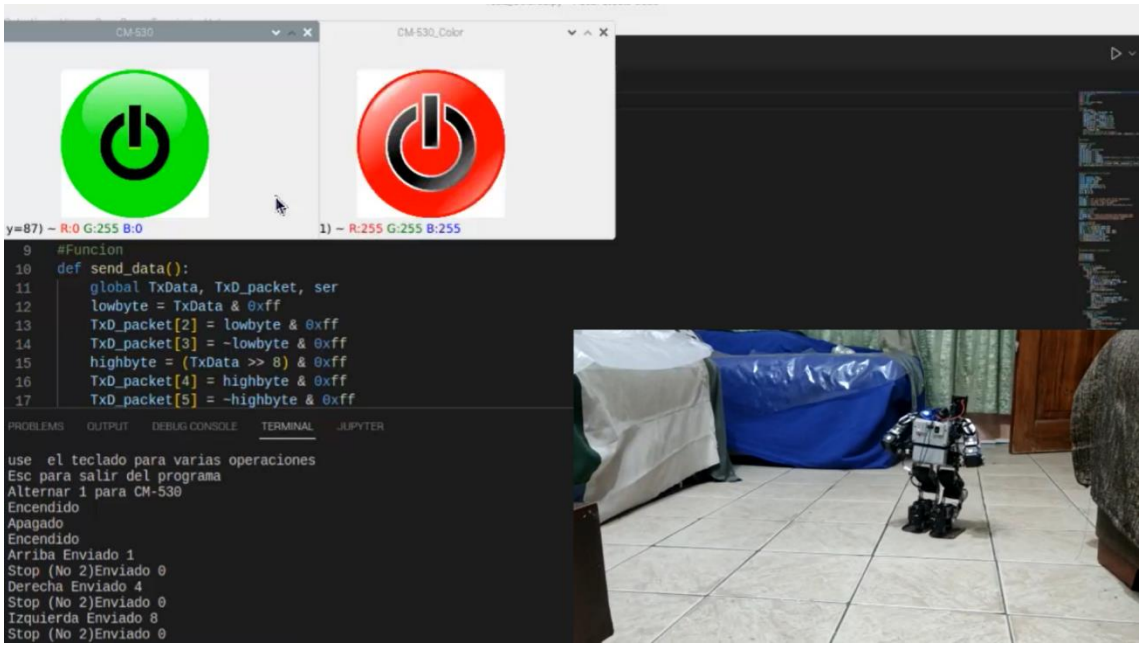


Imagen 42 Comunicación por medio del módulo Bluetooth

En esta imagen (42) se muestra que el Robot Bioloid mediante por el módulo Bluetooth recepta los mandos, mostrará en los mensajes las indicaciones dadas.

5. Anexos

5.1. **Prácticas Comunes**

Estas prácticas son desarrolladas por nuestros compañeros

1

UNIDAD DE APRENDIZAJE
DE ROBÓTICA

MANUAL DE PRÁCTICAS DE ROBOT HUMANOIDE BIOLOID GP

PROGRAMA DE INGENIERÍA ELECTRONICA

UNIVERSIDAD POLITECNICA SALESIANA
SEDE GUAYAQUIL

AUTORES:

CHRISTOPHER GERMAN
NAVARRETE VALENCIA

SAUL DAVID
VALLEJO SANTANA

ECUADOR - GUAYAQUIL

Tabla de contenido

PRESENTACIÓN

Las presentes prácticas fueron desarrolladas en apego al programa de la Unidad de Aprendizaje de Electiva, con la finalidad de que los alumnos puedan poner en práctica los conocimientos teóricos que se van adquiriendo a lo largo del curso. Esta Unidad de Aprendizaje consta de la siguiente estructura:

1.- Comprender los conceptos básicos de robótica móvil identificando los conceptos importantes.
2.- Identificar los tipos de sensores utilizados en el ROBOT
HUMONAOIDE BIOLOID GP
3.- Conocer los programas a utilizar y sus funciones

Cada Práctica consta de Titulo, Objetivos, Introducción, Material y Equipo a Utilizar, Duración, Desarrollo y Evaluación. El alumno debe replicar de forma práctica lo que se encuentra en el apartado de Desarrollo y resolver lo que se pide en el apartado de Evaluación.

INTRODUCCIÓN A LA ROBOTICA:

Actualmente la robótica ha cobrado gran importancia en la vida cotidiana y en procesos industriales. Se ha enfocado en las líneas de producción, donde el monitoreo de la producción desempeña un papel sumamente importante. Por lo que, desarrollar un sistema automatizado genera grandes beneficios.

El termino robot, viene de la palabra checa robota, que significa "labor forzada", servicio o esclavo.

La robótica educativa es el conjunto de actividades pedagógicas que apoyan y fortalecen áreas específicas del conocimiento y desarrollan habilidades y competencias en el alumno, a través de los siguientes procesos: concepción, creación, ensamble y puesta en funcionamiento de robots.

Mediante la manipulación del Kit de Robótica, se lleva a cabo el ensamble de hasta 3 diferentes tipos de robots móviles (todo terreno, montacargas y pinzas), los cuales permiten ilustrar los principios y fundamentos básicos de la robótica móvil. La robótica móvil es una ciencia o rama de la tecnología que estudia el diseño y creación de máquinas que pueden desplazarse de un lugar a otro, cuya finalidad es desempeñar tareas repetitivas o peligrosas por el ser humano o que requieren el uso del pensamiento e inteligencia.

INTRODUCCIÓN A ROBOT HUMANOIDE BIOLOID GP

Los robots se utilizan en la educación para enseñar y desarrollar en el estudiante habilidades y destrezas que les sirvan para resolver problemas de la vida real.

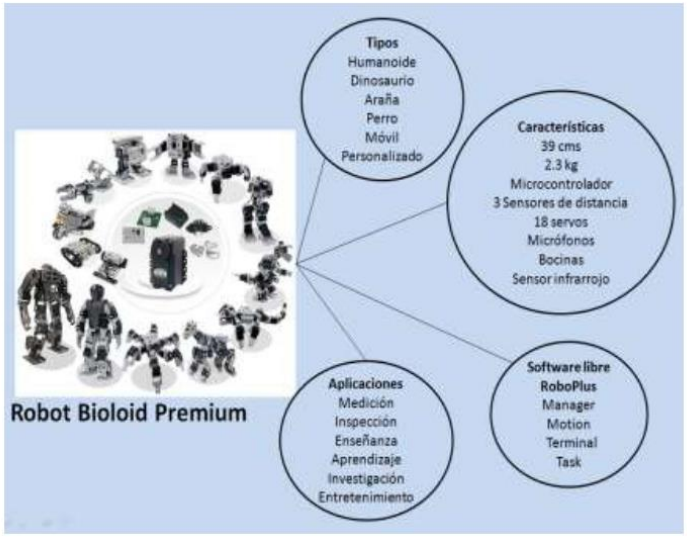
El proceso de programación de un robot consiste en introducir en su sistema de control las instrucciones necesarias para que desempeñe las tareas para las que ha sido diseñado. Para programar un robot se sigue un proceso semejante al de la elaboración de un programa computacional destinado a cualquier otra aplicación (celular, tablet, sito web, etc.). Primero será necesario establecer el algoritmo idóneo que permita al robot llevar a cabo las tareas para las que ha sido diseñado, tras lo cual se traducirá dicho algoritmo en un lenguaje de programación por el sistema de control del robot. Dicho lenguaje debe permitir especificar de forma clara y sencilla las tareas que debe realizar el robot.

