



UNIVERSIDAD DEL AZUAY
FACULTAD DE CIENCIA Y TECNOLOGÍA
ESCUELA DE INGENIERÍA ELECTRÓNICA

**“Uso del LEGO Mindstorms EV3 para el Fortalecimiento
del Aprendizaje en el Campo de la Robótica.”**

Trabajo de graduación previo a la obtención del título de:

INGENIERO ELECTRÓNICO

Autores:

JORGE DAVID HUAYLLA SELLAN

CHRISTIAN GEOVANNY MOROCHO ORTEGA

Director:

ING. HUGO MARCELO TORRES SALAMEA

CUENCA, ECUADOR

2024

I. DEDICATORIA

A mis queridos padres, quienes, con amor, sacrificio y valores inquebrantables han sido mi guía y apoyo incondicional a lo largo de esta travesía académica. A mis queridos hermanos, ustedes son mi constante motivación, este logro también es suyo, con gratitud eterna.

Morocho Ortega Christian Geovanny

Dedico esta tesis a mis padres, mis hijos y mi familia, fuente inagotable de inspiración y apoyo. Su amor y sabiduría han sido mi faro en este viaje académico. A mi mentor, por su guía experta. A mis amigos, por ser mi red de contención. Este logro es también suyo. Gracias por ser mi fuente constante de motivación.

Huaylla Sellan Jorge David

II. AGRADECIMIENTO

Quiero expresar mi profundo agradecimiento al Ing. Hugo Torres, cuya orientación experta y apoyo han sido fundamentales en la realización de este trabajo de titulación. Mi reconocimiento especial a mis queridos padres y hermanos por su inquebrantable respaldo.

Este logro no habría sido posible sin su apoyo constante.

Morocho Ortega Christian Geovanny

Quiero expresar mi sincero agradecimiento a todos aquellos que contribuyeron a la realización de esta tesis. Agradezco a Dios en primer lugar, a mis padres, mi familia y mis hijos por ser ese empuje que necesité en los momentos donde daba mi brazo a torcer, a mi director de tesis por su orientación invaluable, y a mi compañero de tesis por su aliento constante. Este logro no habría sido posible sin la colaboración de todos los que mencioné anteriormente. Muchas gracias.

Huaylla Sellan Jorge David

III. ÍNDICE DE CONTENIDO

I.	DEDICATORIA	I
II.	AGRADECIMIENTO	II
III.	ÍNDICE DE CONTENIDO	III
IV.	ÍNDICE DE FIGURAS	IV
V.	RESUMEN	V
VI.	ABSTRACT	V
I.	INTRODUCCIÓN	2
II.	ESTADO DEL ARTE	2
III.	MARCO TEÓRICO	3
A.	LEGO Mindstorms EV3	3
B.	MATLAB / SIMULINK	4
C.	PYTHON	4
D.	ARDUINO	4
IV.	METODOLOGÍA	5
A.	Banda Transportadora como seleccionador de colores utilizando Simulink	5
1)	Etapa de construcción	5
2)	Etapa de programación	6
3)	Etapa de comunicación	7
B.	Control de Brazo Robótico utilizando Python	8
1)	Etapa de construcción	8
2)	Etapa de programación	9
3)	Etapa de comunicación	9
A.	Carro Transportador utilizando el sistema embebido de Arduino	10
1)	Etapa de construcción	10
2)	Etapa de comunicación	10
3)	Etapa de programación	12
V.	PRUEBAS Y RESULTADOS	12
A.	Pruebas y resultados obtenidos en la simulación del selector de colores utilizando el software de programación gráfica Simulink	13
1)	Prueba de Conexión	13
2)	Prueba de Secuencias de Movimiento:	13
B.	Pruebas y resultados obtenidos en la simulación del proceso del brazo robótico el software de programación Python	13
1)	Configuración Inicial y Conexión	13
C.	Pruebas y resultados obtenidos en la simulación del robot transportador utilizando el software de programación Arduino y la tarjeta Arduino mega	14
1)	Configuración de Comunicación entre Arduino y EV3	14
VI.	CONCLUSIONES	15
VI.	REFERENCIAS	16

IV. ÍNDICE DE FIGURAS

Fig. 1. Set de piezas del LEGO Mindstorms EV3 [17].	3
Fig. 2. Sensores, actuadores y CPU del LEGO Mindstorms EV3 [17].	4
Fig. 3. Kit completo del LEGO Mindstorms EV3 [17].	4
Fig. 4. Conexión Simulink-EV3.	4
Fig. 5. Conexión Python-EV3.	4
Fig. 6. Brick montado en la base.	5
Fig. 7. Base para la banda.	5
Fig. 8. Base para los ladrillos de colores.	5
Fig. 9. Seleccionador de colores completo.	6
Fig. 10. a) Herramienta "Add-Ons" donde estará Get Support Packages Hardware. b) Paquete de soporte Simulink para el Hardware LEGO MINDSTORMS EV3.	6
Fig. 11. Bloque de los sensores, botones y de la pantalla LCD.	6
Fig. 12. Bloques de los valores constantes para velocidad y posición de los motores.	6
Fig. 13. Bloque de los dos motores.	7
Fig. 14. Subsistemas de sensores, parámetro y actuadores.	7
Fig. 15. Bloque "Chart" (sacado de Simulink).	7
Fig. 16. Programación mediante máquinas de estado para el seleccionador de colores.	7
Fig. 17. Herramienta para la configuración de parámetros de comunicación.	7
Fig. 18. Ventana para la configuración de la comunicación Bluetooth.	7
Fig. 19. (a) Base del brazo. (b) Base del brazo identificando sensor touch y motor grande.	8
Fig. 20. Brazo robótico con motor grande para el movimiento de la articulación del brazo.	8
Fig. 21. Pinza con motor pequeño y engranaje conectados al motor para abrir y cerrar pinza.	8
Fig. 22. Brazo armado completamente.	9
Fig. 23. Visual Studio Code.	9
Fig. 24. Pasos para descargar la extensión.	9
Fig. 25. Diagrama de flujo para el brazo robótico.	9
Fig. 26. Firmware a instalar en la microSD.	9
Fig. 27. Insertar microSD.	9
Fig. 28. Cargando firmware en brick EV3.	10
Fig. 29. Estatus activo del brick EV3.	10
Fig. 30. Base del brick.	10
Fig. 31. Esquema de conexiones para el actuador o motor.	10
Fig. 32. Esquema de conexiones para los sensores.	10
Fig. 33. Esquema de conexiones del sensor de colores con la placa Arduino.	10
Fig. 34. Esquema de conexiones para el sensor de colores con la placa Arduino MEGA.	11
Fig. 35. Esquema de conexiones con un motor.	11
Fig. 36. Esquema de conexiones resistencias pull down.	11
Fig. 37. Esquema de conexiones del sensor touch.	11
Fig. 38. Conector RJ12 adecuado para ajustarse a la entrada del hardware.	12
Fig. 39. Conector RJ12 adecuado para ajustarse a la entrada del hardware.	12
Fig. 40. Robot móvil armado.	12
Fig. 41. Creación de variables e inclusión de librerías.	12
Fig. 42. Distancia de comunicación inalámbrica entre 0-5 metros.	13
Fig. 43. Mensaje de error al no poder enlazarse con el dispositivo.	13
Fig. 44. Prueba sensores y actuadores.	13
Fig. 45. Desplazamiento de la base de los ladrillos de colores.	13
Fig. 46. Sensor colocado en una posición adecuada para el reconocimiento del color.	13
Fig. 47. (a) Conexión, (b) Estado conectado.	14
Fig. 48. Sensor o actuador desconectado.	14
Fig. 49. Objeto colocado en el lugar para ser detectado.	14
Fig. 52. Robot móvil o robot transportador.	15
Fig. 53. Conexión entre Arduino Mega, controlador L298N y placa externa.	15



V. RESUMEN

Este trabajo de titulación examina el impacto educativo del LEGO Mindstorms EV3 en robótica, destacando su eficacia para fortalecer habilidades técnicas y conceptos de ingeniería y programación. La integración de plataformas como Simulink, Python y Arduino, junto con la creación exitosa de aplicaciones prácticas como: un seleccionador de colores, un brazo robótico y un robot transportador, indican la utilidad concreta de este enfoque multidisciplinario. Los resultados obtenidos sugieren que la combinación del LEGO Mindstorms EV3 con estas herramientas puede ser estratégica para enriquecer la experiencia educativa en robótica, fomentando tanto habilidades técnicas como el pensamiento crítico y la resolución de problemas.

Palabras clave: Simulink, Python, Arduino, Brick EV3, LEGO Mindstorms.

VI. ABSTRACT

This degree work examines the educational impact of LEGO Mindstorms EV3 in robotics, emphasizing its effectiveness in enhancing technical skills, engineering and programming concepts. The integration of platforms such as Simulink, Python, and Arduino, coupled with the successful development of practical applications such as a color selector, a robotic arm, and a conveyor robot, demonstrates the tangible utility of this multidisciplinary approach. The obtained results suggest that combining LEGO Mindstorms EV3 with these tools can be strategic in enriching the educational experience in robotics, fostering both technical skills and critical thinking, and promoting problem-solving abilities.

Keywords: Simulink, Python, Arduino, Brick EV3, LEGO Mindstorms.

Uso del LEGO Mindstorms EV3 para el Fortalecimiento del Aprendizaje en el Campo de la Robótica

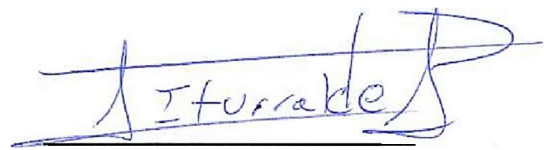
Este trabajo de titulación examina el impacto educativo del LEGO Mindstorm EV3 en robótica, destacando su eficacia para fortalecer habilidades técnicas y conceptos de ingeniería y programación. La integración de plataformas como Simulink, Python y Arduino, junto con la creación exitosa de aplicaciones prácticas como: un seleccionador de colores, un brazo robótico y un robot transportador, indican la utilidad concreta de este enfoque multidisciplinario. Los resultados obtenidos sugieren que la combinación del LEGO Mindstorms EV3 con estas herramientas puede ser estratégica para enriquecer la experiencia educativa en robótica, fomentando tanto habilidades técnicas como el pensamiento crítico y la resolución de problemas.

Palabras clave: Simulink, Python, Arduino, Brick EV3, LEGO Mindstorms.



