

# Taller Smart Robots Challenge

## *Objetivo*

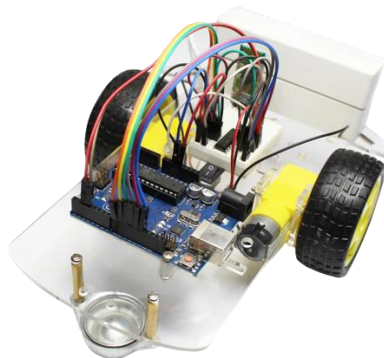
Controlar un carrito usando señas con las manos. Para esto es necesario crear un modelo de clasificación de imágenes para diferenciar las señas que representarán los movimientos del carrito (Adelante, Atrás, Izquierda, Derecha y Parar). Este modelo se usará en Python, el cual se comunica por medio de Bluetooth a un Arduino que controla los motores del carrito para hacer los movimientos que se obtienen de un video en vivo haciendo las señas con las que se entrenó el modelo.

## *Requisitos*

- Computadora con Windows, Cámara e Internet

## *Carrito*

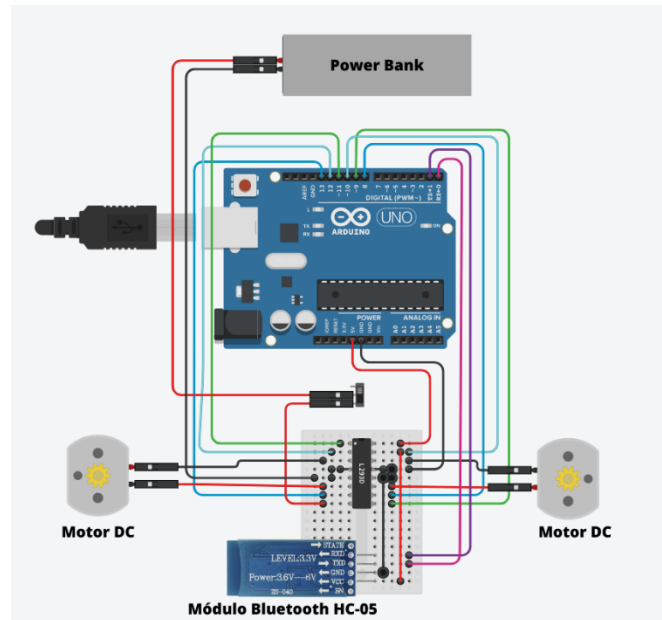
El carrito usado para el taller se puede ver en la siguiente imagen, el link a la página del producto se puede encontrar dando click a la misma.



Dentro de la misma página se puede encontrar un [tutorial](#) para armar el kit del carrito, las consideraciones más importantes al momento del armado es la posición del Arduino y el proto, así como la orientación de los motores para su correcto funcionamiento.

Para la conexión de los circuitos se debe seguir el siguiente diagrama, ya que fue modificado del original que se tiene en el tutorial. Es importante tomar en cuenta que el módulo bluetooth debe de alimentarse con el Arduino y no con la alimentación de los motores, ya que, al requerir mucha corriente para iniciar, las baterías que

vienen con el kit no pueden suministrar suficiente corriente al módulo y este se reinicia, desconectándose de la aplicación.



## *Pasos para controlar el carrito*

### 1. Teachable Machine (Aquí se crea el modelo de clasificación de imágenes)

- Entrar a [Teachable Machine](#), dar click en “Get started” -> “Image Project” -> “Standard image model”
- Crear y renombrar 5 clases (Adelante, Atrás, Izquierda, Derecha y Parar) usando el lápiz al lado del nombre para editarlo, y dando click en “Add a class” al final de las clases creadas para añadir más.
- Usar la webcam para tomar fotos de cada clase, se pedirá permiso para que la página pueda acceder a la cámara. Se recomienda tomar al menos 150 imágenes por clase usando el botón “Hold to record”, así como mover la mano haciendo la misma seña para dar variedad a la clase, alejando y acercando, además de moverse por todo el recuadro. Importante: se clasifica todo el cuadrado, por lo que el fondo puede afectar.
- Dar a train model para empezar el entrenamiento del modelo. Puede tardar en empezar diciendo que la página está tardando en responder, es importante esperar, una vez que lo haga aparecerá una barra de progreso. No se recomienda mover las opciones avanzadas.
- Una vez terminado el entrenamiento se puede probar el modelo con la webcam en la parte de “Preview”, aparecerán unas barras que se llenan de

acuerdo con la señal que se cree que se está haciendo. Es importante considerar que el modelo siempre está tratando de clasificar por lo que cuando se le muestre algo diferente de las imágenes con las que se entrenó va a clasificar de manera errónea aun cuando crea estar 100% seguro.

f) Exportar el modelo dando click al lado de "Preview". Ir a la opción del medio (Tensorflow) y descargarlo en Keras.

g) Descomprimir el archivo descargado. Importante saber dónde se guardaron los archivos resultantes pues se utilizarán en la aplicación de Python.