A continuación, se presentan diferentes tipos de procedimientos de programación de robots:

a) Programación guiada o directa: El operario interviene guiando manualmente el brazo del robot, y hace que este vaya describiendo los movimientos y trace las trayectorias necesarias para cumplir su función. Cada uno de los movimientos realizados se va almacenando en la memoria del robot, de forma que podrán ser repetidos posteriormente, ya sin intervención humana. En este tipo de programación es necesario disponer del propio robot para la elaboración del programa.

b) Programación textual o indirecta: En este caso no es necesaria la presencia del robot para realizar el programa, puesto que este se lleva a cabo en un lenguaje de programación. El programa consiste en un conjunto de instrucciones; cuando el programa sea grabado en la memoria del robot, este realizará las acciones indicadas en el mismo. Este tipo de programación permite realizar operaciones más complejas y con mayor grado de precisión. Además, presenta la ventaja de que es posible establecer relaciones entre el robot y su entorno. Para ello basta con introducir en el programa los datos procedentes de los sensores de forma que el robot actúe en consonancia con los mismos, tal y como ocurre en los robots inteligentes. Este tipo de programación puede dividirse en 2 tipos: explícita (lenguajes de programación estructurados) y especificativa (lenguajes de programación orientados a objetos).

En el siguiente mapa conceptual se presenta un mapa conceptual sobre el Kit de Robótica Bioloid GP de la marca Robotis; el cual hace referencia a los diferentes tipos de configuraciones, características y en qué áreas se pueden utilizar.



Mapa Conceptual.

El Kit de Robótica incluye el siguiente material, el cual lo hace versátil, ya que nos permite ensamblar más de 26 robots diferentes:

Material del Kit de Robótica Bioloid Premium.

Software RoboPlus: Es un software libre que sirve para trabajar con los diferentes modelos (Ollo, Bioloid y Darwin Op) de la marca Robotis. Contiene las siguientes herramientas:

- RoboPlus Task
- RoboPlus Manager
- RoboPlus Motion
- RoboPlus Terminal
- Dynamixel Wizard
- Guía del Usuario
- e-Manual

8

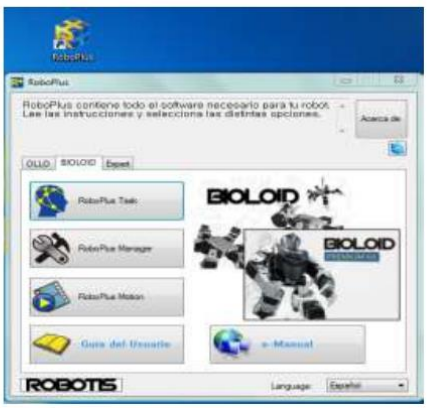
PRÁCTICA 1

OPERACIÓN DEL ROBOT BIOLOID GP CON CONTROL REMOTO EN MODO ESTÁNDAR

MATERIAL Y EQUIPO A UTILIZAR:

Para la realización de esta práctica son necesarios los siguientes componentes:

- Robot Bioloid GP
- Controlador CM-530
- Módulos BT 4.10 (Bluetooth).
- RC-100 (Control remoto)
- Programa Robot Plus



Software RobotPlus

Robot Bioloid GP

RC-100

Controlador CM-530

Módulos BT 4.10

OBJETIVOS:

General:

Operar el Robot Bioloid Gp con el control remoto en el modo estándar.

Específicos:

º Cargar la programación del modo estándar al controlador CM-530.
º Probar el modo estándar con el control remoto.
º Ejecutar combinación de botones, y verificar el cumplimiento de los movimientos del robot.

DESARROLLO:

Cargar la programación del robot al controlador CM-530 usando el programa RobotPlus Task, el nombre del archivo es: "Control Remoto", donde estará predefinido la programación del robot con los comandos establecidos en él, programa dichos comandos se utilizarán para poner en práctica los algoritmos previamente programados usando el control remoto, daremos marcha al robot y probaremos dicha programación.

Una vez que el programa se encuentre cargado en el controlador, pulsaremos en botón "U" en el controlador CM-530, para que ejecute su caminata estándar. Al presionar dicho botón, el controlador emitirá una melodía, que nos indicará que función vamos a ejecutar.

Para el uso del control utilizaremos los siguientes comandos previamente definidas, en la programación que hemos cargado a nuestro robot, que son los siguientes:

Buttons	Motion	Buttons	Motion
U	Adelante	D	Atrás
L	Giro Izquierda	R	Giro Derecha
U + L	Caminar hacia adelante + izquierda	U + R	Caminar hacia adelante + derecha
L + 5	Paso lateral izquierdo	L + 5 + 6	Paso lateral izquierdo rápido
L + U + 5	Paso diagonal delantero izquierdo	L + D + 5	Paso diagonal hacia atrás a la izquierda
R + 5	Paso lateral derecho	R + 5 + 6	Paso lateral derecho rápido
R + U + 5	Paso diagonal derecho hacia adelante	R + D + 5	Paso diagonal hacia atrás derecho
1 + U	se levanta boca arriba	1 + D	se levanta boca abajo

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR

1C

PRÁCTICA 2

OPERACIÓN DEL ROBOT BIOLOID GP CON CONTROL REMOTO EN MODO SOCCER

MATERIAL Y EQUIPO A UTILIZAR:

Para la realización de esta práctica son necesarios los siguientes componentes:

- Robot Bioloid GP
- Controlador CM-530
- Modulos BT 4.10 (Bluetooth).
- RC-100 (Control remoto)
- Programa Robot Plus

Software RobotPlus

Robot Bioloid GP

RC-100

Controlador CM-530

Módulos BT 4.10

OBJETIVOS:

General:

Operar el Robot Bioloid Gp con el control remoto en el modo estándar.

Específicos:

º Cargar la programación del modo estándar al controlador CM-530.
º Probar el modo soccer con el control remoto.
º Ejecutar combinación de botones, y verificar el cumplimiento de los movimientos del robot.