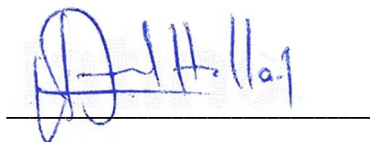
Ing. Hugo Marcelo Torres Salamea

Director de Tesis



Ing. Daniel Esteban Iturralde Piedra

Director de Escuela



Jorge David Huaylla Sellan

Autor



Christian Geovanny Morocho Ortega

Autor

Uso del Lego Mindstorm EV3 para el Fortalecimiento del Aprendizaje en el Campo de la Robótica

Jorge David Huaylla Sellan
Ingeniería Electrónica
Universidad del Azuay
Cuenca, Ecuador
darkdave18@es.uazuay.edu.ec

Christian Geovanny Morocho Ortega
Ingeniería Electrónica
Universidad del Azuay
Cuenca, Ecuador
chris_0496@es.uazuay.edu.ec

Abstract— This degree work examines the educational impact of LEGO Mindstorms EV3 in robotics, emphasizing its effectiveness in enhancing technical skills, engineering and programming concepts. The integration of platforms such as Simulink, Python, and Arduino, coupled with the successful development of practical applications such as a color selector, a robotic arm, and a conveyor robot, demonstrates the tangible utility of this multidisciplinary approach. The obtained results suggest that combining LEGO Mindstorms EV3 with these tools can be strategic in enriching the educational experience in robotics, fostering both technical skills and critical thinking, and promoting problem-solving abilities.

Keywords—*Simulink, Python, Arduino, Brick EV3, LEGO Mindstorms.*

I. INTRODUCCIÓN

La robótica es una tecnología en constante evolución que ofrece una amplia variedad de oportunidades para el aprendizaje y la innovación. Sin embargo, para jóvenes de tercero de bachillerato y primeros años de universidad puede haber varios desafíos que limitan su participación en el campo de la robótica [1]. Uno de los problemas que se puede encontrar es el acceso a recursos, materiales, el costo de los robots y los componentes en donde todo esto puede ser un obstáculo para el aprendizaje, al igual que la falta de conocimientos técnicos hacen que la robótica requiera de una comprensión sólida de las matemáticas, la física, la electrónica y la programación, lo que puede ser intimidante para algunos jóvenes que no han tenido experiencia previa en estas áreas [2] [3]. Es importante abordar estos desafíos para promover una mayor participación de jóvenes en la robótica y aprovechar al máximo su potencial para el aprendizaje y la innovación.

La robótica es una disciplina que combina la ingeniería, la programación y la tecnología para desarrollar sistemas automatizados. La inclusión de la robótica en la educación secundaria y universitaria es importante para preparar a los jóvenes para un futuro en el que la automatización y la tecnología serán cada vez más prevalentes.

Además, la robótica desarrolla habilidades valiosas en los jóvenes, como la resolución de problemas, la colaboración, la creatividad y la lógica. Estas habilidades son esenciales no solo para una carrera en robótica, sino también para muchos otros campos tecnológicos y científicos.

Definitivamente, la robótica también puede ser una herramienta motivadora para los jóvenes, ya que les permite aplicar sus conocimientos teóricos a proyectos prácticos y ver los resultados tangibles de su trabajo. Esto puede ayudar a aumentar su interés en la ciencia y la tecnología, y puede ser especialmente valioso para aquellos jóvenes que pueden no estar interesados en otras áreas de estudio.

Y es por ese motivo que se ha tomado en consideración realizar el proyecto de robótica utilizando el kit LEGO Mindstorms EV3 haciendo uso de diferentes plataformas para programar y controlar sus actuadores y sensores con el objetivo de lograr el fortalecimiento del aprendizaje de la robótica en los estudiantes.

II. ESTADO DEL ARTE

El robot LEGO Mindstorms es utilizado para la realización de proyectos que pueden ser diseñados por los propios estudiantes, donde empiezan a desarrollar una teoría de control básica con la visión de diseñar sistemas de control modernos, logrando integrar las materias fundamentales de robótica y control. El estudio realizado se enfoca en elaborar un instructivo para la implementación de procesos industriales utilizando el robot LEGO Mindstorms EV3 [4].

La aceptación de la robótica en la educación es un tema que no se debe pasar por alto al ser crucial para la era de la revolución de la industria 4.0. Por eso se propone explorar la aceptación de LEGO Mindstorms EV3 como un tipo de tecnología robótica y un recurso de aprendizaje que puede desarrollar las habilidades de los docentes y los estudiantes [5]. Es por eso que, en un curso de sistemas de control de pregrado, con la finalidad de mejorar las experiencias de aprendizaje de los estudiantes, se utiliza Matlab como herramienta de desarrollo [6].

En otro estudio se propone un proyecto basado en el lenguaje natural controlado (CNL) y dirigido al LEGO Mindstorms EV3, con la intención de ayudar tanto a estudiantes de secundaria como a estudiantes que están tomando su decisión por una carrera universitaria, para poder lograr una motivación y que a su vez puedan adquirir destrezas que les ayuden a diseñar robots con menos esfuerzo. El código que realizan es generado en Microsoft Small Basic ya que es un software que se conecta fácilmente con el Brick de LEGO EV3 [1] [7].

Una publicación realizada en la Universidad de Ciencias Aplicadas en Aschaffenburg refiere que, en las clases de ingeniería electrónica, la informática no es una de sus favoritas por lo que la mayoría de los estudiantes tienen muchos problemas con la programación, por la complejidad y el pensamiento abstracto [2].

El objetivo principal es cerrar de alguna manera la brecha existente en las habilidades faltantes, para ello realizaron diferentes eventos de robots y talleres prácticos en su construcción, en donde al final del taller se evaluaron los objetivos que tenían inicialmente [8].

Las prácticas de robótica realizadas por los estudiantes de Mecatrónica, así como los de Ingeniería Eléctrica, Electrónica e Informática se rigen en el lenguaje de programación ANSI, el cual tenían propuesto en el plan de estudio e integraron al LEGO Mindstorms [9]. Además, se optó por la aplicación del aula invertida, la cual está logrando un cambio importante en lo que se refiere a sistemas de control. El aula invertida hace referencia a la creación de materiales que necesiten para aprender los manejos correctos de los sistemas de control fuera del aula [10].

Las plataformas de robots móviles están teniendo un papel sumamente importante en la educación y entre los que se puede encontrar se tiene los robots LEGO con MATLAB/Simulink, siendo utilizado principalmente en las universidades para mejorar la experiencia docente [11]. El kit de LEGO Mindstorms ha sido muy popular en la educación informática durante un tiempo, pudiendo ser configurable y soportando todo tipo de lenguajes de programación. En la versión reciente, llamada EV3, se pueden ejecutar sistemas operativos Linux personalizados en los que se destaca Python que se usa para escribir programas mediante el desarrollo de código el cual permite controlar el robot [12].

Dentro de otros artículos, los proyectos planteados están enfocados en realizar un estudio con las plataformas de Scratch, Open Roberta y LabVIEW para controlar el LEGO Mindstorms Ev3, a su vez observan el efecto que produce en los estudiantes a nivel académico, pero sin duda las plataformas que más se utilizan en el aprendizaje de la robótica y control son LEGO Mindstorms y Arduino haciendo énfasis en que la plataforma de LEGO es más fácil de programar, e inclusive realizar su montaje mecánico [13] [14].

III. MARCO TEÓRICO

A. LEGO Mindstorms EV3

El Kit de LEGO Mindstorms es una línea de robots fabricado para niños y jóvenes producidos por la empresa LEGO, brindando los elementos básicos de la teoría de los robots, como la conexión de partes y la programación de la CPU, lo que permite que varios elementos como sensores y

actuadores interactúen entre sí para lograr hacer movimientos en los robots [15].

El kit básico LEGO Mindstorms se comercializa como una herramienta educativa, originalmente desarrollada para una colaboración entre LEGO y MIT (Massachusetts Institute of Technology). Este kit puede ser utilizado tanto en el hogar como en la institución educativa, contiene 550 piezas en su set básico (Fig. 1) y 857 piezas adicionales en el set de expansión, siendo muy útiles para la creatividad al momento de construir, programar y realizar las pruebas respectivas de la tecnología robótica en la vida real [16].

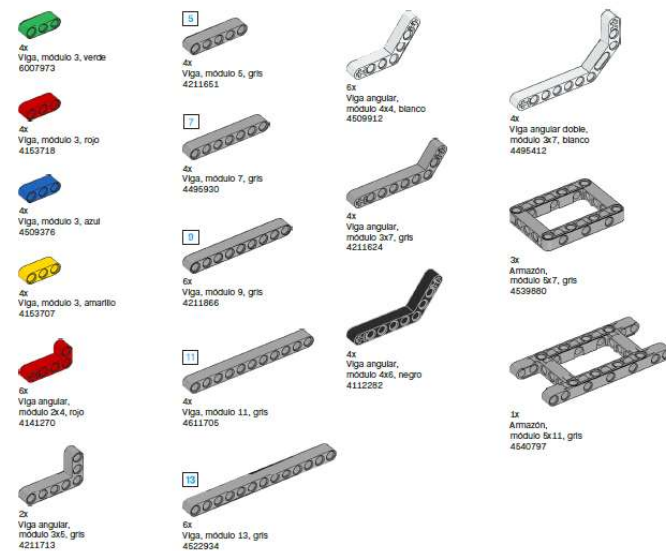


Fig. 1. Set de piezas del LEGO Mindstorms EV3 [17].

De igual manera el kit contiene su propio CPU o brick EV3, un pequeño pero potente ordenador que hace posible el control de servomotores y captar la información de los sensores que ofrece el kit, (Fig. 2). También permite la comunicación inalámbrica como Bluetooth y Wifi o mediante cable tipo USB a micro USB, pudiendo realizar mediante estos la programación y registro de datos [16].

Dentro del kit de LEGO Mindstorms EV3 básico de encuentra lo siguiente:

- Software educativo LEGO Mindstorms EV3 en castellano.
- Batería recargable.
- Cargador de batería.
- Sensor giroscopio.
- Sensor preciso de ultrasonidos.
- Sensor de colores
- Sensor touch
- Servomotores
- Piezas especiales.
- Cajas compartimentadas de plástico, etc.

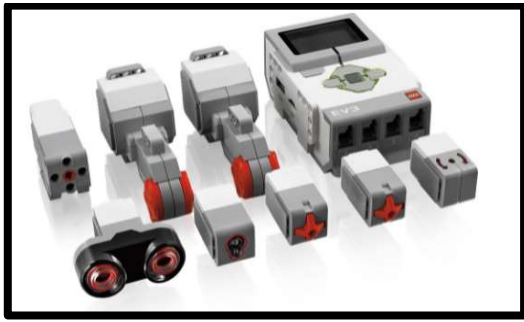


Fig. 2. Sensores, actuadores y CPU del LEGO Mindstorms EV3 [17].



