GOBIERNO DE COLOMBIA

Ruta Grado Once
(11vo) de Secundaria

Sesión 6
Peaje de pago automático

LABORATORIO de Innovación Educativa

OEI

Computadores para Educar

LABORATORIO
de Innovación Educativa



TABLA DE CONTENIDO

Introducción 2
1. Prepara la clase 2
1.1 Recursos de las Nuevas Soluciones Tecnológicas 2
1.2 Espacio de trabajo 2
1.3 Conocimientos previos 2
1.4 Recomendaciones de uso de: las NST, los materiales adicionales y la organización de grupos de trabajo 3
2. Identificación del Contexto y problema para desarrollar 4
2.1 Problemática 4
2.2. Conceptos requeridos 5
3. Metas de aprendizaje STEM 8
3.1 Ciencias 8
3.2 Tecnología 9
3.3 Matemáticas 9
3.4 Ingeniería 9
3.5 Lenguaje 9
3.6 Matemáticas 9
3.7 Ciencias sociales 9
3.8 Ciencias naturales 10
3.9 Tecnología 10
3.10 Competencia ciudadana 10
4. Desarrollo práctico de la sesión 10
4.1 Objetivo 10
4.2 Aplicación práctica 12
4.3 Gestor de Contenidos 13
4.4 Impresión 3D 13
4.5 Construcción 15
4.6 Programación 15
4.7 Simulación 36
4.8 Monitor Interactivo 36
5. Reflexiona, evalúa y concluye 36
5.1 Reflexión y evaluación 36
6. Conclusión 37
7. Bibliografía 38

Introducción

Bienvenidos a la Ruta de Aprendizaje de Grado Once (11vo) de Secundaria. En esta Ruta, encontrará seis (6) sesiones enmarcadas en didácticas que permiten desarrollar en los estudiantes habilidades de Programación intermedia-avanzada, Pensamiento Computacional y trabajo colaborativo para la resolución de problemáticas planteadas en situaciones cotidianas y científicas.

A través de la metodología de aprendizaje basada en problemas (ABP) mediante la gestión de proyectos tecnológicos y el uso de herramientas que hacen parte de las Nuevas Soluciones Tecnológicas (de ahora en adelante, NST) como: la impresora 3D, bloques de construcción de ingeniería, micro:bit y sus kits, Arduino y dispositivos electrónicos, monitor interactivo, gestor de contenidos y elementos adicionales didácticos; los docentes y estudiantes podrán desarrollar actividades relacionadas con STEM (Ciencia, Tecnología, Ingeniería y Matemáticas).

1. Prepara la clase

1.1 Recursos de las Nuevas Soluciones Tecnológicas

En esta segunda sesión, se usarán los siguientes recursos de las NST:

- Gestor de Contenidos
- Monitor Interactivo
- Impresora 3D
- Kit de Ingeniería STEM

Los materiales adicionales que se usarán para el desarrollo de la sesión son:

- 1 rollo de cinta.

1.2 Espacio de trabajo

Organizar el salón de clase para que se pueda desarrollar la dinámica en grupos de a cinco (5) estudiantes que permitirá realizar el trabajo colaborativo y la división de roles propuesto.

1.3 Conocimientos previos

- Qué es un peaje
- Sensor de ultrasonido
- Driver para motor

1.4 Recomendaciones de uso de: las NST, los materiales adicionales y la organización de grupos de trabajo

Para el desarrollo de esta sesión se recomienda verificar el correcto funcionamiento de las NST que se van a utilizar y también los materiales adicionales señalados.

Por otro lado, para fomentar el trabajo colaborativo, se recomienda asignar a los estudiantes los siguientes roles basados en Liderazgo, Ingeniería, Programación y Regulador de tránsito.

La descripción de las labores de cada grupo son:

Líder (1): Es el encargado de guiar al equipo, verificar el correcto avance de cada uno de los roles de trabajo en sus respectivas funciones, asegurar que el trabajo realizado por el equipo esté enfocado en dar solución al problema propuesto y encontrar soluciones para posibles problemáticas que se puedan presentar en el desarrollo de la actividad. Adicionalmente, tendrá que ayudar a los roles de trabajo a realizar sus actividades, lo que significa que deberá programar, construir y conectar dispositivos electrónicos con el fin de completar todo el prototipo, en el tiempo disponible, como se propone en la sesión.

Ingenieros (2): Son los encargados de construir los prototipos sugeridos en la actividad y organizar los materiales físicos a utilizar en la práctica. En este caso, los materiales a usar son los bloques de construcción de ingeniería, cinta, los elementos impresos en 3D, y todos los dispositivos electrónicos que se deben integrar al prototipo.

Programador (1): Es el encargado de realizar las actividades que implican el uso de pensamiento lógico y computacional que se plantean en la dinámica de la sesión. En este caso, deben realizar los códigos propuestos para la práctica, comprender su funcionamiento y asegurarse de que cada una de las tareas programadas se ejecute de forma correcta sin fallas, haciendo uso

de simuladores y pruebas de escritorio, previas a la implementación de los algoritmos en el prototipo final, para verificar su funcionamiento.

Regulador de tránsito (1): Es el encargado de probar y verificar el funcionamiento del prototipo realizado por el grupo, con el fin de comprobar que el modelo cumpla con los objetivos de la práctica y que otorgue la solución esperada al problema propuesto. Adicionalmente, deberá trabajar en las conexiones electrónicas del prototipo siguiendo los pasos indicados en la guía. En este caso, tendrá que verificar que el robot móvil frene automáticamente al detectar obstáculos dentro del rango de distancia establecido, avance cuando no haya obstáculos, que el peaje realice el cobro del vehículo y este pague automáticamente para poder levantar la talanquera y cruzar, y que todas las conexiones electrónicas sean correctas y no se desconecten de sus lugares.

2. Identificación del Contexto y problema para desarrollar

2.1 Problemática

El tráfico de entrada y salida de la ciudad capital Bojomá es demasiado pesado en las horas pico, debido al alto flujo de vehículos y la enorme demanda de importaciones y exportaciones nacionales que realiza la ciudad. El padre de Rafael debe entrar y salir de la capital todos los días debido a su trabajo, y ha notado que una razón de peso que provoca los atascos es el tiempo que le toma a cada persona contar el dinero para pagar el peaje.

Cuando Rafael escuchó a su padre hablar del tráfico y lo demorado que es pagar el peaje, decidió comentarle a sus compañeros sobre el asunto y evaluaron la posibilidad de acelerar el paso vehicular haciendo que los pagos del peaje sean automáticos y debitados de cuentas bancarias o tarjetas pre-pagadas.

Con su grupo de trabajo, decidieron construir un prototipo de peaje de pago automático, donde el proceso de pago se pueda realizar con un débito automático de una cuenta electrónica. Ayuda a Rafael y su equipo a construir, programar y probar el prototipo.

2.2. Conceptos requeridos

Qué es un peaje

Un peaje es un pago obligatorio para tener acceso a una vía de tránsito, ya sea de comunicación terrestre o marítima. Se instalan peajes con el fin de obtener ingresos debido a la explotación, o uso de terceros, de una infraestructura o vía de transporte.



Generalmente, se ubican peajes en autopistas, pasos a nivel, túneles, puentes, o en pasajes marítimos o canales para poder navegar en determinadas zonas marítimas o cruzar de un punto a otro.

La modalidad de cobro de un peaje viene asociado a las siguientes características:

- Su motivación principal es obtener una contraprestación por el uso de unas instalaciones.
- Establece distintas tarifas dependiendo de criterios como:

  - Distancia
  - Tipo de vehículo
  - Tramo de horario
  - Tiempo de disfrute.

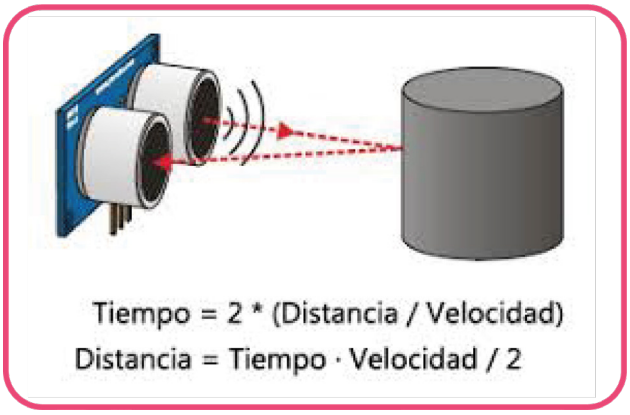
- Los peajes públicos son un elemento recaudatorio.
- Los peajes privados se usan para obtener ganancias, con el fin de recuperar la inversión realizada en la construcción de una vía.
- Es posible pagar de distintas maneras, como efectivo o tarjeta bancaria. Adicionalmente, con el desarrollo tecnológico se han creado nuevos sistemas de pago inalámbrico o mediante red telefónica.

Sánchez, J. (5 de octubre de 2022). Peaje. Economipedia.com. https://economipedia.com/definiciones/peaje.html

Sensor de ultrasonido

Un sensor de ultrasonido es un dispositivo electrónico diseñado para medir distancias a partir del envío y recepción de ondas de ultrasonido.

El sensor posee un cabezal que emite una onda ultrasónica y otro que recibe la onda reflejada, después de que ésta ha chocado contra un objeto (el objeto detectado). Algunos sensores de ultrasonido poseen un único cabezal que realiza ambas tareas. Dependiendo del tiempo que tarda la onda en ir, golpear el objeto y volver al receptor, se determina la distancia a la que se encuentra el elemento.