DESARROLLO:

Cargar la programación del robot al controlador CM-530 usando el programa RobotPlus Task, el nombre del archivo es: "Control Remoto", donde estará predefinido la programación del robot con los comandos establecidos en él, programa dichos comandos se utilizarán para poner en práctica los algoritmos previamente programados usando el control remoto, daremos marcha al robot y probaremos dicha programación.

Una vez que el programa se encuentre cargado en el controlador, pulsaremos en botón "D" en el controlador CM-530, para que ejecute su caminata estándar. Al presionar dicho botón, el controlador emitirá una melodía, que nos indicará que función vamos a ejecutar.

Para el uso del control utilizaremos los siguientes comandos previamente definidas, en la programación que hemos cargado a nuestro robot, que son los siguientes:

Buttons	Motion	Buttons	Motion
2 + U	Pierna Izquierda + Patada Hacia Adelante	4 + U	Pierna Izquierda + Patada Hacia Adelante
2 + D	Pierna Izquierda + Patada Atrás	4 + D	Pierna Derecha + Patada Atrás
2 + L	Pierna Izquierda + Patada Izquierda	4 + L	Pierna Derecha + Patada Izquierda
2 + R	Pierna Izquierda + Patada Derecha	4 + R	Pierna Derecha + Patada Derecha
3	Defensa en espera	3 + L	Bola de Bloqueo + Izquierda
3 + U	Defensa	3 + R	Bola de Bloqueo + Derecha

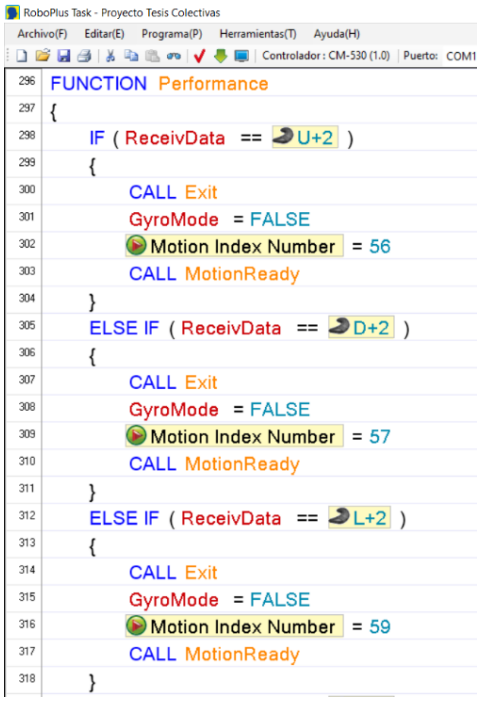


Imagen 43 Prácticas Comunes # 1

Imagen 44 Prácticas comunes # 2**Imagen 45** Prácticas Comunes # 1

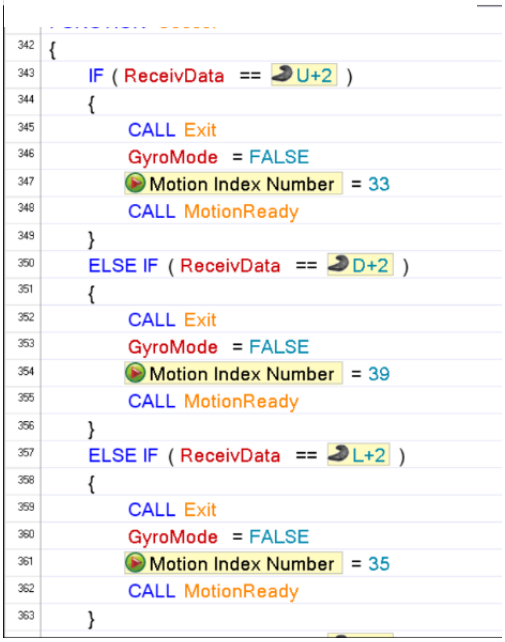


Imagen 46 Prácticas comunes # 2

Imagen 47 Prácticas comunes # 2

5.2. **Manual de prácticas de robot humanoide Bioloid GP**

1

UNIDAD DE APRENDIZAJE
DE ROBÓTICA

MANUAL DE PRÁCTICAS DE ROBOT HUMANOIDE BIOLOID GP

PROGRAMA DE INGENIERÍA ELECTRONICA

UNIVERSIDAD POLITECNICA SALESIANA
SEDE GUAYAQUIL

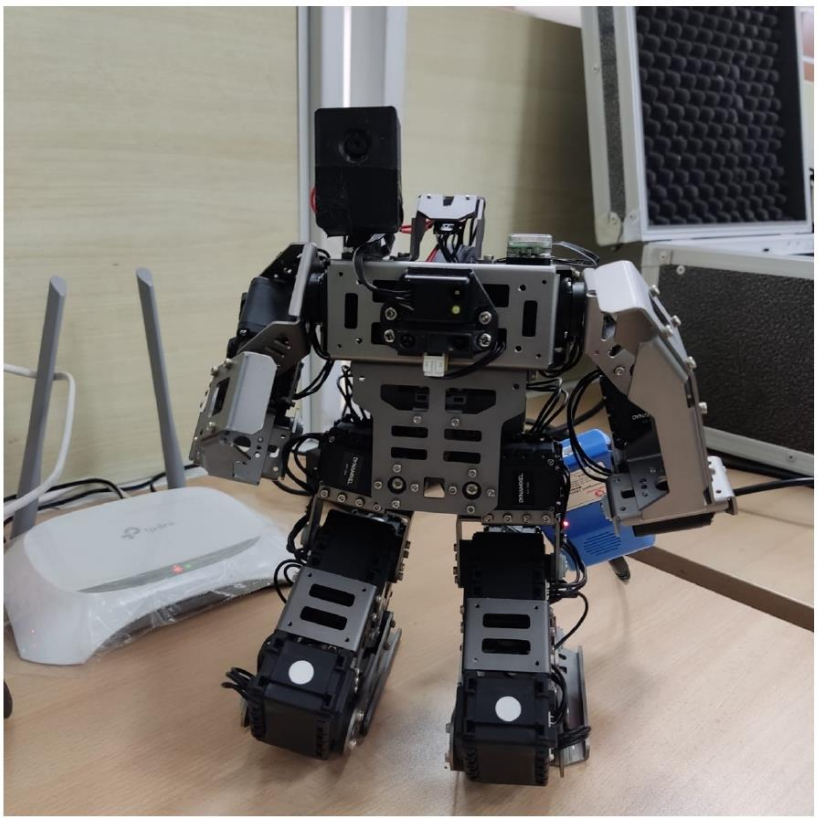
AUTORES:

RURIK ALFREDO VALLADARES YANQUI

CÉSAR ARTURO VALDIVIEZO ROMERO

ECUADOR - GUAYAQUIL

MANUAL DE PRÁCTICA DE ROBÓTICA



PRESENTACIÓN

Las presentes prácticas robótica fueron desarrolladas basadas en un programa de práctica básica sobre el diseño e implementación en Robótica con la finalidad de que los estudiantes puedan poner en práctica los conocimientos básicos teóricos que se van adquiriendo a lo largo de la asignatura Robótica, este manual práctico consta de la siguiente estructura:

1. Comprender los conceptos básicos de Robótica móvil identificando los conceptos y funciones importantes.
2. Identificar los tipos de sensores utilizados en el Robot
3. Conocer los programas a utilizar y sus funciones.