Fig. 3. Kit completo del LEGO Mindstorms EV3 [17].

B. MATLAB / Simulink

MATLAB es una potente herramienta con la que se puede contar al momento de programar el LEGO Mindstorms EV3. El uso de esta herramienta es muy importante en diferentes áreas del desarrollo de la ingeniería, en este caso, dirigido a la robótica y control, por tal motivo, se ocupa una herramienta muy especial de MATLAB que es Simulink, la misma que ayuda a modelar el comportamiento de sistemas dinámicos. Puede modelar sistemas lineales y no lineales, modelos de tiempo continuo y discreto y sistemas que combinen todo lo anterior. Se trata de un entorno gráfico donde se construye un modelo de simulación haciendo clic y arrastrando los distintos bloques que lo componen. Los modelos Simulink se guardan en archivos *.mdl. En las nuevas versiones, Simulink ha ampliado las bibliotecas de bloques (block sets) y sus capacidades, encontrando así librería para una interfaz de programación para LEGO Mindstorms EV3 (Fig. 4) [18].

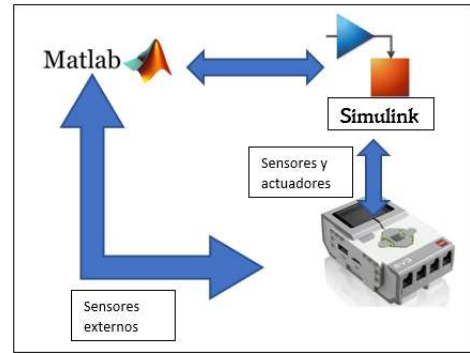


Fig. 4. Conexión Simulink-EV3.

C. Python

La robótica dentro de la educación tanto en niños, jóvenes y adultos es relevante en los últimos tiempos, fomentados una motivación en los estudiantes y permitiendo que los niños miren de cerca a la tecnología, ingeniería y matemática.

Lo que se pretende con Python los alumnos aprenden jugando a realizar acciones complejas que por lo general son difíciles de explicar o en este caso entender mediante una clase.

Para poder utilizar el software de Python se necesitará instalar el programa a un micro SD que se insertará en el Brick o procesador del robot, donde se ejecutarán las líneas de código realizadas previamente en un ordenador[20].



Fig. 5. Conexión Python-EV3.

D. Arduino

Este artículo presenta una iniciativa para mejorar la enseñanza de programación en ciencias mediante la integración de plataformas educativas como Arduino y LEGO. El enfoque interdisciplinario y práctico de este método tiene como objetivo aumentar la participación y comprensión de los estudiantes en el proceso de aprendizaje [21].

Como es de conocimiento, para programar una placa Arduino se utiliza su propia IDE, que es la manera más común de realizar los códigos y se puede obtener su software gratuito desde su página oficial. La plataforma de Arduino tiene una diversidad de compatibilidades con sistemas operativos, tales como Windows, Linux o MacOS, lo que la convierte en una plataforma muy versátil para el manejo de programación y control del LEGO Mindstorms EV3. En la

plataforma de Arduino existe una variedad de modelos de códigos de uso, es por eso que al momento de poder elegir uno de los tantos modelos, se hace complicado por lo que hay que tomar cuenta es la cantidad de entradas y salidas que posee el modelo [20].

Por otro lado, se hace referencia al uso del Arduino Mega puesto que es suficiente para el control del número de servos y sensores que se requiere, los sensores que trae consigo el Mindstorm EV3 son infrarrojos, de contacto, de ultrasonido y de luz, en definitiva, se utilizara un total de tres servos y sus complementos que ayudarán con la información visual y acústica, como son: LEDS y zumbadores.

El cerebro, a diferencia de los dos proyectos propuestos, estará a cargo de una placa Arduino.

IV. METODOLOGÍA

Dentro de este apartado se procederá a realizar la simulación de tres procesos industriales utilizando el kit LEGO Mindstorms EV3, cada uno de ellos con sus respectivos software de programación:

- Seleccionador de colores programado con Simulink
- Brazo robótico programado con Python
- Robot transportador programado con Arduino

Para el primer proceso de simulación se construyó una banda transportadora utilizándose como seleccionador de colores. Para su programación se utilizó el software de MATLAB con su potente herramienta Simulink en donde fue necesario aplicar máquinas de estado para poder seguir una lógica de proceso.

A. Banda Transportadora como seleccionador de colores utilizando Simulink

Los pasos para la construcción de este robot se pueden encontrar dentro de la página oficial de LEGO, fueron necesarias un total de 413 piezas del kit incluyendo sensores y motores, para esto se realizaron tres etapas diferentes:

- Etapa construcción.
- Etapa programación.
- Etapa comunicación.

1) Etapa de construcción

La etapa de construcción está compuesta de tres partes:

- Construcción de la base para el brick (CPU EV3): para ello se utilizan 64 piezas las cuales son usadas para fijar el brick del EV3 y como soporte para la base de la banda (Fig. 6).

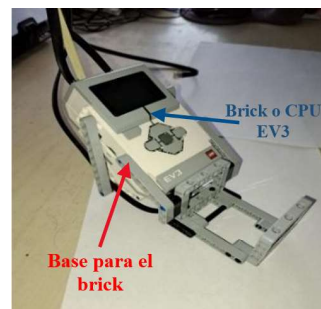


Fig. 6. Brick montado en la base.

- Construcción de la base para la banda transportadora: se utilizan 113 piezas incluido el motor grande el cual permite el movimiento de la banda que a su vez traslada de izquierda a derecha la base de los ladrillos de colores. (Fig. 7).

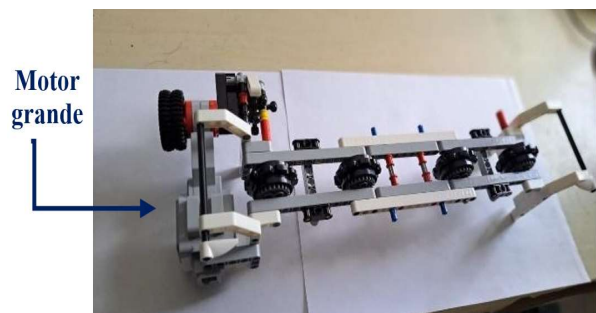


Fig. 7. Base para la banda.

- Construcción de la base para los ladrillos de colores: se utilizan 236 piezas dentro de la misma se tiene el motor pequeño el cual es utilizado para expulsar los ladrillos en su lugar, también, consta del sensor de colores el cual permite identificar y dar la señal a donde dirigirse la banda. (Fig. 8).

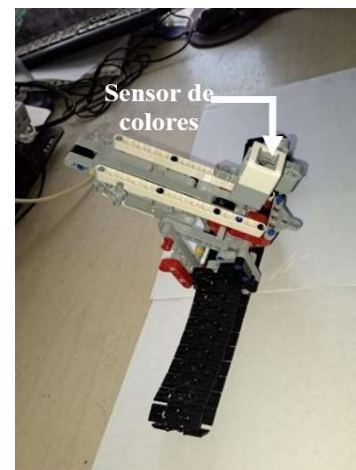


Fig. 8. Base para los ladrillos de colores.

Para la parte final se realizó el armado completo del seleccionador de colores con un total 413 piezas haciendo uso de dos sensores y dos motores y también con los ladrillos de colores que se procederá a seleccionar al ejecutar el programa. (Fig. 9).

Sensores

- Sensor de colores: puede reconocer siete colores diferentes dando un valor numérico
- Sensor touch: reconoce tres estados; sin presionar, presionado o tipo pulsos

Motores

- Motor grande: para mover la banda
- Motor pequeño: para expulsar el ladrillo de colores

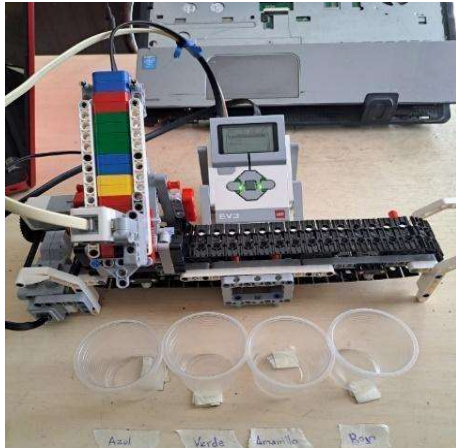


Fig. 9. Seleccionador de colores completo.

2) Etapa de programación

Dentro de esta etapa, para realizar la programación de este robot, se utilizó la plataforma de MATLAB con su herramienta Simulink la misma que aporta con una sub-herramienta denominada StateFlow en la cual se realiza la programación utilizando máquinas de estado.

Las librerías utilizadas para esta programación se obtienen de “Add-Ons” (Fig. 10a) herramienta que se encuentra en la ventana principal de MATLAB o Simulink, dentro de ésta herramienta se encontrarán muchos paquetes de soporte de hardware, entre ellos el que se utiliza *Simulink Support Package for LEGO MINDSTORMS EV3 Hardware*. Ver (Fig. 10b)

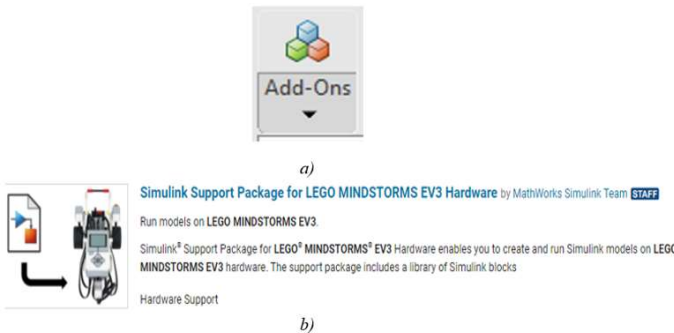


Fig. 10. a) Herramienta "Add-Ons" donde estará Get Support Packages Hardware. b) Paquete de soporte Simulink para el Hardware LEGO MINDSTORMS EV3.

Una vez descargada la librería se procede a abrir Simulink donde arrastrando los respectivos bloques se irá realizando la programación.

Para empezar, se encontrará y se llevará a la ventana de trabajo los bloques de sensores, botones y la pantalla LCD donde se mostraran ciertos valores como posición y valor numérico del color (Fig. 11), estos bloques se ubicarán dentro de un subsistema.