La distancia entre el sensor y el objeto detectado se calcula como el tiempo que tarda la onda en ir y volver (t) multiplicado por la velocidad del sonido (c) y dividido entre dos, como se muestra en la siguiente fórmula:

$$\text{Distancia} = \frac{t*c}{2}$$

Donde c= 343.2 $\frac{m}{s}$ en la atmosfera terrestre

Algunas características relevantes de las mediciones realizadas por este sensor son:

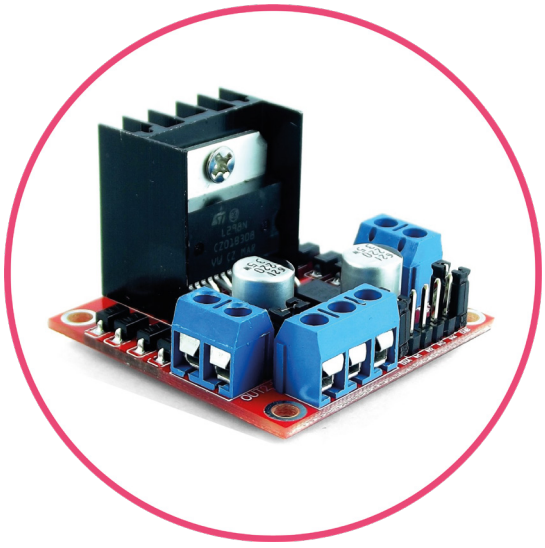
- Detecta objetos transparentes: Esto es posible gracias a que las ondas de ultrasonido pueden rebotar en vidrio o líquidos.
- Es resistente a niebla y suciedad: La detección no se afecta por acumulación de polvo o suciedad.
- Detecta objetos con formas complejas: Logra detectar objetos de formas complejas como bandejas de malla o resortes.

Keyence. (5 de octubre de 2022). ¿Qué es un sensor ultrasónico? Keyence.com.mx. https://www.keyence.com.mx/ss/products/sensor/sensorbasics/ultrasonic/info/

Driver para motor DC

Un driver o controlador de motor es un pequeño amplificador de corriente que se encarga de tomar una señal de control de entrada de baja corriente y convertirla en una señal de corriente más alta que sea capaz de mover un motor.

El kit de ingeniería STEM tiene el driver puente H L298N, el cual está diseñado para controlar 2 motores de hasta 2 amperios cada uno, o un motor paso a paso bipolar/unipolar. Este módulo, permite controlar tanto la dirección como la velocidad de giro de los motores, donde se usan dos pines de entrada para cada motor que permiten definir el sentido de giro dependiendo de cuál de ellos está encendido y cuál apagado, y mediante una modulación de ancho de pulso (PWM) en los pines de entrada se controla la velocidad.



Las características principales de este driver son:

- Tiene 2 canales de control que soportan 2 motores DC o un motor paso a paso.
- Maneja un voltaje de control lógico de hasta 5V.
- Maneja voltajes de potencia para los motores de entre 5V y 35V DC (corriente continua).
- La corriente de control lógica debe estar entre 0 y 36mA.
- Soporta corrientes de potencia para los motores de hasta 2A y picos de corriente de hasta máximo 3A.
- Tiene una potencia máxima de 25W.
- Admite entradas de señal PWM en sus pines de control lógico para regular la velocidad.

Dademuchconnection. (26 de abril de 2018). Driver de motor DC – Electrónica de potencia. Dademuch.com. https://dademuch.com/2018/04/26/driver-de-motor-dc-electronica-de-potencia/

Naylamp Mechatronichs. (5 de octubre de 2022). DRIVER PUENTE H L298N 2A. NaylampMechatronics.com. https://naylampmechatronics.com/drivers/11-driver-puente-h-l298n.html

3. Metas de aprendizaje STEM

3.1 Ciencias

Observar el entorno e identificar problemas y necesidades, con el fin de proponer posibles soluciones que ayuden a solventar o disminuir los problemas identificados.

3.2 Tecnología

Combinar el uso de diferentes dispositivos electrónicos, fuentes de alimentación y tarjetas programables para construir un prototipo funcional, que permita relacionar diferentes elementos de sensado y control, con el fin de construir un sistema tecnológico capaz de solucionar una necesidad o problema de la región, el país y la comunidad.

3.3 Matemáticas

Analizar cómo el uso de elementos geométricos modulares permite construir diferentes estructuras, tanto móviles como estáticas, para la realización de un prototipo.

3.4 Ingeniería

Emplear bloques de construcción ensamblables de distintas formas y tamaños, junto con dispositivos electrónicos, fuentes de alimentación y tarjetas programables, con el fin de construir estructuras móviles y estáticas que interactúen entre sí para dar como resultado un prototipo funcional.

Metas establecidas por el Ministerio de Educación Colombiano

Adicionalmente, durante el desarrollo práctico de esta sesión, se busca que los estudiantes cumplan con las siguientes metas de aprendizaje establecidas por el Ministerio de Educación Colombiano:

3.5 Lenguaje

Comprendo el valor del lenguaje en los procesos de construcción del conocimiento.

3.6 Matemáticas

Justifico resultados obtenidos mediante procesos de aproximación sucesiva, rangos de variación y límites en situaciones de medición.

3.7 Ciencias sociales

Defino los objetivos y la hipótesis del trabajo.

3.8 Ciencias naturales

Analizo el desarrollo de los componentes de los circuitos eléctricos y su impacto en la vida diaria.

3.9 Tecnología

Diseño, construyo y pruebo prototipos de artefactos, sistemas o procesos como respuesta a una necesidad o problema, teniendo en cuenta las restricciones y especificaciones planteadas.

3.10 Competencia ciudadana

Analizo críticamente las decisiones, acciones u omisiones que se toman en el ámbito nacional o internacional y que pueden generar conflictos o afectar los derechos humanos.

4. Desarrollo práctico de la sesión

4.1 Objetivo

Construir, programar y probar un prototipo de un sistema de peaje con pago automático, donde se haga uso del Arduino para controlar un robot móvil, un micro:bit para controlar la talanquera del peaje y un segundo micro:bit para pagar la cuota establecida por el peaje justo cuando se reciba el cobro enviado de forma inalámbrica.

El estudiante deberá integrar todas las tarjetas programables vistas a lo largo de la ruta en un único proyecto y acompañar el sistema con diferentes dispositivos electrónicos que tendrá que conectar y programar de forma adecuada para lograr cumplir con todas las tareas definidas para el peaje.

4.2 Aplicación práctica

Resumen de la actividad

En esta actividad los estudiantes tendrán que construir estructuras móviles y estáticas, e integrarlas con diferentes dispositivos electrónicos que se relacionarán entre sí para alcanzar un prototipo de un sistema de peaje con pago automático.

Los elementos que deberá construir y programar el estudiante son:

- Un robot móvil controlado por Arduino que frene al detectar un obstáculo (en este caso la talanquera del peaje) y avance al no detectar obstáculos. Este elemento simulará el paso de vehículos por el peaje.
- Un peaje controlado por un micro:bit que detecte cuando un carro llegue al peaje, envíe automáticamente un cobro al vehículo, y levante la talanquera cuando el cobro sea efectivo.
- Un segundo micro:bit que reciba el cobro enviado por el peaje, realice el pago correspondiente y le avise al vehículo cuando está listo para cruzar.

Nota : En caso de que el docente considere que el tiempo destinado para la sesión no es suficiente para completar todo el sistema de peaje con cobro automático, puede omitir el robot móvil controlado por Arduino y programar únicamente los 2 micro:bits, usando vehículos caseros, o construidos con las fichas, para simular el paso de los carros por el peaje.

Se le recomienda al docente que distribuya las tareas entre los estudiantes de la siguiente manera, donde todos trabajen de forma simultánea, con el fin de aprovechar al máximo el tiempo disponible para la sesión:

- **Grupo de ensamble “Peaje”:** este equipo, conformado por los roles de ingeniería, debe estar a cargo de la construcción física del peaje usando los bloques de construcción de ingeniería. Como este modelo tiene pocos pasos de ensamble, deben pasar a ayudar al “grupo de ensamble carro” una vez que finalicen sus actividades.
- **Grupo de ensamble “Carro”:** este equipo, conformado por los roles de ingeniería, debe estar a cargo de la construcción física del carro usando los bloques de construcción de ingeniería. Es importante que tengan en cuenta la ubicación y conexión de los dispositivos electrónicos para evitar que se caigan o se suelten del modelo cuando este se mueva. Deben trabajar de la mano con el “grupo de conexión electrónica” con el fin de ubicar correctamente todos los elementos del prototipo.
- **Grupo de programación micro:bit:** este equipo, conformado por los roles de programación, debe estar a cargo de la programación de los 2 micro:bits. Una vez que finalicen sus actividades, deben pasar a conectar el micro:bit emisor al GamePad, ponerlo en el prototipo construido del peaje y unirle la talanquera 3D al eje del servomotor, con el fin de probar el peaje y verificar su funcionamiento.
- **Grupo de programación Arduino:** este equipo, conformado por los roles de programación, debe estar a cargo de la programación del Arduino. Una vez que

finalicen sus actividades, deben trabajar de la mano con el "grupo de conexión electrónica" para conectar el Arduino a todos los dispositivos electrónicos del carro, y finalmente, ambos grupos se deben unir al "grupo de ensamble carro" para unir toda la parte electrónica con el modelo físico del prototipo.

- **Grupo de conexión electrónica:** este equipo, conformado por los roles de líder y regulador de tránsito, debe estar a cargo de toda la conexión de los dispositivos electrónicos, siguiendo las imágenes y pasos descritos en esta sesión, con el fin de conectar todos los elementos entre sí, verificar que ninguno se desconecte por accidente y asegurar que las fuentes de alimentación se conecten correctamente.