Cada práctica consta de un título, un objetivo, una introducción, materiales y equipos a utilizar, instrucciones, recomendaciones, desarrollo y programación. El estudiante debe de replicar de forma práctica lo que se encuentra en el apartado de desarrollo y resolver lo que se pide en el apartado de evaluación.

INTRODUCCIÓN

Generalmente, un robot es un dispositivo mecánico articulado capaz de imitar ciertas funciones humanas como la como la movilidad, la detección de objetos sus colores, como el mover objetos y poder ir de un lugar a otro y además, con el fin de sustituir ciertas funcionalidades humanas en ciertas tareas específicas, está realizaciones son más o menos autónoma pero limitadas según las facultades sensoriales de percepción del entorno con la cual el robot se ha equipado y programado. La robótica se la considera como el conjunto de actividades de construcción y puesta en marcha de los robots.

Finalmente, por exceso de lenguaje consideramos como un sistema robotizado a cualquier dispositivo automatizado. Sin embargo, la principal diferencia entre la robótica y la automatización es que la parte operacional del sistema es un sistema mecánico articulado de todas formas, también que los conceptos generales utilizados en la automatización se pueden aplicar también a la robótica.

Actualmente la robótica ha sido de gran impacto tanto en la sociedad como en la ciencia y en educación y en la programación y de gran ayuda para la vida cotidiana y de los procesos industriales y tecnológicos todo esto se ha enfocado en las líneas de producción donde el monitoreo de la producción

desempeña un rol muy importante por lo que se por lo que se por lo que se puede desarrollar un sistema automatizado que genere buenos beneficios

El término “Robot” proviene de la palabra robota qué quiere decir “labor forzada” o “servicio o esclavo”.

La Robótica educativa es el conjunto de actividades pedagógicas que apoyan y fortalecen áreas específicas del conocimiento y desarrollan habilidades y competencias en los estudiantes Mediante los siguientes procesos Concepción creación ensamble y puesta en funcionamiento de robot mediante la manipulación del kit de robótica se lleva a cabo el ensamble de hasta 3 diferentes tipos de robots móviles las cuales permiten ilustrar los principios y fundamentos básicos de la robótica móvil la robótica móvil es una ciencia una rama de la tecnología que estudia el diseño y creación de máquinas que pueden desplazarse de un lugar a otro pueden moverse pueden detectar objetos puede interactuar con los colores de su entorno también cuya finalidad es desempeñar tareas repetitivas o peligrosas qué el ser humano no puede y es necesario la utilidad de la inteligencia artificial por el ser humano.

PRÁCTICA 1

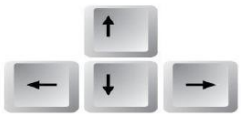
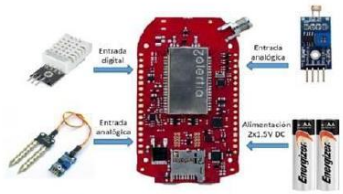
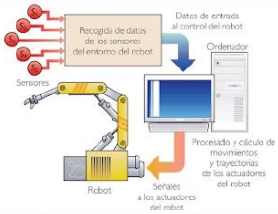
Título: Operación del robot con control remoto en modo Bluetooth

Materiales y Equipo

Para realizar la realización de esta práctica son necesarios los siguientes componentes:

- RoboPlus Task
- Robot Bioloid GP
- Controlador BT 4.10 (Bluetooth)
- RC-100 (Control Remoto)
- Programa Robot Plus

Es necesario que el estudiante conozca qué es familiarizarse con un robot humanoide y que es solo familiarizarse con un controlador de un robot.

Elementos y componentes	Observaciones
	Teclas direccionales
	Sensores inalámbricos
	Proceso de inalámbrico
	Pendrive Bluetooth

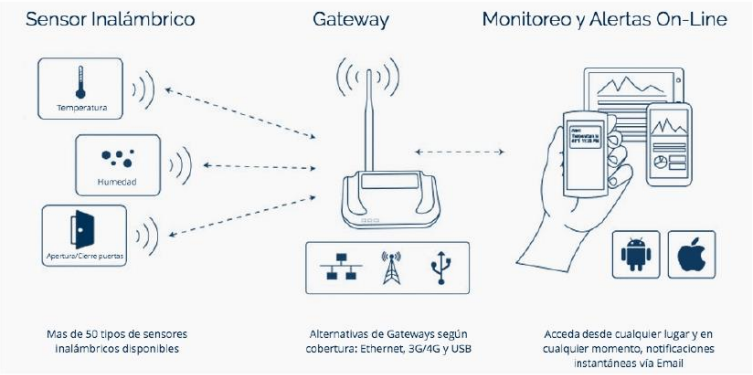
	Switch

Objetivo

El estudiante deberá conocer los elementos principales sobre la Operacionalidad del robot para una mejor compresión de la práctica, a su vez debe identificar y familiarizarse con cada uno de los elementos y sus principales componentes.

Desarrollo

El estado del robot inicialmente está apagado, se enciende con el botón de encendido, se conecta la Raspberry con el cable de poder e inmediatamente se activa el Bluetooth. Gracias al comando previamente ejecutado permite que el robot pueda caminar de forma inalámbrica, sin necesidad de estar conectado por medio de un cable.



Instrucciones básicas:

- Identifica que cada una del elemento constituyente del robot humanoide
- Manipule el robot apagando y encendiendo.

Funcionalidades

1. Registro automático de la lectura de los sensores
2. La aplicación incluye alertas vía correo electrónico
3. Notificaciones configurables a una cadena de usuarios. Si el responsable no acusa recibo, entonces el siguiente de la cadena es notificado.
4. Reportes de gestión
5. Permite generar gráficos en dispositivos móviles y PC
6. Proporciona datos históricos para fines auditables
7. Reporte diario de cumplimiento normativo área Salud: (Temperaturas Máximas, Mínimas, Cinética Media y MKT)
8. Mapa con ubicación y estado de los sensores
9. Sensores alimentados por baterías con más de 10 años de duración.
10. No requieren acceso a la red Wifi.
11. De fácil instalación

PRÁCTICA 2

Título: Operación del robot con los sistemas de movimiento y sistema de coordenadas.

Materiales y Equipo

Elementos de que componente de un controlador de un robot:

- Tarjetas de para servomotores
- Fuente de poder
- Computador

- Tarjeta I/O
- Puesto de comunicación
- Botonería

Objetivo

El estudiante deberá conocer, mover, y dominar los diferentes tipos de movimiento de un robot, así como Joint, Rectangular, Herramientas y de usuario.