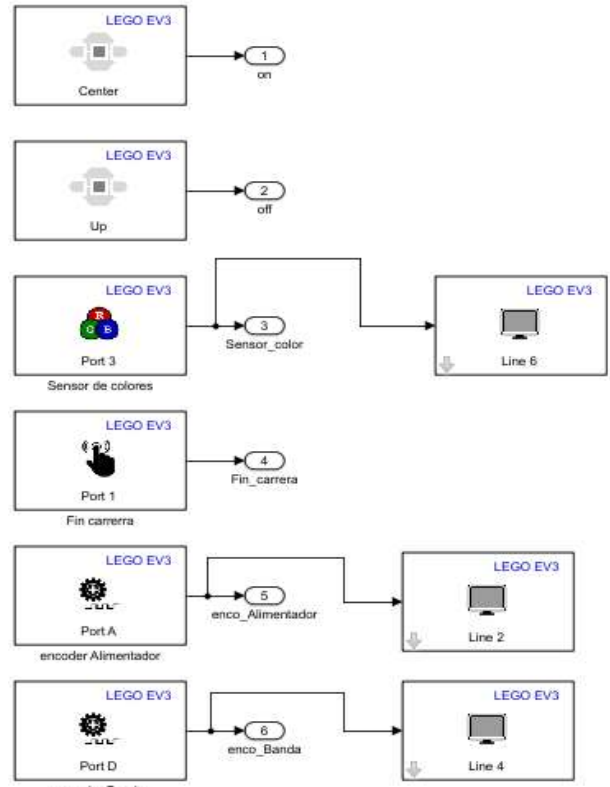


Fig. 11. Bloque de los sensores, botones y de la pantalla LCD.

De la misma manera se procede a ubicar bloques para las constantes dentro de un subsistema, el cual permitirá dar valores de velocidad y posición para los motores (Fig. 12), estos se encontrarán dentro de un solo grupo. (Fig. 13). Una vez realizado todos los subsistemas tendremos el siguiente esquema (Fig. 14).

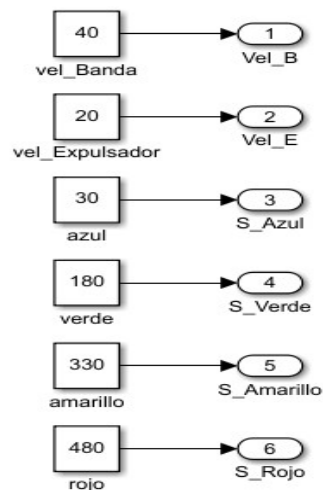


Fig. 12. Bloques de los valores constantes para velocidad y posición de los motores.

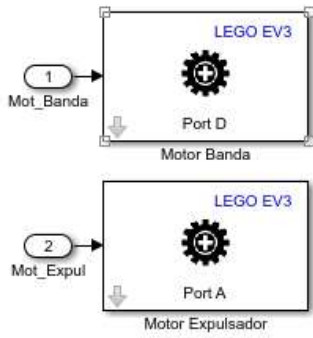


Fig. 13. Bloque de los dos motores.

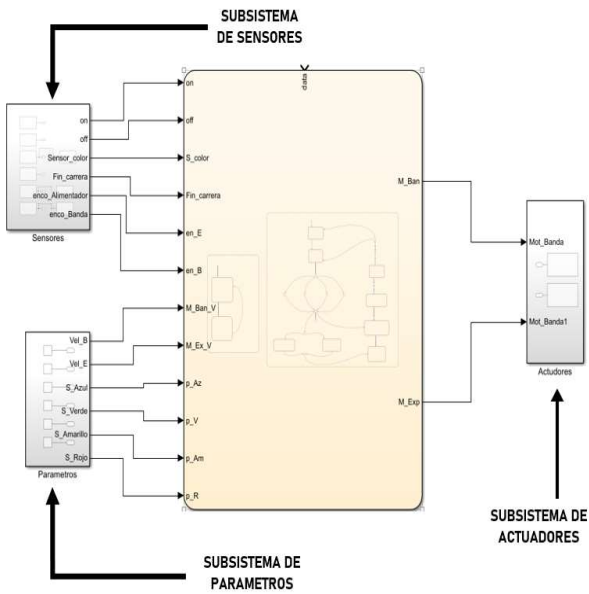


Fig. 14. Subsistemas de sensores, parámetro y actuadores.

Una vez que se tiene todos los bloques dentro de los respectivos subsistemas se busca un bloque llamado "Chart" (Fig. 15) que se encuentra en la librería StateFlow, la misma que está dentro de las librerías de Simulink, el cual es utilizado para la programación por máquinas de estado donde se da las respectivas entradas y salidas dependiendo de los sensores, parámetros y actuadores utilizados, también se utilizan variables auxiliares y los cambios de estado que puedan existir en los botones.(Fig. 16).

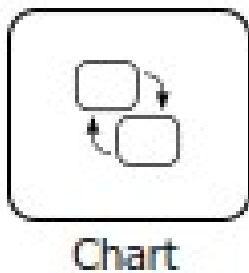


Fig. 15. Bloque "Chart" (sacado de Simulink).

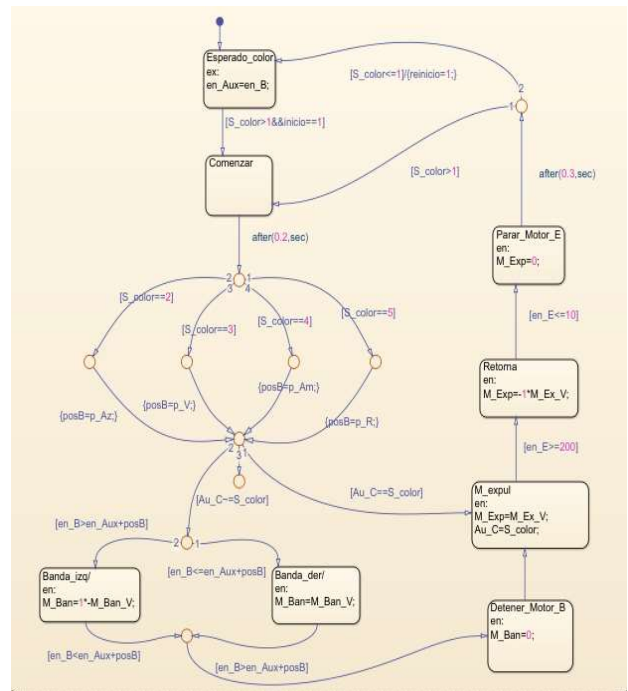


Fig. 16. Programación mediante máquinas de estado para el seleccionador de colores.

3) Etapa de comunicación

Para poder transferir el programado desde Simulink al dispositivo Brick EV3 y aprovechando sus características se optó por utilizar una comunicación inalámbrica mediante Bluetooth, la misma que permite una transferencia de datos evitando la conexión de cables desde el ordenador hacia el dispositivo Brick EV3, cabe recalcar que el ordenador debe poseer conexión Bluetooth.

Para la configuración de esta comunicación se la realizará dando clic en "Model configuration parameters" que se encuentra en la barra de herramientas "MODELING". (Fig. 17)

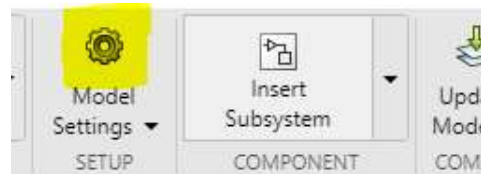


Fig. 17. Herramienta para la configuración de parámetros de comunicación.

A continuación, se indica la siguiente ventana y se dará clic en "Hardware Implementation", luego se tiene que dirigir a "Host to Target Connection" y en la opción "Connection Type" se escogerá "Bluetooth" y finalmente se dará clic en "aceptar" terminando así la configuración de la comunicación (Fig. 18).

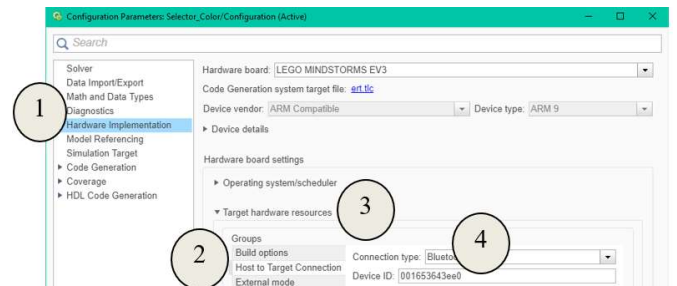


Fig. 18. Ventana para la configuración de la comunicación Bluetooth.

B. Control de Brazo Robótico utilizando Python

El brazo robótico está compuesto por 3 motores que permiten el movimiento de las diferentes partes del brazo. Estos motores se pueden programar para mover el brazo hacia arriba y hacia abajo, girar en diferentes direcciones y abrir o cerrar la pinza. Además, el brazo robótico también está equipado con sensores, de ultrasonido, sensores de color y sensor touch, que pueden ayudar al robot a interactuar con su entorno.

De igual manera la construcción del brazo robótico se lo puede encontrar dentro de la página oficial de LEGO, fueron necesarias un total de 282 piezas del kit incluyendo sensores y motores, para esto se realizó tres etapas diferentes:

- Etapa construcción
- Etapa programación
- Etapa comunicación

1) Etapa de construcción

La etapa de construcción está compuesta de tres partes donde cada una de ella cumple su función.

- Primera parte. - se realiza la construcción de la base que sostiene y permite el giro del brazo de izquierda a derecha (Fig. 19a)., para ello se utiliza un motor grande y un sensor touch que trabaja como fin de carrera para iniciar el movimiento del robot (Fig. 19b).

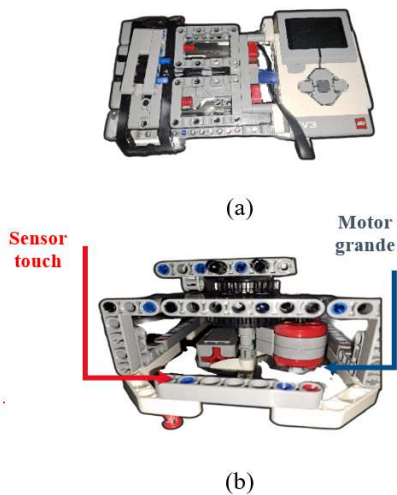


Fig. 19. (a) Base del brazo. (b) Base del brazo identificando sensor touch y motor grande.

- Segunda parte. – se construye el brazo, antebrazo y la articulación que permite el movimiento de arriba hacia abajo (Fig. 20), para esto también se utilizara un motor grande y el sensor de colores el cual actuará como fin carrera para que no permita que la articulación suba demasiado y evitar algún daño.