Nota : se recomienda que el docente esté pendiente del trabajo de cada grupo y que apoye principalmente al "grupo de conexión electrónica", con el fin de verificar que todos los elementos se conecten correctamente antes de alimentar el sistema, y así evitar cualquier falla o daño electrónico.

4.3 Gestor de Contenidos

¿Qué es el Gestor de Contenidos?

El Gestor de Contenidos es un repositorio de recursos educativos que genera una red inalámbrica con ayuda de un router al que se encuentra conectado para permitir la conexión de hasta 45 dispositivos externos como computadoras, celulares y tabletas, los cuales pueden acceder al Gestor para explorar, visualizar y descargar sus contenidos y generar un correcto desarrollo de las sesiones. Este sistema trabaja completamente Offline (desconectado) y le entregará al docente todos los recursos que necesitará para las sesiones.

En el Gestor de Contenidos se aloja todo el material digital necesario para el correcto desarrollo de esta sesión. Encontraremos en él:

- Esta sesión en formato PDF.
- Archivos de impresión 3D en formato editable .stl y en formato de impresión .gcode. La ruta que se debe seguir en el Gestor de Contenidos para encontrar los archivos de impresión 3D es: ***Repositorio Recursos Rutas CPE. Fase II – Ruta 11– Sesión6- Recursos Impresión 3D***
- Un video que muestra el funcionamiento del prototipo.

- Peaje pago automático
- La ruta para encontrar este video en el Gestor de Contenidos es: ***Repositorio Recursos Rutas CPE. Fase II – Ruta 11–Sesión6- Videos didácticos***

- Videos relacionados con los conceptos requeridos para esta sesión. El docente los puede utilizar para exponer a los estudiantes de una manera visual conceptos que son necesarios para este tema. Los videos sugeridos son los siguientes:

- Cómo usar el Driver del motor L298N con Arduino
- Qué es un sensor de ultrasonido
- Qué es un peaje
- La ruta para encontrar los videos en el Gestor de Contenidos es: ***Repositorio Recursos Rutas CPE. Fase II – Ruta 11–Sesión6- Videos didácticos***

4.4 Impresión 3D

¿Qué es la impresión 3D?

La impresión 3D es una tecnología que usa la fabricación por adición donde un objeto tridimensional es creado a través de superposición de capas sucesivas de material basado en polímeros plásticos como lo son el PLA, ABS, entre otros. La impresora derrite dicho polímero a altas temperaturas y a través de una boquilla de diámetro muy fino deposita el material derretido capa por capa hasta formar el prototipo creado anteriormente en un software de diseño digital.

Antes de iniciar con el desarrollo de la didáctica, se recomienda imprimir los prototipos sugeridos para esta sesión 1 hora antes, lo cual permitirá ahorrar tiempo en el desarrollo de la sesión.

El prototipo a imprimir es una talanquera, la cual permitirá darle la indicación al vehículo sobre cuándo avanzar y cuándo frenar. El modelo tiene una hendidura rectangular diseñada para acoplarla con un accesorio del eje del servomotor y ajustarlo posteriormente con cinta.

El prototipo a imprimir cuenta con el siguiente diseño:

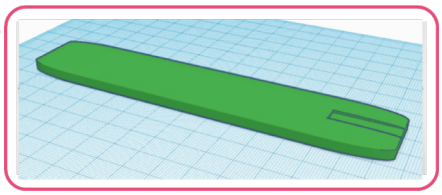


Imagen 1. Talanquera para servomotor

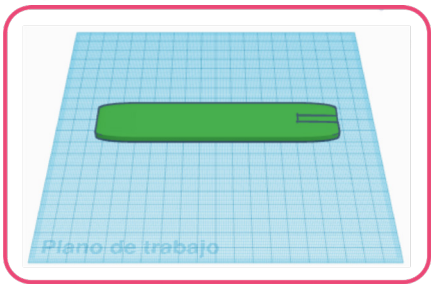


Imagen 2. Objeto a escala con respecto a la plataforma de impresión. Sus dimensiones son de aproximadamente 150mm x 30mm x 5mm

PARÁMETROS DE IMPRESIÓN

Formato de archivo editable	.slt	**Creación de balsa**	No
Formato de archivo para la impresión .gcode	.gcode	**Dimensiones y escala del prototipo a imprimir**	Establecidas en el prototipo guardado en el gestor de contenidos. Editable en formato .stl, listo para imprimir en formato .gcode.
Color de filamento	A elección del docente	**Velocidad de impresión recomendada**	50 mm/s
Calidad de impresión	Standard	**Creación de soportes**	Si
Relleno sugerido	30%	**Tiempo estimado de impresión**	55 minutos
Temperatura de extrusión	200° Celsius		
Temperatura de base o cama de impresión	60° Celsius	**Cantidad de filamento usado en el prototipo**	5180mm

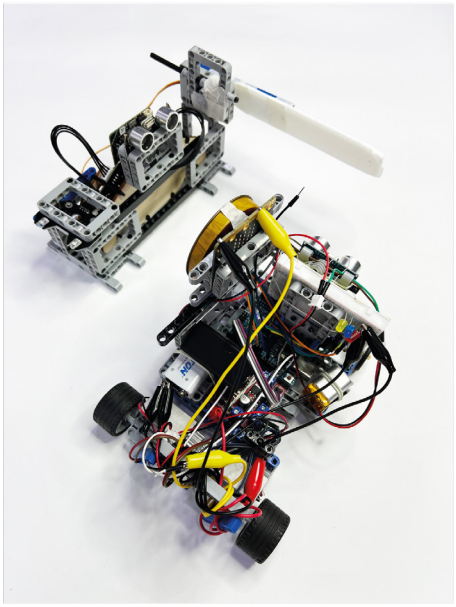
Nota : Parámetros adicionales que se presentan en el software Cura no deben ser ajustados o modificados. Ya el prototipo presentado para esta sesión, cuenta con los parámetros guardados para asegurar una impresión de calidad.

Biblioteca de archivos descargables para la impresión 3D. Los archivos editables y descargables para la impresión del prototipo de esta sesión se encuentran en el Gestor de Contenidos siguiendo la siguiente ruta: ***Repositorio Recursos Rutas CPE. Fase II – Ruta 11– Sesión6- Recursos Impresión 3D***

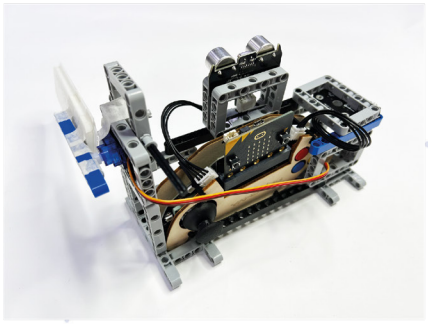
4.5 Construcción

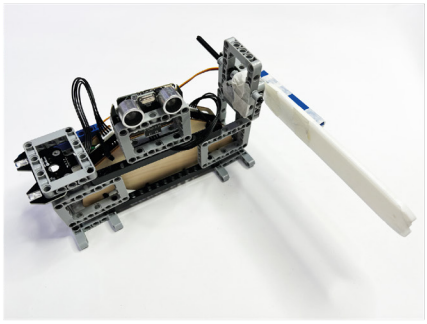
Para el correcto desarrollo de esta actividad, es necesario que se realicen las siguientes estructuras para simular un peaje y un vehículo motorizado. El peaje permitirá ubicar un servomotor y un sensor de ultrasonido para detectar la presencia de vehículos y permitir o negar el paso según el pago registrado, y el robot móvil representará el vehículo que paga el peaje y cruza al otro lado.

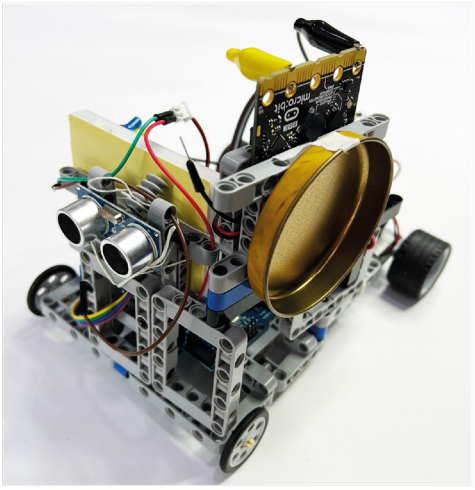
Prototipo completo:

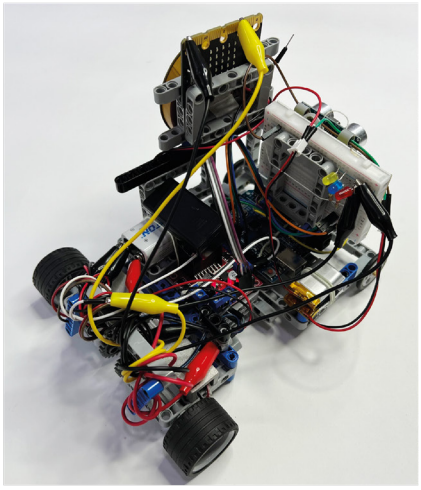


Peaje









Vehículo:

Los pasos para ensamblar este prototipo se muestran a continuación. Recuerde que estas imágenes y esta estructura son solo una guía para el correcto desarrollo de la práctica, los estudiantes tienen completa libertad para armar otra estructura o modificar la propuesta en este documento, siempre y cuando dé solución a la problemática indicada en esta sesión:

PASOS DE ENSAMBLE

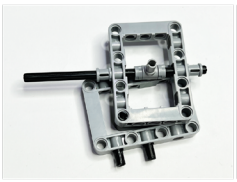
1



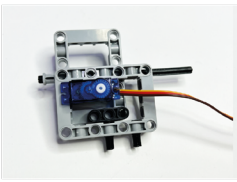
2



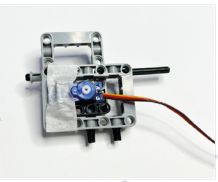
3



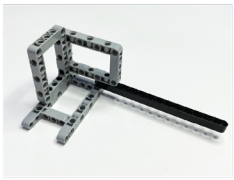
4



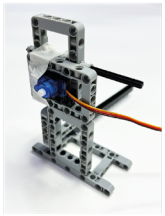
5



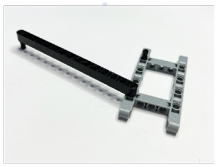
6



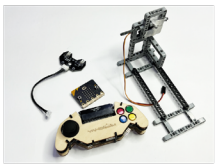
7



8



9



10



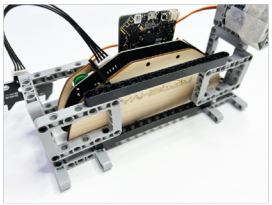
11



12



13



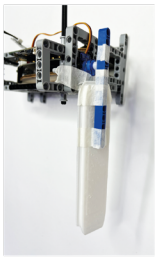
14



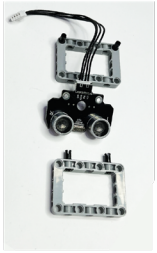
15



16



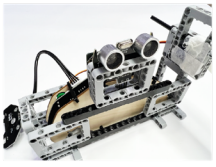
17



18



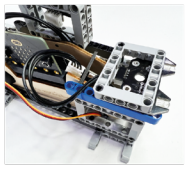
19



20



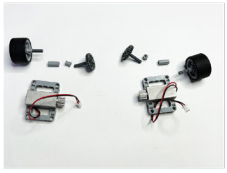
21



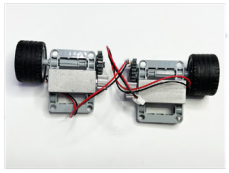
Peaje:

PASOS DE ENSAMBLE

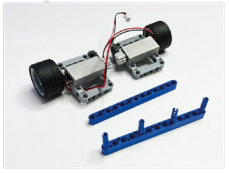
1



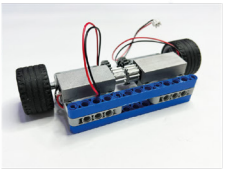
2



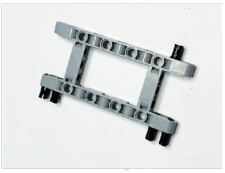
3



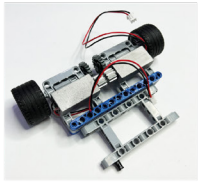
4



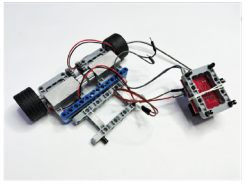
5



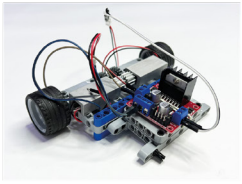
6



7



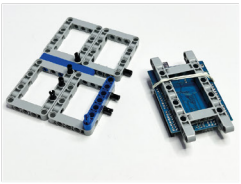
8



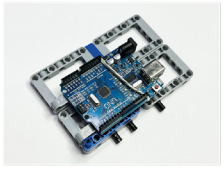
9



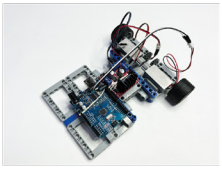
10



11



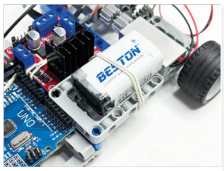
12



13



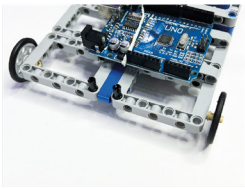
14



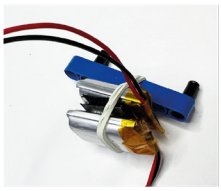
15



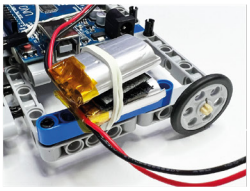
16



17



18



19

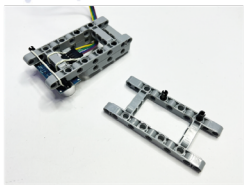


20

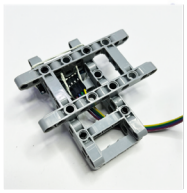


PASOS DE ENSAMBLE

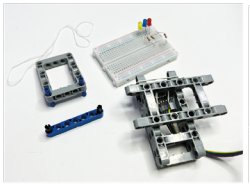
21



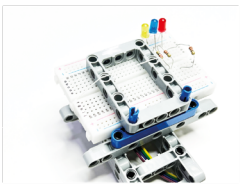
22



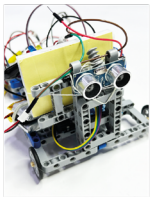
23



24



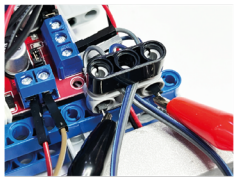
25



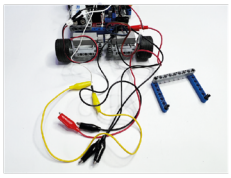
26



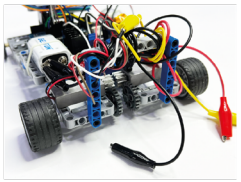
27



28



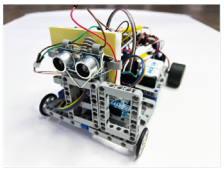
29



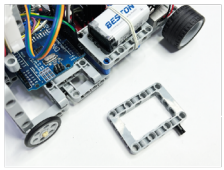
30



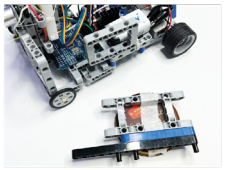
31



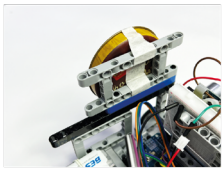
32



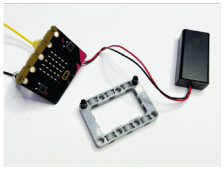
33



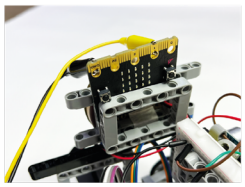
34



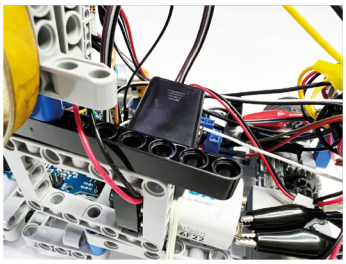
35



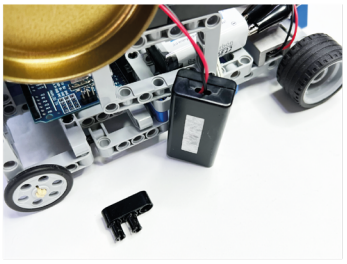
36



37



38



4.6 Programación

Micro:bit

¿Qué es la micro:bit?

La micro:bit es una tarjeta programable de 4x5cm que se puede programar mediante lenguaje de bloques y utilizarse en todo tipo de aplicaciones tecnológicas, ya sea sola o conectada a otras micro:bit, acompañada de cables tipo caimán y hojas de papel aluminio, o conectada a todo tipo de sensores y robots.

Esta tarjeta cuenta con una matriz LED de 5x5 LEDs, 2 botones programables, un parlante y un amplio conjunto de sensores integrados que se pueden programar para usarse en cualquier aplicación, como un acelerómetro, un micrófono, un sensor de luz, un sensor de temperatura, una brújula, un sensor touch y un conjunto de hasta 20 pines de entrada y salida digitales y analógicos.

MakeCode

¿Qué es MakeCode?

MakeCode es una plataforma de programación web que se puede abrir desde cualquier navegador donde se utiliza un método de programación por bloques para construir los algoritmos que se descargarán en la micro:bit con el fin de realizar alguna actividad tecnológica. Los bloques tienen diferentes formas geométricas que determinan cómo se conectan, están organizados por categorías y la mayoría de ellos están traducidos al español. La plataforma también cuenta con un emulador de la micro:bit donde se pueden probar todos los algoritmos antes de descargarlos en la tarjeta.

Adicionalmente, si se trabaja MakeCode en el navegador Google Chrome actualizado, es posible conectar la tarjeta al navegador ya sea para visualizar la lectura de sus sensores o los datos que se deseen mostrar en el monitor serial, o para programarla directamente con un solo clic.

En esta sesión se usará la micro:bit y la plataforma de programación MakeCode para realizar un algoritmo que cumpla con las siguientes tareas:

Micro:bit emisor: esta tarjeta estará conectada al GamePad, el cual tendrá conectado un módulo RGB, un sensor de ultrasonido y un servomotor.

- Cuando el sensor de ultrasonido detecte un objeto a más de 10cm significará que no hay vehículo, por lo tanto el LED RGB se enciende de color rojo y el servomotor mantiene cerrada la talanquera (talanquera abajo).
- Cuando el sensor de ultrasonido detecte un objeto a menos de 10cm entonces envía por radio una cadena que dice “Pagar”. Si el micro:bit receptor le responde por radio el texto “Pagado” entonces el micro:bit emisor activa el servomotor para subir la talanquera por 3 segundos, enciende el LED RGB de color verde y, al terminar los 3 segundos baja la talanquera y enciende el LED RGB de color rojo.