Encendido correcto del Robot

- Conectar la clavija de alimentación
- Accionar el interruptor principal
- Accionar el switch

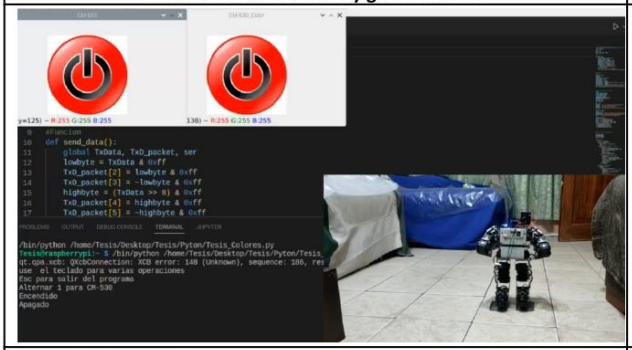
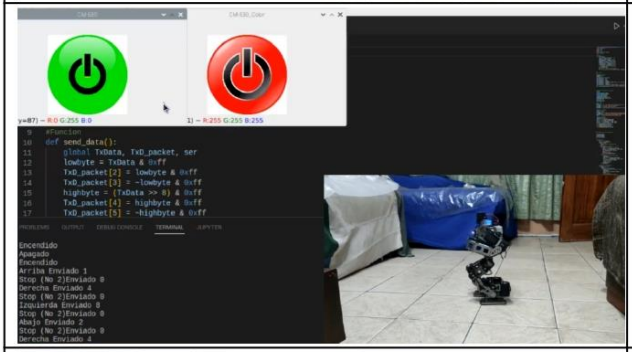
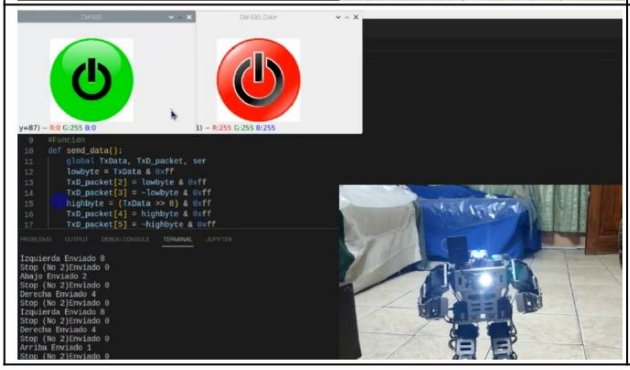
Sistema de Coordenada

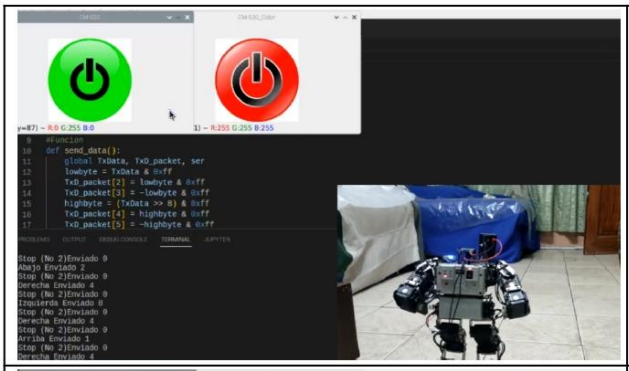
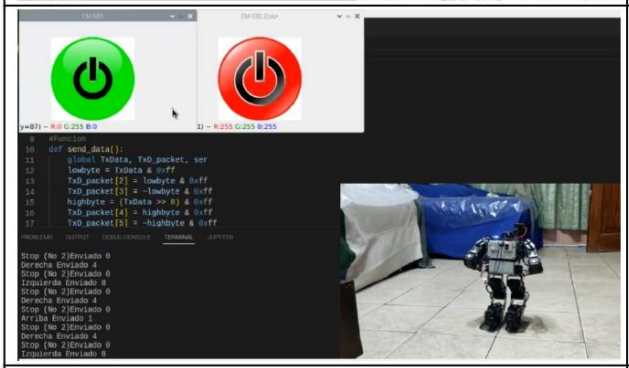
- Para cambiar el sistema de coordenadas únicamente se presiona el botón.
- Sistema de coordenada Rectangulares
- Sistema de coordenadas TOOL
- Sistema de coordenada USER

Desarrollo

Bueno en el eje en esta práctica se llevará a cabo las instrucciones básicas para el control de movimiento del robot y para esto se desea programar una tarea llamada “hacia delante” o “hacia atrás”, en la que los pies del robot se deslizan desde un punto hacia otro. El inicia desde una posición inicial o de reposo y se mueve hacia una posición final aproximadamente de 50 metros posteriormente debe girar y luego regresar al punto de inicio.

La siguiente imagen representa a un robot en movimiento.

Estados de movimiento y giro del Robot	Observación
	Estado apagado
	Estado encendido Hacia adelante, gira hacia la izquierda, gira hacia la derecha y luego de retro.
	Girando hacia adelante por la izquierda y luego se mueve hacia adelante.

	Girando hacia la derecha
	Moviéndose hacia delante.

PRÁCTICA 3

Título: Operación del robot con la detección de Objetos y su forma

Materiales y Equipo

Para realizar la realización de esta práctica son necesarios los siguientes componentes

- Cámara
- Controlador

Objetivo

El estudiante deberá saber e identificar las diferentes formas de los objetos detectados por el robot.

Instrucciones

Diseñar un programa que nos permite la adquisición de datos y comunicación entre el robot y el mando inalámbrico en el mandato inalámbrico de los sensores delantero y posterior montados en el robot.

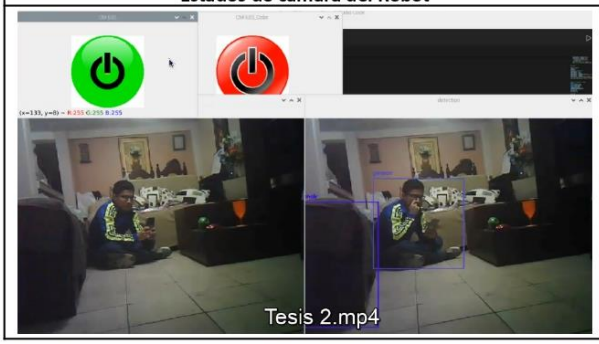
Recomendaciones

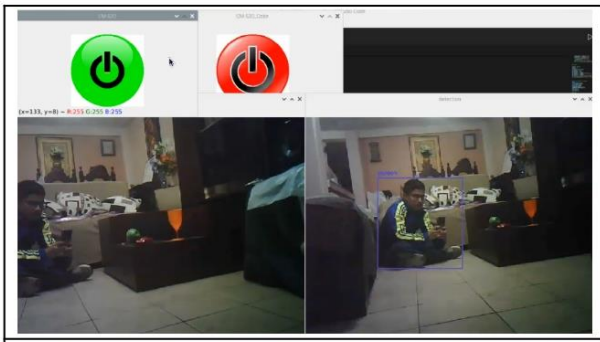
- Investigar datos técnicos acerca de los sensores
- Tomar previas de realización.