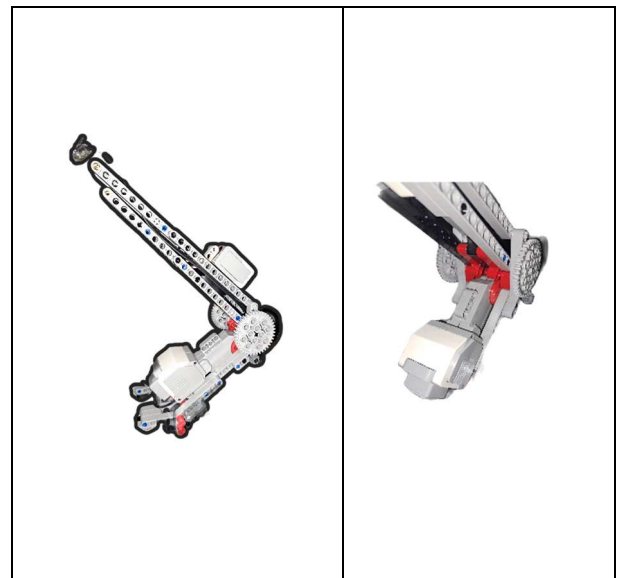


Fig. 20. Brazo robótico con motor grande para el movimiento de la articulación del brazo.

- Tercera parte. – se construye la pinza (Fig. 21), ésta permite sujetar un objeto para poder mover o transportarlo a otro lugar, para poder abrir y cerrar la pinza se utiliza el motor pequeño, el cual, mediante la programación puede mandar una señal que simula la presión que ejerce la pinza y así evitar daños al motor.

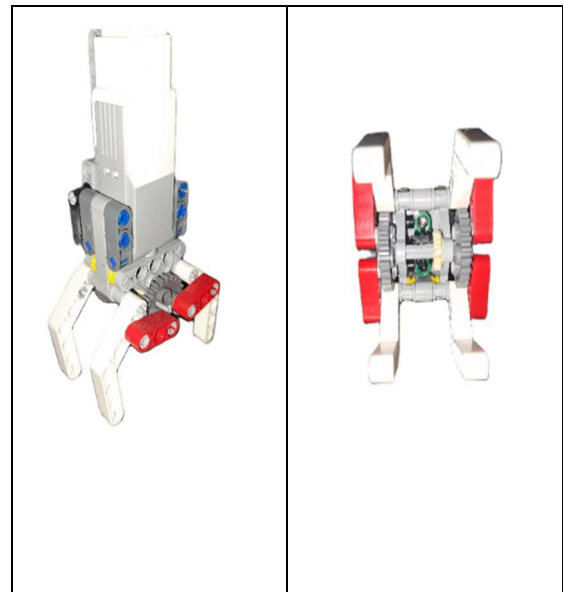


Fig. 21. Pinza con motor pequeño y engranaje conectados al motor para abrir y cerrar pinza.

Finalmente se realizó el armado completo del brazo robótico (Fig. 22), uniendo las diferentes partes haciendo uso de los sensor y actuadores que ya se mencionaron anteriormente, para este proyecto también se consideró usar el sensor de ultrasonido para que se pueda detectar un objeto y así el robot pueda iniciar su proceso.

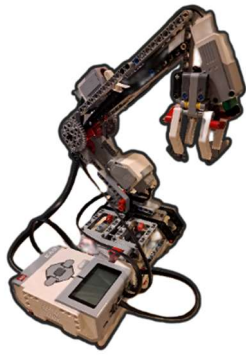


Fig. 22. Brazo armado completamente.

2) Etapa de programación

Después de construir el brazo robótico, se debe programar para que pueda realizar una variedad de movimientos y funciones. Esto se puede hacer usando Python y el entorno de programación de LEGO Mindstorms.

Para ello se ocupa una imagen o firmware de MicroPython, una implementación optimizada de Python 3 diseñada específicamente para sistemas embebidos y microcontroladores con recursos limitados. MicroPython es un lenguaje de programación de alto nivel que facilita la creación de programas para controlar dispositivos físicos y llevar a cabo tareas específicas en sistemas embebidos y microcontroladores

Para iniciar la programación del proceso industrial del brazo robótico en el lenguaje de programación Python, fue necesario descargar el software VSC (Visual Studio Code), (Fig. 23) por ser éste compatible con múltiples lenguajes de programación tales como C++, C, Visual Basic, etc.



Fig. 23. Visual Studio Code.

Este software proporciona la extensión necesaria, (Fig. 24) para poder realizar la programación y comunicación entre brick EV3 y el ordenador.

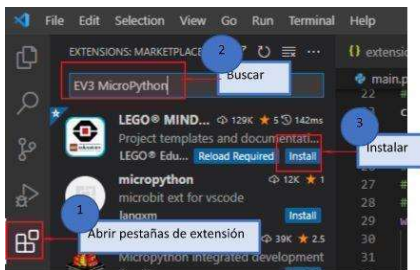


Fig. 24. Pasos para descargar la extensión.

Una vez instalada la extensión se procede a realizar el código en un nuevo archivo. Para poder iniciar es necesario importar las librerías y crear las respectivas variables.

Para el código tomaremos en cuenta el siguiente diagrama de flujo (Fig. 25).

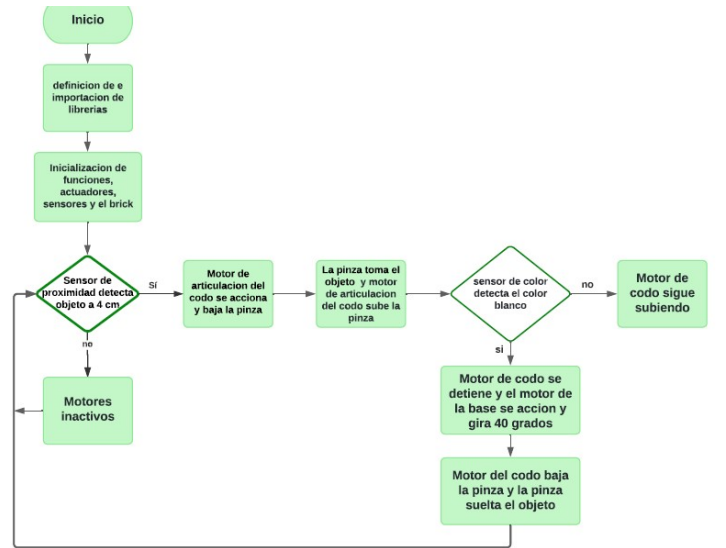


Fig. 25. Diagrama de flujo para el brazo robótico

3) Etapa de comunicación

La imagen o firmware utilizada para la comunicación, es descargada y flasheada dentro de una microSD utilizando, en este caso, "Etcher". Asegurándose de que el archivo de firmware tenga el nombre correcto (generalmente, "LEGO Ev3 v2.0.0.bin"), ver Fig. 26.

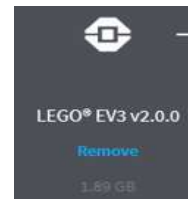


Fig. 26. Firmware a instalar en la microSD.

Inserta la tarjeta microSD en el slot para tarjetas SD del brick de LEGO Mindstorms, ver Fig. 27.



Fig. 27. Insertar microSD.

Enciende el LEGO Mindstorms y espera a que cargue el firmware de Micro Python desde la tarjeta microSD, como se observa en la Fig. 28.



Fig. 28. Cargando firmware en brick EV3.

Para la comunicación entre el ordenador y el brick EV3 se utiliza un cable USB a mini USB. Para la ejecución del programa realizado se debe tomar en cuenta que el dispositivo EV3 sea reconocido por VSC como se muestra a continuación en la Fig. 29.

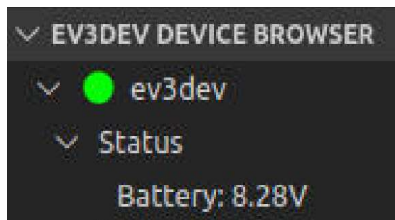


Fig. 29. Estatus activo del brick EV3.

Para finalizar con la simulación del proceso del brazo robótico se procede a ejecutar el programa.

A. Carro Transportador utilizando el sistema embebido de Arduino

El carro transportador es un robot móvil que está compuesto de dos motores, los cuales permiten el movimiento hacia adelante, atrás, izquierda y derecha todo esto mediante los sensores (encoder) que llevan incorporados. Además, este robot hace uso del sensor de colores, sensor touch y pull down.

1) Etapa de construcción

Los pasos de la construcción de este robot se pueden encontrar en la página oficial de LEGO. La construcción del robot es muy sencilla en especial porque sólo es necesaria la base del brick, (Fig. 30) ya que el mismo será reemplazado por una placa Arduino MEGA. Se utilizaron un total de 58 piezas incluyendo motores y sensores.

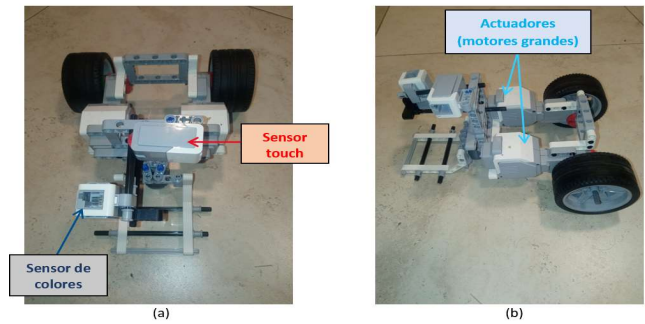


Fig. 30. Base del brick.

2) Etapa de comunicación

Después de construir parte del robot móvil, se debe realizar esquemas para la conexión de los actuadores con la placa Arduino MEGA y el controlador L298N y de la misma manera con el sensor de colores, el sensor touch y unas placas con resistencias en pull down.

El motor del kit LEGO es un motor DC por lo tanto se puede controlar la velocidad y sentido de giro, esto se podrá hacer mediante las entradas de PWM_1 y PWM_2. Para obtener información o señales de los encoder, se alimentará el circuito interno en los pines 3 (GND) y pin 4 ($V_{in} = 5V$) y por último las salidas de los encoder están en el pin 5 (Encoder_A) y pin 6 (Encoder_B), (Fig. 31).

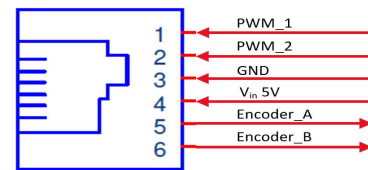


Fig. 31. Esquema de conexiones para el actuador o motor.

De igual manera para los sensores se tendrá la siguiente información de las entradas y salidas para el control de los dispositivos, ver (Fig. 32). Este sensor al ser digital y para la comunicación con la placa Arduino se utiliza con un esquema de pull up en el pin 1 y pull down en el pin 6, (Fig. 33).