Micro:bit receptor: Esta tarjeta estará instalada en el robot móvil junto al Arduino. El pin P1 del micro:bit debe estar conectado al pin 8 del Arduino.

- Cuando el micro:bit receptor reciba por radio la palabra “Pagar” entonces enviará por radio la palabra “Pagado”, y escribirá en el pin P1 un 1 para indicarle al Arduino que ya se ha pagado el peaje. Cuando el Arduino detecte el 1, encenderá un LED amarillo por 3 segundos como señal de que ha pagado exitosamente.

Tenga en cuenta las siguientes recomendaciones antes de iniciar la programación:

- Para ingresar a MakeCode debe escribir el siguiente link en el buscador (preferiblemente Google Chrome), o instalar y abrir MakeCode Offline en el dispositivo donde se vaya a trabajar (el instalador se encuentra en el Gestor de Contenidos)

  - Dirección URL de MakeCode: ***https://makecode.microbit.org/***
  - Dirección en el Gestor de Contenidos de la aplicación MakeCode Offline: ***Repositorio Técnico NST. Fase II – Instaladores***

- Para resolver cualquier duda o inquietud sobre los bloques de programación, el correcto uso de MakeCode y micro:bit, o la conexión de los elementos electrónicos, puede remitirse al Manual del Kit de Ingeniería STEM en la sección

de “Software” o “Hardware” según corresponda.

- Dirección del Manual del Kit de Ingeniería STEM en el Gestor de Contenidos: ***Repositorio Técnico NST. Fase II – Manuales***

- Para usar las librerías necesarias en los algoritmos propuestos, recuerde cargar la librería que se encuentra en el Gestor de Contenidos, siguiendo los pasos que se describen en el Manual del Kit de Ingeniería STEM.

  - Dirección de las librerías en el Gestor de Contenidos: Repositorio Técnico NST. Fase II – Librerías
  - Dirección del Manual del Kit de Ingeniería STEM en el Gestor de Contenidos: Repositorio Técnico NST. Fase II – Manuales.

- Si presenta dificultades para programar el algoritmo propuesto, puede descargarlo del Gestor de Contenidos en la siguiente dirección. Recuerde que estos algoritmos puede cargarlos a MakeCode si los quiere ver, editar o simular, o descargarlos directamente en la micro:bit:
Repositorio Recursos Rutas CPE. Fase II – Ruta 11–Sesión6- Micro:bit

En la siguiente imagen se muestra un esquema de conexión de la tarjeta micro:bit a los elementos electrónicos adicionales requeridos para esta aplicación. Conecte todo tal y como se muestra en la imagen para evitar posibles fallas:

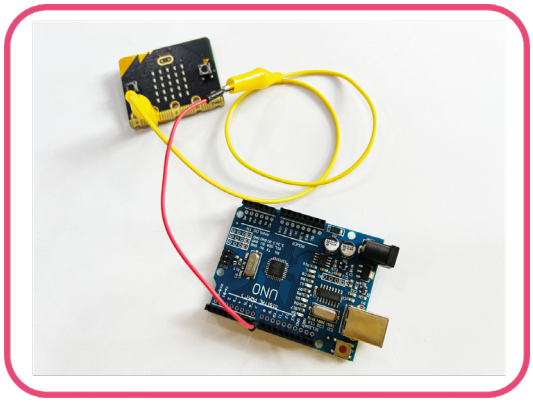
Micro:bit emisor y GamePad:

- Conecte los pines dorados del micro:bit en la ranura para micro:bit del GamePad. Asegúrese de que la matriz LED del micro:bit mire hacia delante.
- Conecte el sensor de ultrasonido en el puerto para ultrasonido del GamePad.
- Conecte el servomotor en los pines J5 para servomotor del GamePad.
- Conecte el módulo RGB en el puerto para módulo RGB del GamePad.

Nota : En el Manual de uso del kit STEM en la sección “Cómo usar el kit de ingeniería STEM en las rutas didácticas – Hardware” puede encontrar información sobre los puertos de conexión del GamePad y cómo conectar los sensores y actuadores al GamePad.



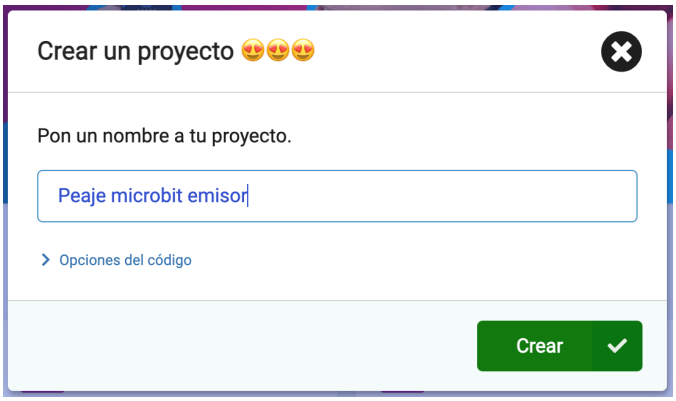
Micro:bit receptor:



A continuación, se presentan los pasos necesarios para programar el algoritmo de esta sesión:

Micro:bit emisor:

1. Crea un nuevo proyecto y defínele un nombre, por ejemplo, “Peaje microbit emisor”.



2. Carga la librería “Librería-CPE.hex” siguiendo los pasos que se indican en el Manual de uso del kit STEM en la sección “Cómo usar el kit de ingeniería STEM en las rutas didácticas – Software – Librerías del kit STEM”. Pasa al siguiente paso cuando en tu listado de categorías aparezca la categoría “GHBit”.

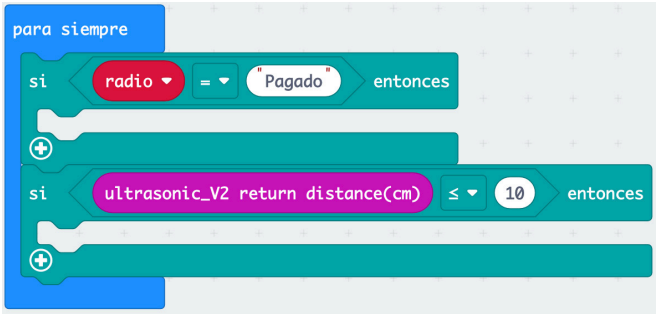
3. En el bloque “al iniciar” coloca el bloque de “radio establecer grupo” de la categoría “Radio” y define el número 1. Luego, busca en la nueva categoría del MakeCode “GHBit” los bloques:

- “Servo_Handle num” y colócalo en el bloque “al iniciar” con el valor 90° y S1. El valor S1 indica que se ha conectado el servomotor en los pines J5, y el 90° indica que el servomotor tendrá la talanquera abajo.
- “RGB_Colorful” y escoge el color “RED” para rojo.



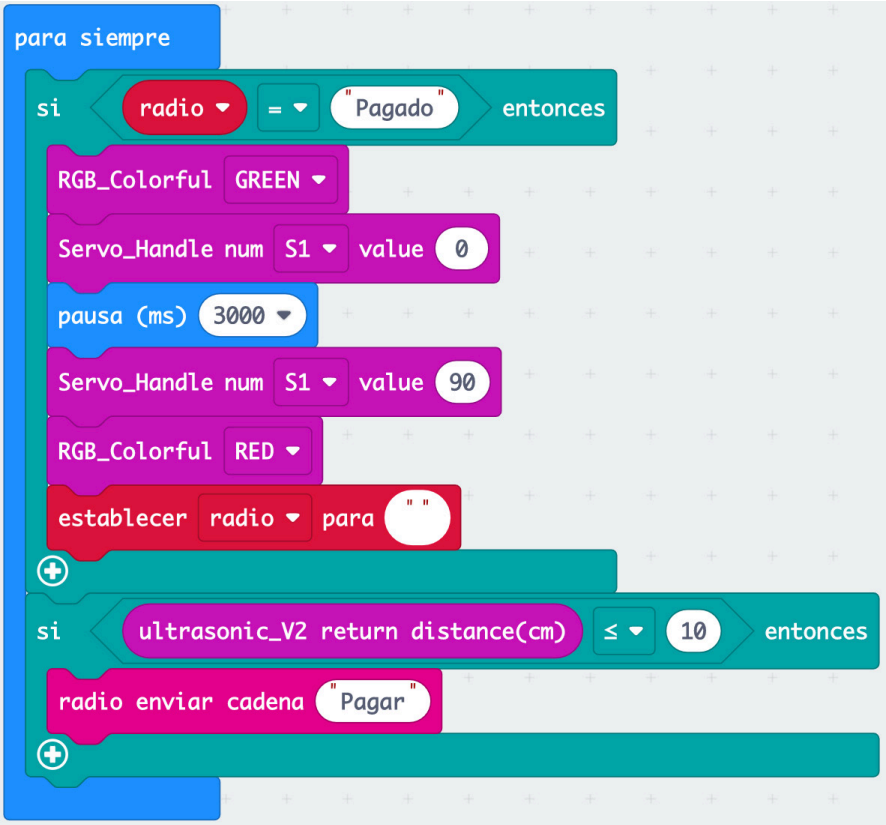
4. Primero, crea una variable llamada “radio”, luego en el bloque “para siempre” agrega dos condicionales “si... entonces”, uno debajo del otro, y programa las condiciones:

- “Si radio = “Pagado”” para realizar la apertura y cierre de la talanquera cuando el carro realice el pago del peaje.
- “si ultrasonic_V2 return distance (cm) es menor o igual a 10”. En la categoría “GHBit” encontrarás el bloque “ultrasonic_V2 return distance (cm)”. Para detectar cuando el carro llegue al peaje.