Desarrollo

Esta práctica llevará a cabo las detecciones básicas de los objetivos para la identificación de las formas del robot. Para la realización de esta práctica se bajo un programa previamente establecido de internet y se lo acopló a nuestro programa, para que el robot sea capaz de reconocer diferentes tipos de objetos.

La siguiente tabla de imágenes representa a un robot detectando

Estados de cámara del Robot	Observación
	Detectar los objetos según su forma

	Detecta la copa de vino

Programación

Estructura de Programación en el Lenguaje

Tesis-CAM.py

```
1  import cv2
2  import matplotlib.pyplot as plt
3  import cvlib as cv
4  import urllib.request
5  import numpy as np
6  from cvlib.object_detection import draw_bbox
7  import concurrent.futures
8
9  url='http://192.168.1.110/cam-hi.jpg'
10 im=None
```

Tesis-CAM.py

```
12 def run1():
13     #cv2.namedWindow("live transmission", cv2.WINDOW_AUTOSIZE)
14     cv2.namedWindow("live transmission", cv2.WINDOW_NORMAL)
15     while True:
16         img_resp=urllib.request.urlopen(url)
17         imgnp=np.array(bytearray(img_resp.read()),dtype=np.uint8)
18         im = cv2.imdecode(imgnp,-1)
19         #live_transmission = cv2.resize('live transmission', (150, 150))
20         cv2.moveWindow('live transmission',0,200)
21         cv2.resizeWindow('live transmission', 600, 500)
22         cv2.imshow('live transmission',im)
23         key=cv2.waitKey(5)
24         if key==ord('q'):
25             break
26
27     cv2.destroyAllWindows()
28
```

Tesis-CAM.py

```
def run2():
    #cv2.namedWindow("detection", cv2.WINDOW_AUTOSIZE)
    cv2.namedWindow("detection", cv2.WINDOW_NORMAL)
    while True:
        img_resp=urllib.request.urlopen(url)
        imgnp=np.array(bytearray(img_resp.read()),dtype=np.uint8)

        im = cv2.imdecode(imgnp,-1)

        bbox, label, conf = cv.detect_common_objects(im)
        im = draw_bbox(im, bbox, label, conf)
        cv2.moveWindow('detection',600,200)
        cv2.resizeWindow('detection', 600, 500)

        cv2.imshow('detection',im)
        key=cv2.waitKey(5)
        if key==ord('q'):
            break

    cv2.destroyAllWindows()
```

```
if __name__ == '__main__':
    print("started")
    with concurrent.futures.ProcessPoolExecutor() as executer:
            f1= executer.submit(run1)
            f2= executer.submit(run2)
```

url dirección ip de la camar

PRÁCTICA 4

Título: Operación del robot con la detección de Objetos y sus colores

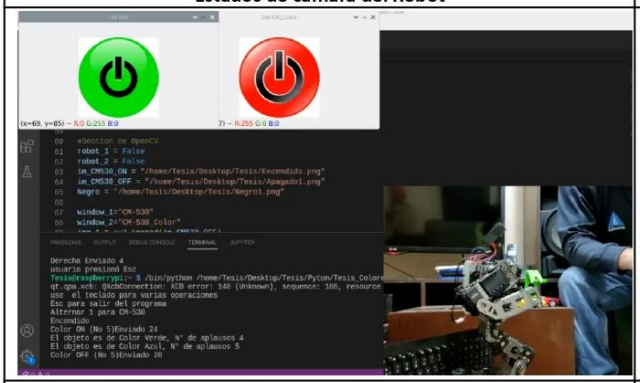
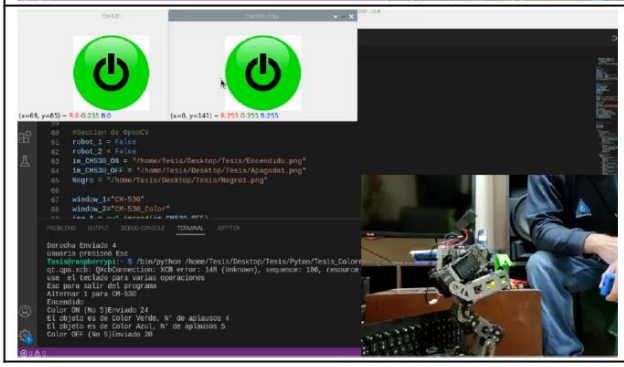
Objetivo

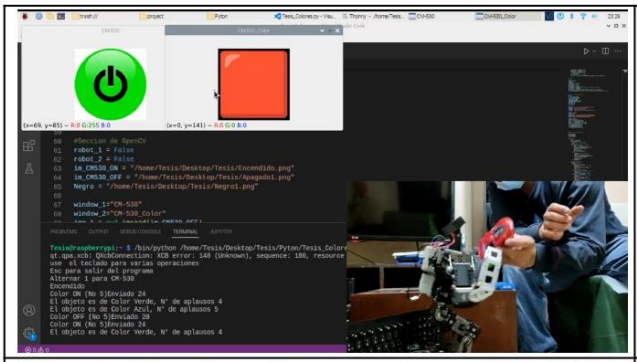
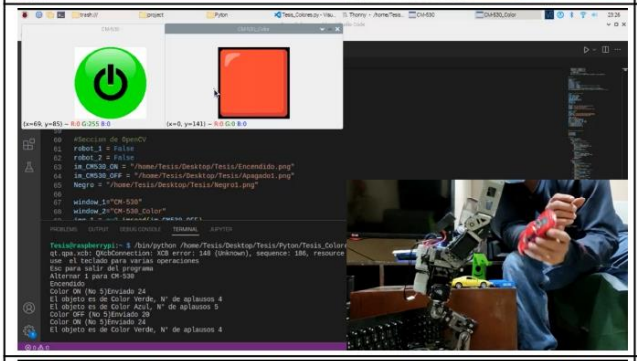
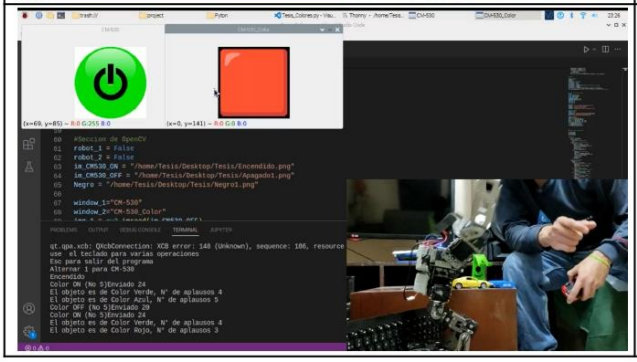
El estudiante deberá comparar las acciones físicas e identificar los colores respectivos en cada acción ejecutada por el robot.