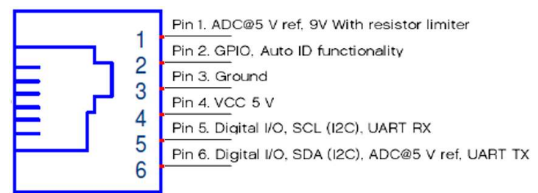


Fig. 32. Esquema de conexiones para los sensores.

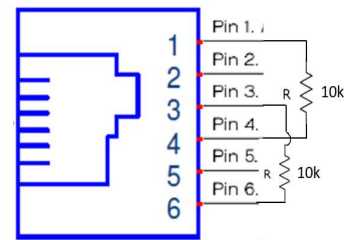


Fig. 33. Esquema de conexiones del sensor de colores con la placa Arduino.

Con la información obtenida se podrá realizar un esquema de conexión para la comunicación del sensor de colores y de los actuadores (dos motores) con la placa Arduino MEGA.

Para la conexión del sensor se tiene el siguiente esquema: (Fig. 34).

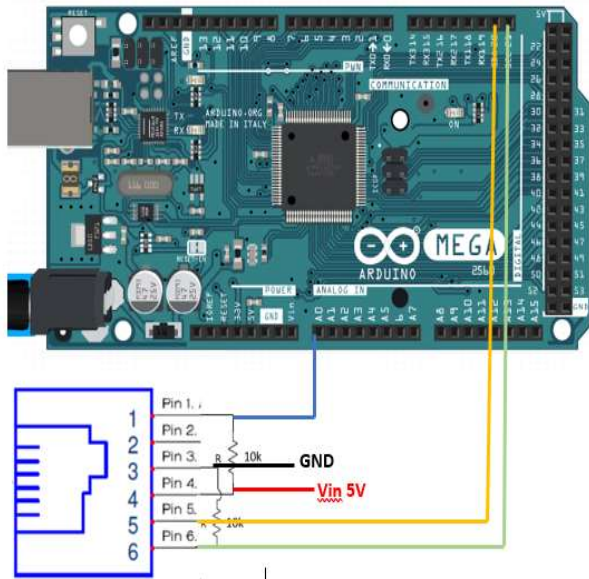


Fig. 34. Esquema de conexiones para el sensor de colores con la placa Arduino MEGA.

De la misma manera el esquema de conexión para el actuador se puede ver en la (Fig. 35).

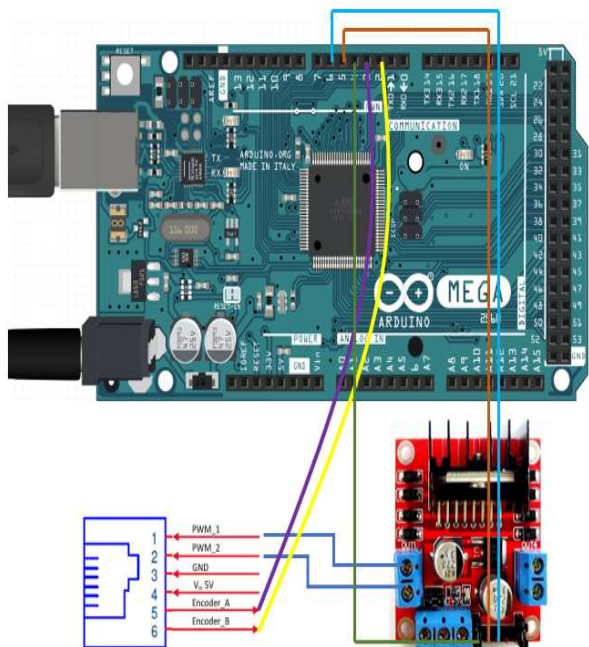


Fig. 35. Esquema de conexiones con un motor.

Se utilizarán resistencias de pull down las cuales permitirán elegir el color a transportar el esquema lo podremos encontrar en la (Fig. 36).

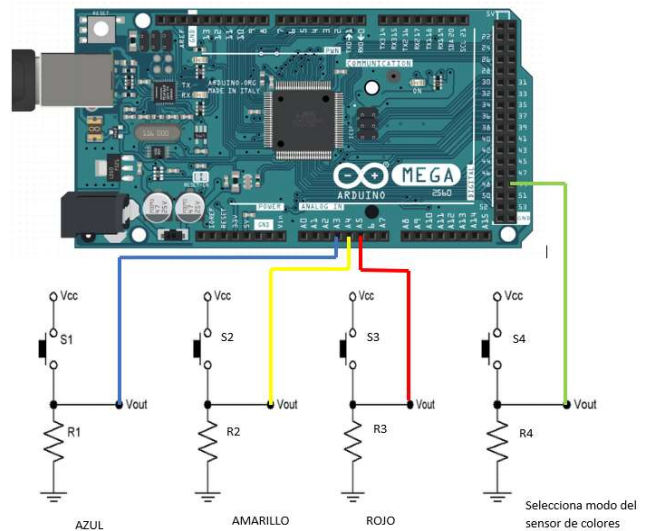


Fig. 36. Esquema de conexiones resistencias pull down.

El esquema para la conexión del sensor touch al ser un sensor analógico su salida se conecta directamente a un pin de la placa Arduino. (Fig. 37)

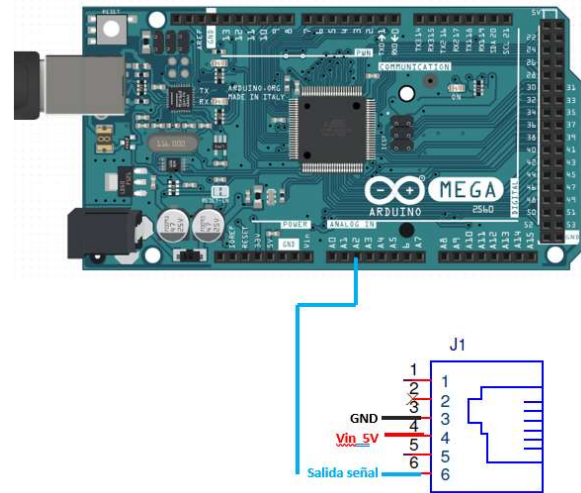


Fig. 37. Esquema de conexiones del sensor touch

El conector que se utiliza para los datos y alimentación, tanto de actuadores y sensores, es un RJ12 con una característica especial, su pestaña de seguridad se encuentra al lado derecho. Este tipo de conector no es común, es por este motivo que se optó por usar un conector RJ12 normal y adecuarlo para que se ajuste a la entrada del dispositivo, como se ve en la (Fig. 38), otro motivo por la cual se tomó esa decisión es que al otro extremo al ser imposible encontrar el conector hembra del RJ12 se necesita separar los hilos del cable UTP para las respectivas conexiones.

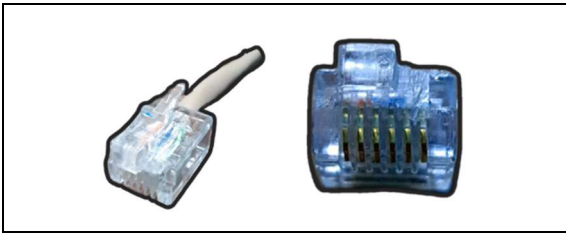


Fig. 38. Conector RJ12 adecuado para ajustarse a la entrada del hardware.

Ahora que se tiene la información completa y los esquemas de las respectivas conexiones, se procede a realizar las conexiones, para ello se fabrican las placas con los materiales necesarios para que se pueda realizar la comunicación, como se ve en la Fig. 39.

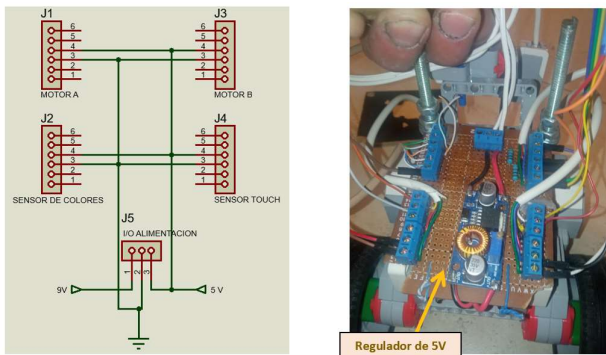


Fig. 39. Conector RJ12 adecuado para ajustarse a la entrada del hardware.

Ya con el material listo se terminará de construir el robot móvil como se ve en la Fig. 40 y de esta manera poder realizar la programación.

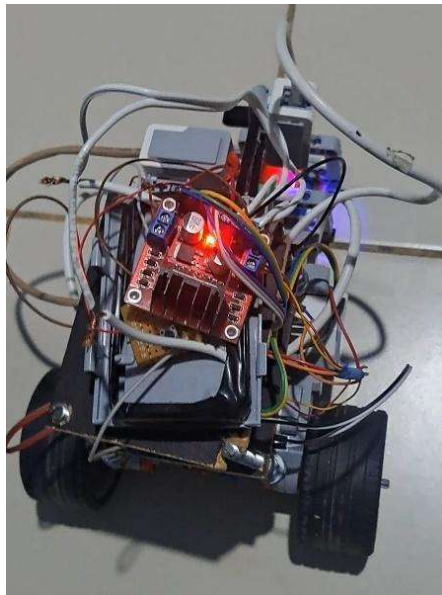


Fig. 40. Robot móvil armado.

3) Etapa de programación

Para realizar la programación del robot se utilizará un diagrama de bloques el cual ayuda a tener una idea de los pasos a seguir para implementar el código en Arduino, Fig. 41.