5. Dentro de los condicionales que acabas de poner, programa las siguientes tareas:

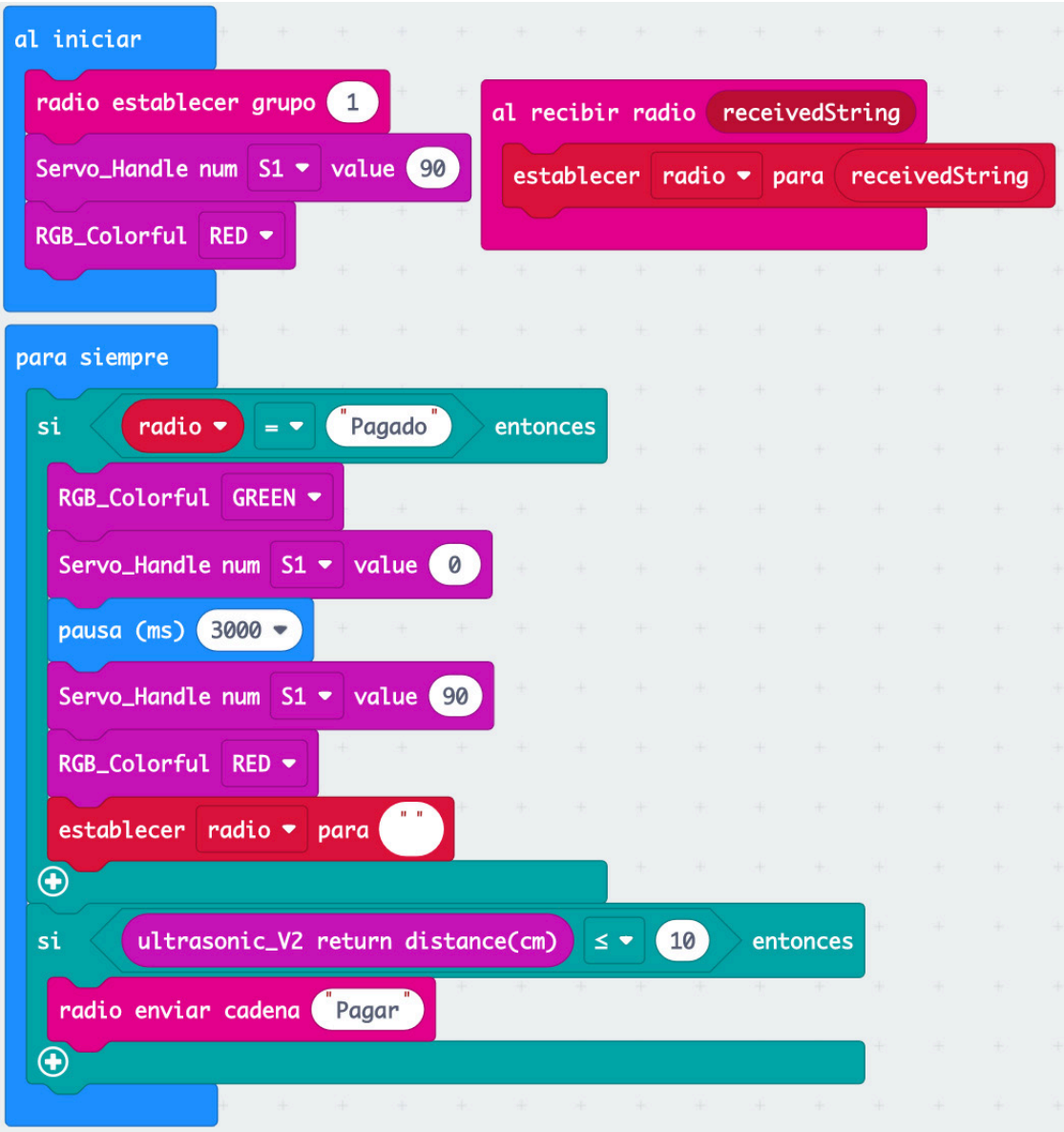
- Si radio = “Pagado”: Encender el módulo LED RGB de color verde, mover el servomotor a 0° para subir la talanquera, esperar 3 segundos, mover el servomotor a 90° para bajar la talanquera, encender el modulo LED RGB de color rojo y borrar la variable radio.
- Si ultrasonic_V2 return distance (cm) es menor o igual a 10: Enviar por radio la cadena “Pagar” para realizar el cobro al carro que ha llegado al peaje.



6. Agrega un bloque de la categoría “Radio” llamado “al recibir radio receivedString” al área de trabajo y programa dentro de él la instrucción de guardar en la variable “radio” el valor que se haya recibido por la comunicación por radio.

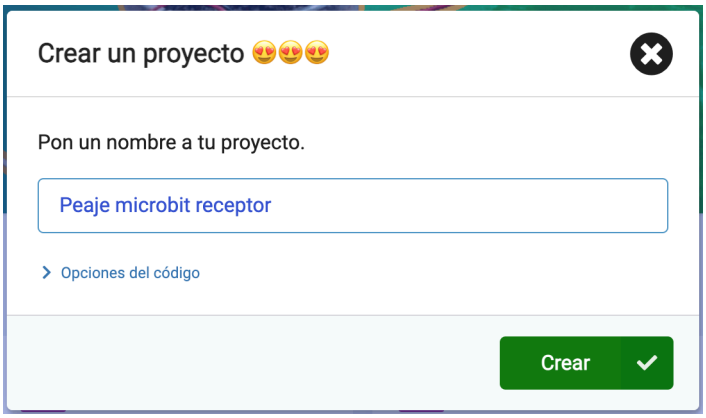


7. Tu algoritmo debe quedar como se muestra a continuación:

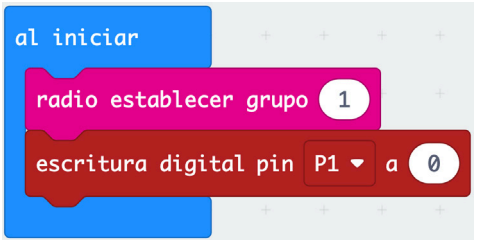


Micro:bit receptor:

1. Crea un nuevo proyecto y defínele un nombre, por ejemplo, “Peaje microbit receptor”.



2. En el bloque “al iniciar” coloca el bloque de “radio establecer grupo” de la categoría “Radio” y define el número 1. Luego, ve a la categoría “Pines”, busca el bloque “Escritura digital pin” y colócalo en el bloque “al iniciar”, con el valor P1 y 0.

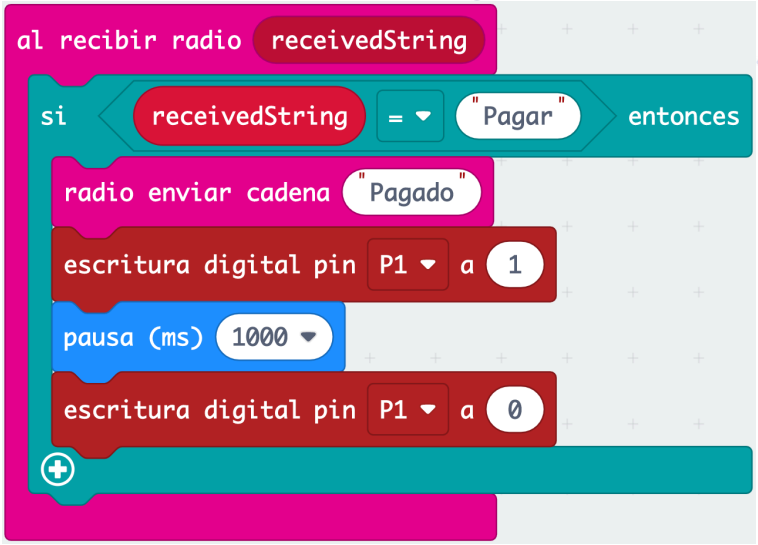


3. Agrega un bloque de la categoría “Radio” llamado “al recibir radio receivedString” al área de trabajo y programa dentro de él un condicional “si... entonces” con la condición “si receivedString = “Pagar””.