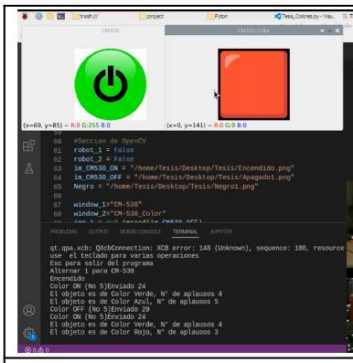
Desarrollo

En esta práctica estamos llevando a cabo instrucciones para la identificación de objetos de acuerdo a su color, para esta práctica hicimos que el robot haga un número de aplausos por cada color que detecte. Siendo así que si detecta el color rojo, el robot dará 3 aplausos. Si detecta el color verde, dará 4 aplausos.

Si detecta el color azul, dará un número de 5 aplausos. Si detecta el color amarillo, dara 6 aplausos.

Estados de cámara del Robot	Observación
	Se prepara para la detección de colores.
	Sensa el color azul.

	Da cierto número de aplausos al detectar el color rojo.
	
	



Programación

Estructura de Programación en Lenguaje

Tesis_Colores.py

```
#Variables
#
#PySerial section
highbyte = 0
lowbyte = 0
TxD_packet = bytearray(6)
TxD_packet[0] = 0xff
TxD_packet[1] = 0x55
TxD_packet[2] = lowbyte
TxD_packet[3] = ~lowbyte & 0xff #asegura el contenido de ~lowbyte
TxD_packet[4] = highbyte
TxD_packet[5] = ~highbyte & 0xff #asegura el contenido de ~highbyte
ser = serial.Serial("/dev/ttyUSB0",57600, timeout=0.1, write_timeout=0.1)
ser.reset_output_buffer() #Limmpia el buffer usado para el envio de dato al CM-530
```

Tesis_Colores.py

```


#Seccion de interfaz de teclado
ESC=27
FLECHA_DERECHA= 65361
FLECHA_IZQUIERDA= 65363
FLECHA_ARRIBA= 65362
FLECHA_ABAJO= 65364
LEVANTARSE_FRENTE_No_3= 51
LEVANTARSE_ESPALDA_No_4= 52
STOP_No_2= 50
Color_No_5= 53
Color_No_6= 54

No_1 = 49
message1 = "use  el teclado para varias operaciones"
message2 = "Esc para salir del programa"
message3 = "Alternar 1 para CM-530"
message4 = "Tecla derecha o Tecla Izquierda para rotar"
```

Tesis_Colores.py

```
#Seccion de OpenCV
robot_1 = False
robot_2 = False
im_CM530_ON = "/home/Tesis/Desktop/Tesis/Encendido.png"
im_CM530_OFF = "/home/Tesis/Desktop/Tesis/Apagado1.png"
Negro = "/home/Tesis/Desktop/Tesis/Negro1.png"

window_1="CM-530"
window_2="CM-530_Color"
img_1 = cv2.imread(im_CM530_OFF)
img_2 = cv2.imread(im_CM530_OFF)
img_1S = cv2.resize(img_1, (150, 150))
img_2S = cv2.resize(img_2, (150, 150))
cv2.imshow(window_1,img_1S)
cv2.moveWindow(window_1,10,50)
cv2.imshow(window_2,img_2S)
cv2.moveWindow(window_2,310,50)
```

Tesis_Colores.py

```
#Programa Inicial comienza aqui
#
print(message1)
print(message2)
print(message3)
#print(message4)

while True:
    #Seccion de Teclado
    key = cv2.waitKeyEx(33)
    if(key == ESC):
        print("usuario presionó Esc")
        break
    if(key == No_1): # alternar en tecla 1
        if(robot_1 == False):
            img_1 = cv2.imread(im_CM530_ON)
            img_1S = cv2.resize(img_1, (150, 150))
            cv2.imshow(window_1,img_1S)
            robot_1 = True
            print("Encendido")
            ser.reset_output_buffer()
        else:
            TxData+= 0 #Valor STOP TxData
            send_data()
            img_1 = cv2.imread(im_CM530_OFF)
            img_1S = cv2.resize(img_1, (150, 150))
            cv2.imshow(window_1,img_1S)
            robot_1 = False
            print("Apagado")
        cv2.waitKey(1)
```

Tesis_Colores.py

```
    #Preparando comunicación
    if(robot_1 == True):
        TxData = 0
        if(key == FLECHA_DERECHA):
            TxData+= 4 #Valor derecha de  TxData
            send_data()
            print("Derecha Enviado",TxData)
            TxData = 0
            #time.sleep(2.0)
        elif (key == FLECHA_IZQUIERDA):
            TxData+= 8 #Valor izquierda de  TxData
            send_data()
            print("Izquierda Enviado",TxData)
        elif (key == FLECHA_ARRIBA):
            TxData+= 1 #Valor Arriba de  TxData
            send_data()
            print("Arriba Enviado",TxData)
        elif (key == FLECHA_ABAJO):
            TxData+= 2 #Valor Abajo de  TxData
            send_data()
            print("Abajo Enviado",TxData)
        elif (key == LEVANTARSE_FRENTE_No_3):
            TxData+= 17 #Valor Levantarse de frente de  TxData
            send_data()
            print("Levantarse de frente (No 3)  Enviado",TxData)
```

Tesis_Colores.py

```
elif (key == LEVANTARSE_ESPALDA_No_4):
    TxData+= 18 #Valor Levantarse de espalda de  TxData
    send_data()
    print("Levantarse de espalda (No 4) Enviado",TxData)
elif (key == STOP_No_2):
    TxData+= 0 #Valor STOP TxData
    send_data()
    print("Stop (No 2)Enviado",TxData)

elif (key == Color_No_5): # alternar en tecla 5
    if(robot_2 == False):
        img_2 = cv2.imread(im_CM530_ON)
        img_2S = cv2.resize(img_2, (150, 150))
        cv2.imshow(window_2,img_2S)
        robot_2 = True
        TxData+= 24 #Valor STOP TxData
        send_data()
        print("Color ON (No 5)Enviado",TxData)

    else:
        TxData+= 20 #Valor STOP TxData
        send_data()
        img_2 = cv2.imread(im_CM530_OFF)
        img_2S = cv2.resize(img_2, (150, 150))
        cv2.imshow(window_2,img_2S)
        robot_2 = False
        print("Color OFF (No 5)Enviado",TxData)
```