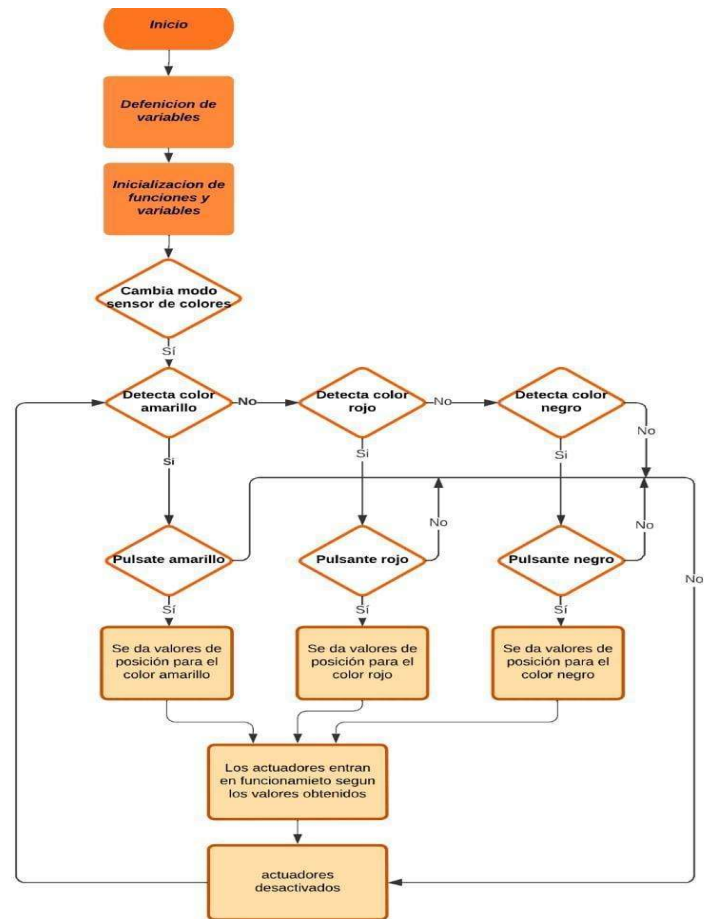


Fig. 41. Creación de variables e inclusión de librerías.

Para esta etapa se necesitará tener presente los pines de la placa Arduino MEGA que fueron utilizados en la conexión de los diferentes dispositivos utilizados. Por lo tanto, para iniciar, primeramente, se realizará un llamado a las respectivas librerías a utilizar y de igual manera se definen las variables necesarias.

En la función "void loop()", es el bucle donde se encuentra el código el cual permite la comunicación y lectura entre la tarjeta Arduino y el sensor de colores, enviando y recibiendo datos para una constante conexión de su comunicación.

Se crean funciones, las cuales ayuden a leer los datos del sensor y de esta manera se pueda realizar la detección de colores, los cuales permitirán el funcionamiento de los actuadores que, dependiendo del color que se esté reconociendo el robot, se dirige a una posición ingresada en el código.

La posición que recorra el robot depende del ángulo o posición del encoder y con una librería que ofrece Arduino se pueden leer los datos del sensor interno de estos actuadores, de esta manera el robot podrá dirigirse a cierta posición y regresar a la posición de inicio.

V. PRUEBAS Y RESULTADOS

En este apartado, se detallan las pruebas realizadas para evaluar el sistema implementado en el proyecto, donde se combina el uso de LEGO Mindstorms EV3 utilizando los entornos de programación de Simulink, Python y Arduino.

Estas pruebas fueron diseñadas para validar la funcionalidad y la eficiente interacción entre las diferentes tecnologías empleadas

A. Pruebas y resultados obtenidos en la simulación del selector de colores utilizando el software de programación gráfica Simulink

1) Prueba de Conexión

Durante las pruebas experimentales, se enfocó en verificar la capacidad de Simulink para establecer la conexión efectiva con el brick EV3 después de la instalación de sus paquetes de soporte. La comunicación esencial para el proceso del selector de colores se realiza mediante un enlace inalámbrico conocido como Bluetooth. Se realizaron las pruebas de conexión a diferentes distancias obteniendo como resultado que la comunicación no puede exceder una distancia de emisión - recepción de 5 metros, como se observa en la Fig. 42, concluyendo que la compilación del proyecto es exitosa dentro del rango propuesto.



Fig. 42. Distancia de comunicación inalámbrica entre 0-5 metros.

En las pruebas se observó que al sobrepasar una distancia mayor 5 metros la compilación no es exitosa presentando mensajes de error, como se ve en la Fig. 43.

```
### Downloading the application to EV3 ...
### Build procedure for Selector_Color aborted due to an error.

The following error occurred during deployment to your hardware board:
Failed to download application to EV3: Cannot connect to EV3 brick through Bluetooth.
Make sure:
1) EV3 is connected properly through Bluetooth and the Device ID entered in the Configuration
parameters matches the device ID of the Brick.
2) No connection to the EV3 brick is established through EV3I/O. Clear any existing EV3I/O object.
```

Fig. 43. Mensaje de error al no poder enlazarse con el dispositivo.

2) Prueba de Secuencias de Movimiento:

En la simulación del proceso, al realizar las primeras pruebas de control básico, se observó que los datos del sensor fueron correctos dando los valores numéricos respectivos de cada color, como se muestra en la Fig. 44.

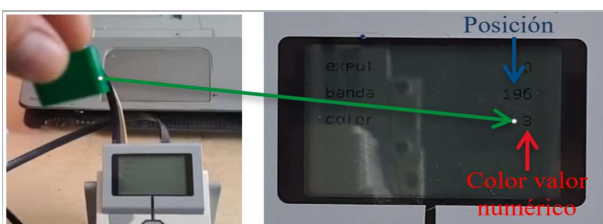


Fig. 44. Prueba sensores y actuadores.

Para la realización de la prueba de secuencia de movimientos con el seleccionador de colores se necesitó encontrar el límite de desplazamiento de la banda, como se muestra en la Fig. 45, concluyendo que el límite de la banda entre el punto inicial y el punto final es de 22 cm con un error de $\pm 2\%$, este límite debe ser exacto para proteger los respectivos actuadores de una posible avería.

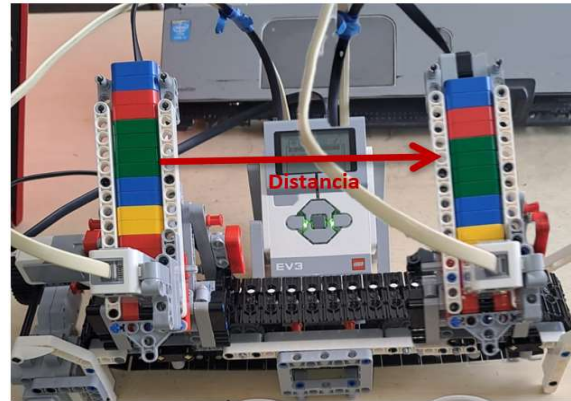


Fig. 45. Desplazamiento de la base de los ladrillos de colores.

En las primeras pruebas de selección de colores se observó que el proceso se realizó con un 80% de éxito, esto se debe a que el sensor de colores tenía errores al momento de realizar el respectivo reconocimiento. En esta prueba se pudo constatar que el error se dio por la posición del sensor, y se concluyó que el sensor, al no estar dirigido correctamente y a una cierta distancia del color a reconocer, puede dar valores no acordes a los reales (ejemplo el color verde fue reconocido como azul o amarillo). Este error pudo ser solucionado moviendo el sensor a una posición adecuada, (Fig. 46), seguidamente las pruebas se volvieron a realizar y esta vez la selección fue realizada con un 100% de éxito.



Fig. 46. Sensor colocado en una posición adecuada para el reconocimiento del color.

B. Pruebas y resultados obtenidos en la simulación del proceso del brazo robótico utilizando el software de programación Python

1) Configuración Inicial y Conexión

La plataforma de programación Python es la encargada en organizar todas las secuencias para el correcto funcionamiento del brazo robótico, para las pruebas de comunicación fue necesario utilizar un cable de datos USB a mini USB, como se ilustra en la Fig. 47 (a). Para verificar la correcta conexión se procedió con la primera prueba que

consistió en examinar la interfaz de Virtual Studio Code, asegurándose que el indicador visual de conexión se observe en un tono verde. Esta prueba indica que el dispositivo está listo para poder realizar la ejecución del programa indicando de este modo una vinculación exitosa del dispositivo, tal como se evidencia en la Figura 47 (b).



(a)



(b)

Fig. 47. (a) Conexión, (b) Estado conectado.

Durante las pruebas iniciales de la simulación del proceso del brazo robótico, se detectó un inconveniente con el brick EV3 de LEGO, este problema se manifestó al realizar la prueba de compilación del programa, la misma que no pudo ser ejecutada con éxito, concluyendo que el problema se suscitó cuando los sensores o actuadores no son debidamente conectados, generando mensajes de error como se observa en la Fig. 48. Al encontrar este problema se puede hacer énfasis en la importancia de una configuración y conexión precisa de los componentes físicos del robot para garantizar una ejecución sin contratiempos.



```
A sensor or motor is not connected to the specified port:
--> Check the cables to each motor and sensor.
--> Check the port settings in your script.
--> Check the line in your script that matches
    the line number given in the 'Traceback' above.
```

Fig. 48. Sensor o actuador desconectado.

En la prueba de grabación o flasheo del firmware, en primera instancia se realizó el grabado en una tarjeta microSD genérica, donde no se pudo realizar de manera correcta la carga del firmware, concluyendo que para poder realizar la grabación o flasheo se necesita la autenticidad de la tarjeta microSD.

Con respecto a la simulación del proceso del brazo robótico, se realizó una prueba con dos tipos de objetos, cilíndricos y prismáticos, empleando el sensor de ultrasonido para la detección de dichos objetos. Sin embargo, durante las primeras pruebas, se evidenció un 50% de éxito en la detección de objetos cilíndricos y un 80% en la detección de los prismáticos, provocando que el proceso no pueda ejecutar las acciones asignadas. Este sensor enfrentaba desafíos al detectar objetos cilíndricos que no estuvieran posicionados de manera precisa. Para solventar este problema, se optó por una estrategia que consistió en utilizar una señal para ubicar el objeto a detectar en la posición correcta, donde se realizó pruebas con diferentes distancias concluyendo que el objeto debe encontrarse a 8cm del sensor de ultrasonido y a 4 cm de la base del brazo robótico, tal como se detalla en la (Fig. 49). Este enfoque fue una solución efectiva para superar los inconvenientes asociados con la detección de objetos cilíndricos, procediendo así a realizar nuevamente las pruebas de funcionamiento logrando un éxito del 100% en la detección.

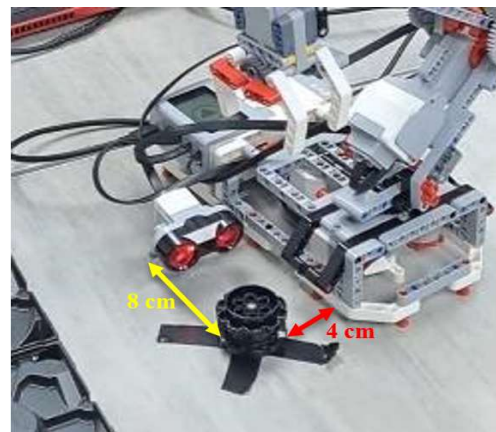


Fig. 49. Objeto colocado en el lugar para ser detectado.