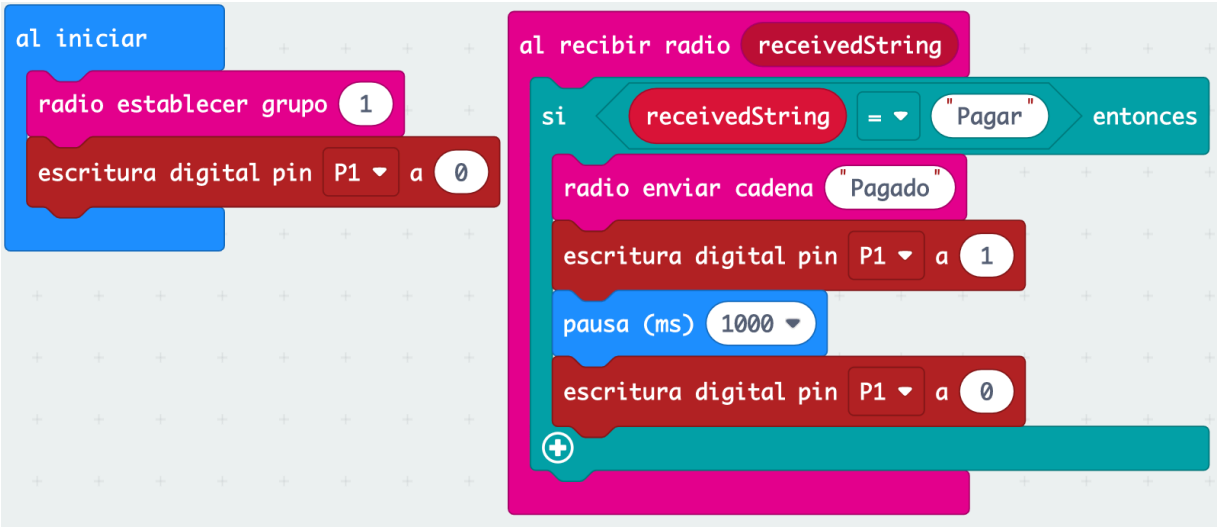


4. Dentro del condicional que acabas de agregar programa las tareas:

- Enviar cadena por radio “Pagado”
- Escribir un 1 en el pin P1
- Esperar 1 segundo
- Escribir un 0 en el pin P1



5. Tu algoritmo debe quedar como se muestra a continuación:



Arduino UNO

¿Qué es Arduino?

Arduino es una plataforma de programación basada en una placa electrónica de hardware libre que incorpora un microcontrolador re-programable y una serie de pines hembra. Estos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla.

La placa electrónica Arduino es una placa de circuito impreso o PCB por sus siglas en ingles (Printed Circuit Board), que implementa un determinado diseño de circuitería interna y de esta forma el usuario final no se debe preocupar por las conexiones eléctricas que necesita el microcontrolador para funcionar, y puede empezar directamente a desarrollar las diferentes aplicaciones electrónicas que requiera[1].

En esta sesión se usará el Arduino para realizar un algoritmo que cumpla con las siguientes tareas:

- Leer la distancia en centímetros registrada por un sensor de ultrasonido y frenar el carro si detecta un obstáculo a menos de 10cm, y avanzar en caso contrario.
- Si se detecta un 1 en el pin 8 encender un LED amarillo por 800ms.

Tenga en cuenta las siguientes recomendaciones antes de iniciar la programación:

- Si va a usar el monitor serial, recuerde que es indispensable tener conectado el Arduino al computador o al dispositivo donde tenga instalada la aplicación Arduino.
- El Arduino solo se puede alimentar de 3 formas diferentes y recuerde que NO debe conectar más de una fuente de alimentación a la vez porque puede dañar la tarjeta:

  - Fuente de alimentación 1: Con el cable del computador.
  - Fuente de alimentación 2: Con la batería de 9V.
  - Fuente de alimentación 3: Con una batería externa de 6V conectada al pin Vin y GND.

Para resolver cualquier duda o inquietud sobre la programación, el correcto uso de la tarjeta Arduino, o la conexión de los elementos electrónicos, puede remitirse al Manual del Kit de Ingeniería STEM en la sección de "Software" o "Hardware" según corresponda.

- Dirección del Manual del Kit de Ingeniería STEM en el Gestor de Contenidos: ***Repositorio Técnico NST. Fase II – Manuales***

- Si presenta dificultades para programar el algoritmo propuesto, puede descargarlo del Gestor de Contenidos en la siguiente dirección. Recuerde que estos algoritmos debe abrirlos en la aplicación de Arduino tanto para verlos como para editarlos y descargarlos en la tarjeta:
- ***Repositorio Recursos Rutas CPE. Fase II – Ruta 11–Sesión6- Arduino***

En la siguiente imagen se muestra un esquema de conexión de la tarjeta Arduino a los elementos electrónicos adicionales requeridos para esta aplicación. Conecte todo tal y como se muestra en la imagen para evitar posibles fallas:

- Pin P1 del micro:bit receptor en el pin 8 del Arduino.
- GND del micro:bit conectado con GND del Arduino. Hacer uso de la protoboard para esta conexión tal y como se muestra en la imagen.
- Alimentar el micro:bit receptor con su portapilas y las pilas AAA.
- Conectar el ultrasonido:

  - VCC en el pin de voltaje del Arduino. Hacer uso de la protoboard para esta conexión tal y como se muestra en la imagen.
  - GND en el pin de tierra del Arduino. Hacer uso de la protoboard para esta conexión tal y como se muestra en la imagen.
  - Trig en el pin 2 del Arduino.
  - Echo en el pin 3 del Arduino.

- Conectar el Driver para motor:

  - Conectar el motor 1 en OUT1 y OUT2, y el motor 2 en OUT3 y OUT4.
  - Conectar las entradas IN del driver de la siguiente manera:
    - IN1 con el pin 4 del Arduino.
    - IN2 con el pin 5 del Arduino.
    - IN3 con el pin 6 del Arduino.
    - IN4 con el pin 7 del Arduino.
- Conectar una pila de 9V en la entrada 12V y GND del driver, asegurándote

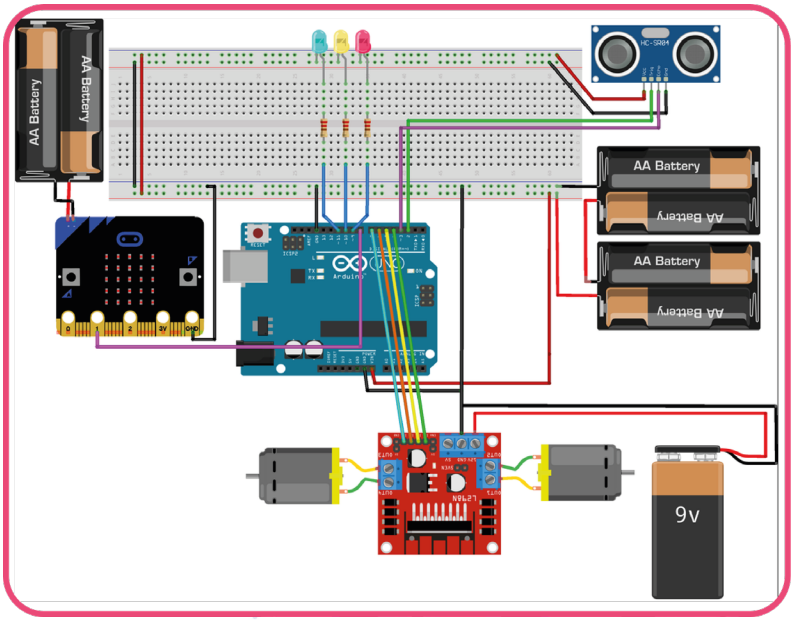
de conectar el negativo de la pila (-) en GND y el positivo (+) en 12V.

- Conectar los LEDs de colores:

  - Conectar el pin 9 a una resistencia de 220 ohmios, la resistencia al pin positivo del LED rojo (pata más larga) y el pin negativo del LED rojo (pata más corta) a la tierra del Arduino. Hacer uso de la protoboard para esta conexión tal y como se muestra en la imagen.
  - Conectar el pin 10 a una resistencia de 220 ohmios, la resistencia al pin positivo del LED amarillo (pata más larga) y el pin negativo del LED amarillo (pata más corta) a la tierra del Arduino. Hacer uso de la protoboard para esta conexión tal y como se muestra en la imagen.
  - Conectar el pin 11 a una resistencia de 220 ohmios, la resistencia al pin positivo del LED azul (pata más larga) y el pin negativo del LED azul (pata más corta) a la tierra del Arduino. Hacer uso de la protoboard para esta conexión tal y como se muestra en la imagen.

- Alimentación del Arduino con pilas externas:
  - Conecta las 2 pilas de litio en serie, es decir, con un cable macho-macho conecta el pin negativo (negro) de una pila con el pin positivo (rojo) de la segunda (ver imagen “Detalle de conexión de las pilas de litio de 3 voltios en serie”).
  - **¡NO ALIMENTE EL ARDUINO CON EL COMPUTADOR NI LA PILA DE 9V!** En esta sesión, la alimentación del Arduino se hace a través de las pilas de litio en serie, conectadas al Vin y GND del Arduino.
- Conecta la tierra (GND) del Arduino con:
  - La tierra de la protoboard
  - La tierra del micro:bit
  - La tierra de la pila de 9V
  - La tierra del conjunto de 2 pilas de litio en serie.

Nota : Es muy importante que las tierras estén conectadas tal y como se acaba de mencionar, ya que si alguna de ellas no está conectada, el programa presentará fallas. De igual manera, asegúrate de usar correctamente las fuentes de alimentación, ya que el carro tiene 3 fuentes: las pilas AAA del micro:bit, la batería de 9V y las dos pilas de 3V en serie.