Tesis_Colores.py

```
if ser.in_waiting > 0:
    line = ser.readline().decode('utf-8').rstrip()

    if(line == "    1"):
        img_2 = cv2.imread("/home/Tesis/Desktop/Tesis/Blanco1.png")
        img_2S = cv2.resize(img_2, (150, 150))
        cv2.imshow(window_2,img_2S)
        print("El objeto es de Color Blanco, N° de aplausos 1")
    elif(line == "    2"):
        img_2 = cv2.imread(Negro)
        img_2S = cv2.resize(img_2, (150, 150))
        cv2.imshow(window_2,img_2S)
        print("El objeto es de Color Negro")
    elif (line == "    3"):
        img_2 = cv2.imread("/home/Tesis/Desktop/Tesis/rojo1.png")
        img_2S = cv2.resize(img_2, (150, 150))
        cv2.imshow(window_2,img_2S)
```

Tesis_Colores.py

```
        print("El objeto es de Color Rojo, N° de aplausos 3")
    elif (line == "    4"):
        img_2 = cv2.imread("/home/Tesis/Desktop/Tesis/Verde1.png")
        img_2S = cv2.resize(img_2, (150, 150))
        cv2.imshow(window_2,img_2S)
        print("El objeto es de Color Verde, N° de aplausos 4")
    elif (line == "    5"):
        img_2 = cv2.imread("/home/Tesis/Desktop/Tesis/Azul1.png")
        img_2S = cv2.resize(img_2, (150, 150))
        cv2.imshow(window_2,img_2S)
        print("El objeto es de Color Azul, N° de aplausos 5")
    elif (line == "    6"):
        img_2 = cv2.imread("/home/Tesis/Desktop/Tesis/Amarillo1.png")
        img_2S = cv2.resize(img_2, (150, 150))
        cv2.imshow(window_2,img_2S)
        print("El objeto es de Color Amarillo, N° de aplausos 6")
    #print(line)

#Final del Bucle Inicial
cv2.destroyAllWindows()
sys.exit()
```

Es la comunicación serial entre el Raspberry y el Bioloid

Conclusiones

- Gracias al software Phyton se pudo desarrollar a cabalidad la programación que posteriormente sería para el funcionamiento prototipo del Robot Bioloid GP.
- El beneficio de la tecnología de comunicación inalámbrica Bluetooth pudo ser usada en la transmisión de datos al prototipo del Robot Bioloid GP evitando problemas operativos y límite de distancia funcionamiento del prototipo, debido que este no depende de conexión por cable.
- Para optimizar la conectividad se hizo uso de cable Ethernet (UTP) ya que este ofrece mejores prestaciones de velocidad y estabilidad en la trasmisión de datos evitando latencia en el envío de datos.
- Se concluye que después de hacer las respectivas pruebas con el prototipo Bioloid GP, se obtuvieron los resultados esperados.

Recomendaciones

- Se recomienda que para realizar las pruebas esperadas se coloque al prototipo una batería con más potencia, ya que, la que viene por defecto con el robot pierde su carga muy rápido.
- En lo posible se recomienda de que el prototipo esté libre de elementos o elementos livianos encima de su cuerpo ya que pueden hacerle contrapeso y dificultar los movimientos del robot.
- Es recomendable seguir desarrollando actualizaciones para que a largo plazo, el mismo pueda cumplir funciones complejas tanto administrativas como operativas en un periodo de tiempo reducido.

Bibliografía

Adam, L. (2019). *Raspberry Pi 4: What is it and how can you use it?* . Retrieved from https://hackernoon.com/raspberry-pi-4-what-is-it-and-how-can-you-use-it-jw26e30kb

Barrientos, A., Peñin, L., Balaguer, C., & Aracil, R. (1997). *Fundamentos de robótica.* McGRAW-HILL/INTERAMERICANA DE ESPAÑA S.A.

Castro, A. (2019). *¿Qué es Raspberry Pi, dónde comprarla y cómo usarla?* . Retrieved from https://computerhoy.com/noticias/hardware/que-es-raspberry-pi-donde-comprarlacomo-usarla-8614

Challenger, I., Diaz, Y., & Becerra, R. (2014). *El lenguaje de programación Python.* Ciencias Holguín.

Challenger, I., Díaz, Y., & Becerra, R. A. (2019). *El lenguaje de programación Python*. Retrieved from https://www.redalyc.org/pdf/1815/181531232001.pdf

Escalante, D., & Vargas, D. (2020). *Raspberry pi: la tecnología reducida en placa*. Retrieved from https://repository.usc.edu.co/bitstream/handle/20.500.12421/4250/RASPBERRY%20PI.pdf?sequence=3&isAllowed=y

Garcia, M., Alvarez, J., & Cava, D. (2003). *Robotrónica aplicaciones de la robótica.* Universitat Politecnica de Catalunya.

Hisour. (2022). Retrieved from https://www.hisour.com/es/humanoid-robot-43044/?nonamp=1%2F

Lucas, J. (2019). *Qué es Raspberry Pi | OpenWebinars*. Retrieved from https://openwebinars.net/blog/que-es-raspberry-pi/

MCkibre. (2020). *Visual Studio Code. Configuración recomendada para el curso Introducción a la programación con Python*. Retrieved from https://www.mclibre.org/consultar/python/otros/vsc-python-configuracion.html

Minsky, M. (2022). *Filosofia*. Retrieved from https://filosofia.co/general/definicion-de-inteligencia-artificial-segun-autores/

RC-100B Remote Control. (2019). *RC-100B Remote Control*. Retrieved from https://www.generationrobots.com/en/400890-rc-100b-bioloid-wireless-remote-control-robotis.html

Revista de robots. (2022). Retrieved from https://revistaderobots.com/robots-y-robotica/que-es-un-robot-y-tipos-de-robots

ro-botica. (2019). *Guía de introducción a ROBOTIS PREMIUM*. Retrieved from https://ro-botica.com/PDF/Premium/Guia_inicio_ROBOTIS_PREMIUM.pdf

Rossum, G., & Drake, F. (2010). *The Python Library Reference. Release.*

Rouhiainen, L. (2018). *Inteligencia artificial.* Alienta Editorial.

Sandorobotics. (2022). *DYNAMIXEL AX-12A*. Retrieved from https://sandorobotics.com/producto/902-0003-001/

Sensor de distancia Sharp DMS-80. (2020). *Sensor de distancia Sharp DMS-80*. Retrieved from https://www.ro-botica.com/Producto/Sensor-de-distancia-Sharp-DMS_80/

Silva Segovia, M. (2010). *Diseño y construcción de un robot humanoide con 16 grados de libertad como material de apoyo para el área de inteligencia artificial de la escuela de ingeniería en sistemas de la Pucesa para el periodo 2009 2010.* Pontificia Universidad Católica del Ecuador Sede Ambato.

Tipos de Robots humanoide. (2019). *Tipos de Robots humanoide*. Retrieved from https://ro-botica.com/Producto/ROBOTIS-BIOLOID-GP/