C. Pruebas y resultados obtenidos en la simulación del robot transportador utilizando el software de programación Arduino y la tarjeta Arduino Mega

1) Configuración de Comunicación entre Arduino y EV3

En la construcción del robot transportador, para que pudiera llevar a cabo la acción de transporte o ejecutar el proceso correspondiente, se llevaron a cabo pruebas para evaluar el funcionamiento de los actuadores y sensores, realizando ajustes tanto en el código como en el hardware.

En las pruebas para la integración del sensor de colores, dentro del código se empezó a definir variables como, la transmisión de datos (baudios), variables indicadoras y variables direccionales y de manera complementaria, se

diseñó una función que hace uso de estas variables. Este enfoque fue esencial debido a que el sensor es un dispositivo digital y posee su propio protocolo de comunicación.

Estas pruebas no solo reflejan una metodología para la integración de componentes, sino también la capacidad de adaptación del código y del hardware para optimizar el rendimiento del robot transportador.

En la simulación del robot transportador, ver Fig. 50, se realizaron pruebas en las cuales se pudo observar que el sensor tiene errores al cumplir con la función de reconocimiento de colores, la cual se realizó con un 50% de éxito. Este error se reflejó aún más con el color azul, debido a que durante las pruebas el mismo no era reconocido, esto se debe a la ubicación del sensor y la luz ambiente, es por este motivo que se tomó en consideración probar diferentes puntos de reconocimiento. Con el sensor ubicado correctamente se realizaron nuevamente las pruebas respectivas y se observó que la detección de los colores tuvo un éxito de un 95%.

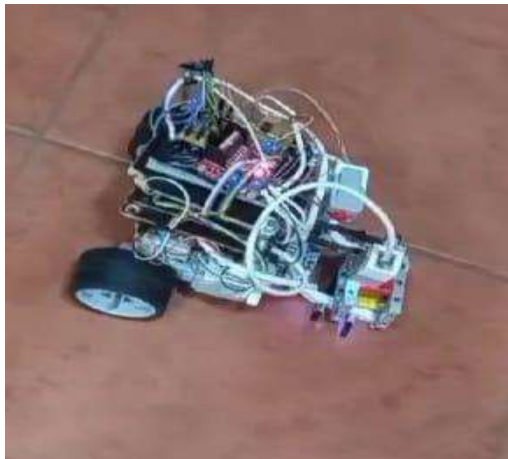


Fig. 50. Robot móvil o robot transportador.

Para las pruebas de movimiento del robot se estableció una conexión física entre el Arduino Mega, los sensores y los actuadores del EV3, como se observa en la Fig. 51. Se verificó la correcta conexión de los cables y la asignación de salidas y entradas para los dispositivos.

Para el control de los actuadores, al ser motores DC, se hizo uso del módulo L298N (Controlador de Motores) y se realizaron pruebas del control de velocidad y posición considerando la lectura de los encoder, esto hizo posible realizar las pruebas de movimientos del robot como giro a la derecha o izquierda, moverse para adelante o retroceder para transportar la carga a su respectivo lugar dependiendo del color.



Fig. 51. Conexión entre Arduino Mega, controlador L298N y placa externa.

Posteriormente, con las respectivas modificaciones en el código, se realizan las pruebas de control de movimiento obteniendo errores menores a los iniciales dando como resultado que el robot realizó sus acciones con un 90% de éxito.

VI. CONCLUSIONES

Para los diferentes procesos implementados después de realizar las respectivas modificaciones se obtuvo con éxito la ejecución de su funcionalidad, tanto en comunicación con las distintas plataformas de programación como al realizar las acciones utilizando sensores y actuadores.

En la configuración de las plataformas de programación como lo son MATLAB y Python no hubo mayor inconveniente porque se logró con éxito la instalación de paquetes de soporte y extensiones, las mismas que se utilizan al momento de realizar los respectivos códigos.

En la plataforma Arduino hubo inconvenientes porque los sensores llevan su propio protocolo de comunicación y no existen librerías para ellos. También, se procedió a crear placas externas para la funcionalidad del proceso.

Durante la validación de las prácticas se pudo observar que algunas páginas web vienen detalladas en el idioma inglés, esto conlleva un inconveniente para la mayoría de estudiantes que no están familiarizados con el idioma, generando algunas dificultades durante el proceso de instalación.

El uso del brick EV3 en sistemas robotizados ha demostrado ser un recurso altamente beneficioso. También, se destaca por su facilidad de integración y versatilidad, esto se pudo observar durante los procesos simulados del brazo robótico con Python y el seleccionador de colores con Simulink, en ambos casos haciendo uso del EV3, esto ha facilitado una interfaz intuitiva y accesible para la programación y control, permitiendo una rápida identificación y corrección de posibles inconvenientes. Además, la capacidad de adaptación del EV3 ha sido evidente al ser reemplazado por la tarjeta Arduino Mega en el robot transportador, mostrando flexibilidad y eficacia en diferentes entornos.

En la validación de las prácticas la observación destaca la importancia de que los estudiantes posean conocimientos básicos en diversas plataformas, ya que ello facilita la comprensión de la lógica de programación, incluyendo estructuras como condicionales (if), bucles (for), bloques de estado, entre otros. Estos fundamentos son esenciales para la manipulación efectiva de códigos y la resolución de problemas en entornos de desarrollo de sistemas robotizados.

El uso del EV3 como herramienta para la implementación de sistemas robóticos en el entorno educativo dirigido a niños de entre 10 y 14 años se revela como una experiencia altamente valiosa y formativa. La naturaleza modular y accesible del EV3 permite a los niños explorar la construcción física de modelos mediante el ensamblaje de piezas de manera intuitiva.

Los jóvenes de entre 15 y 18 años que ya cuentan con un cierto conocimiento de sistemas de programación y en algunos casos en la programación del EV3, la experiencia se transforma una estimulación hacia la exploración y la creatividad tecnológica. El EV3 no solo ofrece la oportunidad de ensamblar piezas físicas, sino que también permite la implementación de lógica de programación avanzada.

Para estudiantes en sus primeros años de universidad, el uso del EV3 como herramienta de enseñanza de robótica se presenta como un terreno propicio para la aplicación práctica y la personalización avanzada de sistemas robóticos y programación. La capacidad de modificar códigos de programación del EV3 haciendo uso de plataformas diferentes no solo requiere un entendimiento sólido de la lógica de programación, sino que también brinda una oportunidad única para la experimentación y la innovación.

VI. REFERENCIAS

- [1] Y. Zhan y M. Hsiao, Una aplicación de programación de lenguaje natural para LEGO Mindstorms EV3, Taichung, Taiwan: Conferencia Internacional IEEE 2018 sobre Inteligencia Artificial y Realidad Virtual (AIVR), 2018.
- [2] C. G.-V. J. A. Simón Rodríguez Pérez, «A New Didactic Method for Programming in C for Freshmen Students Using LEGO Mindstorms EV3, » University of Applied Sciences Aschaffenburg, 2016.
- [3] S. E. Kerry Emmanuel y M. P. Fredy Antonio, Estudio de la Aplicación Didáctica del LEGO Mindstorms NXT para la enseñanza de la robótica en la carrera de ingeniería en telecomunicaciones y electrónica en Control y Automatización,» 2016.
- [4] Kapitonov, Antonov, Artemov, Dobriborsci y Zamora, «LEGO Mindstorms EV3 para enseñar los conceptos básicos de los problemas de control de trayectoria,» 2018.
- [5] M. Mástil, Ambiyar, N. Jalinus, Ridwan y B. Hendrik, Robotic Education in 21st Century: Teacher Acceptance of LEGO Mindstorms as Powerful Educational Tools, 2021.
- [6] Z. Meirong y W. Yan, improving learning experiences using LEGO Mindstorms EV3 robots in control systems course, TX, 2020.
- [7] E. Samara, P. Angeldis y A. Kotsiari, «6ª Conferencia de Automatización del Diseño, Ingeniería Informática, Redes Informáticas y Redes Sociales del Sudeste de Europa (SEEDS-CECNSR),» 2021.
- [8] J. Chuan Tan, L. Iocchi, A. Eguchi y H. Okada, Artistas, Uniendo la Educación en Robótica entre la Escuela Secundaria y la Universidad: RoboCup@Home Education. [Art]. IEEE ÁFRICA, 2019.
- [9] P. Degener, V. Haak, C. Gold-Veerkamp y J. Abke, Artistas, Hacia la visión de una simulación LMS integrada basada en navegador para programar LEGO MindStorms EV3. [Art]. Conferencia Global de Educación en Ingeniería IEEE 2019 (EDUCON), 2019.
- [10] C. Buhr, Artist, Sesiones especiales: Enseñanza de controles simplificada con MATLAB y Simulink. [Art]. MSEC Salón de Plenos, 2017.
- [11] M. Nicolas, N. Rosillo, C. M. Marta y L. Hilario, A Novel Real-Time MATLAB/Simulink/LEGO EV3 Platform for Academic Use in Robotics and Computer Science, 2021.
- [12] L. Cornelissen, Simulating LEGO Mindstorms EV3 robots using Unity and Python, 2019.
- [13] O. Korkmaz, «The Effect of Scratch- and LEGO Mindstorms Ev3-Based Programming Activities on Academic Achievement, Problem-Solving Skills and Logical-Mathematical Thinking Skills of Students, » MOJES, 2016.
- [14] E. Fernández Moreno y J. Pastor-Mendoza, Artistas, Synergy LEGO Mindstorms – Arduino: Aprovechando ambas plataformas. [Art]. Universidad de Alcalá, España, 2018.
- [15] The LEGO Group, Guía de uso, LEGO, 2016.
- [16] V. C. Maricela, Introducción a LEGO Mindstorms EV3, Estado de Hidalgo, 2020.
- [17] LEGO, «LEGO Mindstorms EV3,» 2022. [En línea]. Disponible en: <https://www.lego.com/en-us/product/lego-mindstorms-ev3-31313>. [Último acceso: agosto 2022].
- [18] Mathworks, «LEGO Mindstorms EV3 Soporte de Simulink, » 2022. [En línea]. Available: <https://la.mathworks.com/hardware-support/LEGO-mindstorms-ev3-simulink.html>. [Último acceso: 04 2022].
- [19] V. Patrik, MakeCode for LEGO Mindstorms EV3, 2020.
- [20] J. Vega y J. M. Cañas, Entorno docente con Arduino y Python para Entorno docente con Arduino y Python para, Fuenlabrada, Madrid.
- [21] Rubio, Miguel Ángel Mañoso, Carolina Romero Zaliz, Rocío, Ángel, P. de Madrid, Uso de las plataformas LEGO y Arduino en la enseñanza de la programación, Cataluña España