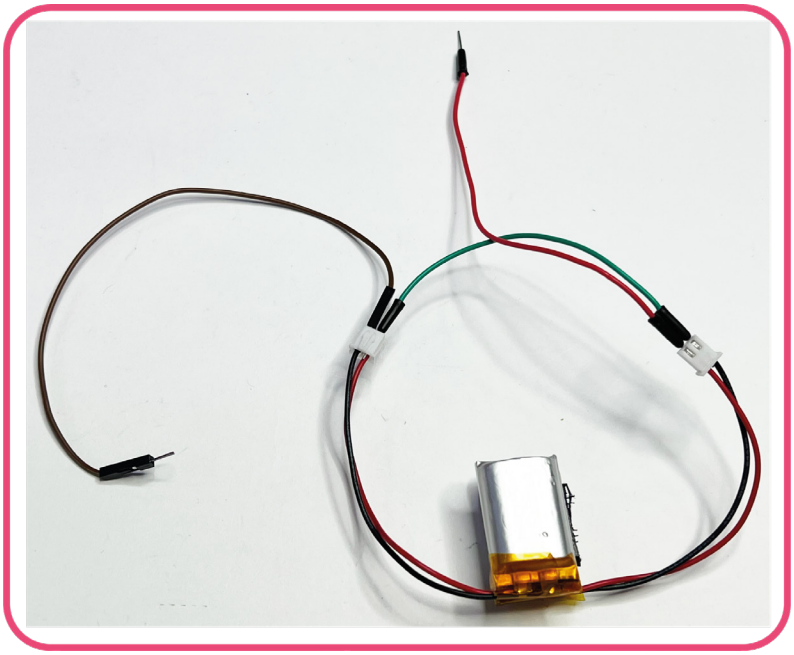


Nota : ten en cuenta que las pilas AA que se muestran en la imagen, conectadas al puerto Vin del Arduino, representan las pilas de litio de 3V.

Detalle de conexión de las pilas de litio de 3 voltios en serie:

Mira con cuidado cómo están conectados los cables antes de unir las pilas al prototipo.

A continuación, se presentan los pasos necesarios para programar el algoritmo de esta sesión:



Nota : Una vez que los estudiantes finalicen todas las conexiones electrónicas de los micro:bits y el Arduino, deben ubicar los elementos electrónicos en la estructura del robot móvil y el peaje que hayan construido. Recuérdele a los estudiantes verificar a detalle que nada se desconecte al momento de integrar los elementos.

1. Abra la aplicación Arduino. Se creará automáticamente un nuevo proyecto. De clic en guardar y ponga el nombre “Detector_Obstaculos”.
2. Programe la definición de los pines a usar. Escríbalo antes de “void setup”.
3. Defina dentro de “void setup” cuales pines serán de entrada y salida,



establezca la comunicación serial y apague los pines del motor para asegurar que el carro inicie detenido.

```
// ULTRASONIDO PINES
const int Trigger = 2;   //Pin digital 2 para el Trigger del sensor
const int Echo = 3;   //Pin digital 3 para el Echo del sensor

//MOTORES CARRO PINES
int MotorD1 = 4;
int MotorD2 = 5;
int MotorI1 = 6;
int MotorI2 = 7;

//LEDs
int R = 9;
int A = 10;
int B = 11;

//Pin microbit
int PinMicrobit = 8;
```

4. Escribe en “void loop” las siguientes tareas:

- Enviar un pulso de 10 micro segundos al Trigger el ultrasonido y esperar

```
void setup() {
  // ULTRASONIDO
  Serial.begin(9600);//iniciailzamos la comunicación
  pinMode(Trigger, OUTPUT); //pin como salida
  pinMode(Echo, INPUT);  //pin como entrada
  digitalWrite(Trigger, LOW);//Inicializamos el pin con 0
  // MOTORES CARRO
  pinMode(MotorD1, OUTPUT);
  pinMode(MotorD2, OUTPUT);
  pinMode(MotorI1, OUTPUT);
  pinMode(MotorI2, OUTPUT);
    digitalWrite(MotorD1, LOW);
    digitalWrite(MotorD2, LOW);
    digitalWrite(MotorI1, LOW);
    digitalWrite(MotorI2, LOW);
  // Pin micro:bit
  pinMode(PinMicrobit, INPUT);
  // Pines LED RGB
  pinMode(R, OUTPUT);
  pinMode(A, OUTPUT);
  pinMode(B, OUTPUT);
}
```

la respuesta recibida por el Echo para medir la distancia en centímetros (usando la fórmula de distancia del ultrasonido vista en el marco teórico).

- Si la distancia registrada por el sensor es menor a 10, entonces detener el carro (apagar todas las entradas de los motores), sino hacer que el carro avance (encender un pin de cada motor). Cuando el carro esté detenido en-

```
void loop()
{

  long t; //timepo que demora en llegar el eco
  long d; //distancia en centimetros

  digitalWrite(Trigger, HIGH);
  delayMicroseconds(10);          //Enviamos un pulso de 10us
  digitalWrite(Trigger, LOW);

  t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
  d = t/59;             //escalamos el tiempo a una distancia en cm

  Serial.print("Distancia: ");
  Serial.print(d);      //Enviamos serialmente el valor de la distancia
  Serial.print("cm");
  Serial.println();
  delay(100);          //Hacemos una pausa de 100ms
```

cender el LED rojo, y cuando esté avanzando encender el LED azul.
- Cuando se detecte un 1 en el pin 8 del Arduino, encender el LED amarillo por 800 milisegundos. Cerrar el corchete de "void loop".

```
if (d < 10) {
  //Frenar
  digitalWrite(MotorD1, LOW);
  digitalWrite(MotorD2, LOW);
  digitalWrite(MotorI1, LOW);
  digitalWrite(MotorI2, LOW);

  //RGB color rojo - detenido
  digitalWrite(R, 1);
  digitalWrite(A, 0);
  digitalWrite(B, 0);
}
else {
  //Avanzar
  digitalWrite(MotorD1, HIGH);
  digitalWrite(MotorD2, LOW);
  digitalWrite(MotorI1, LOW);
  digitalWrite(MotorI2, HIGH);

  //RGB color verde - avanzar
  digitalWrite(R, 0);
  digitalWrite(A, 0);
  digitalWrite(B, 1);
}
```

5. En este punto el algoritmo queda terminado. Acceda al Gestor de Contenidos para descargar el programa y ábralo en la aplicación de Arduino para ver el resultado final.

```
  if (digitalRead(PinMicrobit) == 1) {
    //RGB color amarillo - pagado
    digitalWrite(R, 0);
    digitalWrite(A, 1);
    digitalWrite(B, 0);
    delay(800);
  }
}
```

4.7 Simulación

Esta actividad no se puede simular. El docente debe descargar el algoritmo directamente en las tarjetas micro:bit y Arduino para probar el programa. Asegúrese de programar cada tarjeta con el algoritmo correspondiente.

4.8 Monitor Interactivo

¿Qué es el Monitor Interactivo?

El Monitor Interactivo es una pantalla táctil que está conectada al Gestor de Contenidos y permite mostrarle a los estudiantes los archivos almacenados del Gestor, los cuales incluyen el PDF de la sesión, videos relacionados, imágenes y demás. Adicionalmente, su componente táctil le permite al docente invitar a los estudiantes a interactuar con los contenidos presentados en el monitor, editarlos, moverlos, rayarlos o dibujar nuevos elementos para expresar sus ideas, resolver un problema o mostrar algo relevante a la clase.

En esta sesión, se usará el monitor interactivo para:

- Mostrar los videos de apoyo del marco teórico sobre el sensor de ultrasonido, el driver para motor y los peajes.
- Solicitarle a los estudiantes que dibujen todas las conexiones electrónicas que hicieron en el prototipo e identifiquen y marquen cada uno de los elementos.
- Dibujen planos sencillos en los que muestren dónde y cómo ubicaron cada uno de los elementos que componen el prototipo.

5. Reflexiona, evalúa y concluye

5.1 Reflexión y evaluación

Se sugiere que el docente realice las siguientes actividades de evaluación y conclusión con los estudiantes:

Realice las siguientes preguntas:

- ¿Cuál es la importancia de los peajes en la sociedad?

- ¿Consideras que automatizar el sistema de pago es una solución efectiva para el problema de atascamiento en los peajes?
- ¿Qué otras soluciones se pueden proponer para reducir el tiempo de espera en los peajes?
- ¿Por qué son importantes los sistemas de comunicación en las tarjetas programables?
- ¿De qué otra manera se podría integrar el micro:bit con la tarjeta Arduino?
- Haciendo uso del monitor interactivo, solicite a los estudiantes que escriban:
- Sus conclusiones sobre el problema planteado y la solución construida.
- Las dificultades que tuvieron durante el desarrollo de la actividad.

6. Conclusión

En esta sesión, los estudiantes tuvieron la oportunidad de integrar dos micro:bits con la tarjeta programable Arduino y diferentes dispositivos electrónicos para construir un prototipo tecnológico capaz de automatizar un sistema de pago en un peaje, controlando de forma automática el movimiento de la talanquera según la presencia de vehículos en el peaje y el estado de la transacción del pago electrónico.

Esta propuesta de solución le permitió a los estudiantes integrar sus conocimientos sobre pensamiento computacional y programación, con situaciones de la vida real y problemas socio-económicos del país, en busca de mejorar la calidad de vida de las personas que entran y salen regularmente de la ciudad capital.

Finalmente, los estudiantes tuvieron la oportunidad de enfrentarse a un problema de la vida real, proponer y construir una solución, y probarla en un prototipo para comprender la lógica de funcionamiento y de programación requerida para dar solución a un problema.

7. Bibliografía

- Sánchez, J. (5 de octubre de 2022). Peaje. Economipedia.com. ***https://eco-***

nomipedia.com/definiciones/peaje.html

- Keyence. (5 de octubre de 2022). ¿Qué es un sensor ultrasónico? Keyence. com.mx. ***https://www.keyence.com.mx/ss/products/sensor/sensorbasics/ultrasonic/info/***

- Dademuchconnection. (26 de abril de 2018). Driver de motor DC – Electrónica de potencia. Dademuch.com. ***https://dademuch.com/2018/04/26/driver-de-motor-dc-electronica-de-potencia/***

- Naylamp Mechatronichs. (5 de octubre de 2022). DRIVER PUENTE H L298N 2A. NaylampMechatronics.com. ***https://naylampmechatronics.com/drivers/11-driver-puente-h-l298n.html***