## 2. Python (Aquí se hace la clasificación para mandar los movimientos a Arduino)

a) Descargar Python 3.10 (Última versión probada). Asegurarse de marcar la opción "Add Python to Path" antes de iniciar la instalación y la opción "Disable path length limit" al finalizar la instalación para evitar problemas con las librerías después.

b) Abrir la terminal de Windows (PowerShell o CMD).

c) Descargar las librerías tkinter, pillow, opencv, tensorflow, numpy y pyserial utilizando los siguientes comandos:

- i. `python -m pip install tk`
- ii. `python -m pip install pillow`
- iii. `python -m pip install opencv-python`
- iv. `python -m pip install tensorflow`
- v. `python -m pip install imutils`
- vi. `python -m pip install numpy`
- vii. `python -m pip install pyserial`

*\*En caso de no funcionar intentar con "py" en lugar de "python"*

d) Ejecutar el programa SmartRobotsChallenge.py con el comando "python SmartRobotsChallenge.py" o abrir directamente con Python. Se abrirá una ventana con la aplicación para poder hacer la clasificación, así como mandar los movimientos al carrito.

e) Cargar el archivo de labels generado con Teachable Machine, una vez cargado aparecerán los nombres de las clases al lado del botón, estos se usarán para indicar que clase se está clasificando.

f) Cargar el archivo del modelo generado con Teachable Machine (extensión .h5), este archivo se usa para hacer la clasificación de imágenes con el modelo entrenado previamente. Tarda en cargar un poco, por lo que la

aplicación puede trabarse, una vez lo haga el nombre del modelo aparecerá indicado al lado del botón.

g) Encender la cámara. Debajo de los botones para cargar los archivos anteriores se tienen dos botones para encender y apagar el video, por defecto se toma la primera cámara que tenga la computadora. Una vez encendido el video y se cargaron ambos archivos correctamente se puede ver la predicción de la seña a la derecha junto con el porcentaje de la seguridad de que sea esa seña.

h) La tolerancia se puede ajustar para que la seña se mande a Arduino solo si su porcentaje es mayor al de la tolerancia (Se muestra en verde cuando si se manda y en rojo cuando no).

### 3. Arduino (Control de los motores para el movimiento del carrito)

a) Descargar Arduino IDE 1.8.19 (Última versión probada).

b) Abrir el archivo SmartRobotsChallenge.ino, este código ya cuenta con funciones hechas para no tener que controlar directamente el puente H, sino que se usan como movimientos que va a hacer el carrito, especificando la velocidad a la que se quiere mover los motores.

c) Editar el final del código siguiendo el ejemplo. Usar condiciones if comparando la variable "opcion" con el número que aparece en el archivo labels.txt y usar la función del movimiento que corresponda, entre paréntesis se pone la velocidad de 0 a 255 excepto en la función Parar().

d) Por ejemplo si en el archivo labels dice "2 Izquierda" entonces se debe poner `if (option == '2'){Izquierda(255);}`

e) En Herramientas seleccionar la tarjeta "Arduino UNO" y el puerto correspondiente.

f) Compilar y subir programa al Arduino.

\*Se puede usar el archivo SOLUCION-SmartRobotsChallenge.ino como referencia para saber como debía quedar el código y subir este mismo que ya ha sido probado para evitar errores.

### 4. Probar todo junto

En el programa de Python:

- a) Dar al botón de refrescar para mostrar las posibles conexiones que aparecerán debajo de este, en caso de no encontrar ninguna comunicación serial por el puerto COM, no se mostrará nada.
- b) Seleccionar el puerto que corresponde al Bluetooth (Inalámbrico) o al Arduino (Alámbrico) y dar a Conectar. Para que aparezca el puerto del Bluetooth es necesario primero enlazar el módulo con la computadora, el nombre por defecto es HC-05 y la contraseña es “1234”, una vez enlazado en la aplicación de Python aparecerán dos puertos, intentar conectarse con ambos, si se conecta con el correcto el módulo empezará a parpadear de forma más lenta, en caso contrario probar con el otro puerto. En el caso de hacer la conexión directa con el cable USB se debe usar el mismo puerto que se usó para subir el programa desde Arduino. En cualquiera de los casos el programa tiene un delay para asegurar la conexión por lo que se trabará durante unos segundos. Una vez conectado se muestra un mensaje indicándolo, en caso de que se desconecte marcará el mensaje “Conexión fallida” o “Sin conexión” por lo cual será necesario dar nuevamente al botón de conectar.
- c) Una vez que se tenga conectado el Arduino y la cámara este clasificando las imágenes, los movimientos se empezarán a mandar al Arduino.
- d) Si el programa de Arduino es correcto, el carrito se moverá de acuerdo con la señal que se haga.

### *Consideraciones importantes para dar el taller:*

- ✓ Cualquiera puede acceder a la página de Teachable Machine para generar su modelo, sin importar el sistema operativo, solo se necesita cámara y conexión a internet.
- ✓ Se recomienda tener cargada la aplicación de Python ya que tarda en abrir debido a las librerías que se usan.
- ✓ Es importante cargar bien los power banks (no todas las que venían con el kit funcionan correctamente) para que el bluetooth y los motores funcionen correctamente.
- ✓ Si se desconecta el bluetooth este comenzará a parpadear rápidamente, solo hay que dar en desconectar y volver a conectar, sin embargo, si el problema persiste de manera muy continua puede deberse a que la batería no le suministra suficiente energía.
- ✓ Se recomienda renombrar los módulos bluetooth para poder identificarlos de manera más rápida y así asignar cada uno con una computadora diferente. Se puede seguir el siguiente [tutorial](#) (Parte 4) para hacerlo.
- ✓ Puede que los jumpers se rompan internamente provocando que una o ambas llantas dejen de funcionar en un sentido o ambos, para comprobar su

correcto funcionamiento se recomienda usar el archivo `test_motors.ino`, el cual mandará todos los movimientos al carrito de forma consecutiva, una vez se identifique el motor se deberá checar la continuidad de los jumpers para identificar el que se debe cambiar.

- ✓ Se recomienda usar imágenes impresas en hojas de papel para hacer el modelo que controle el carrito, ya que las hojas tienen una imagen y fondo fijos haciendo más fácil su identificación por parte del modelo, se sugiere entrenar este modelo previamente y cambiar la solución para que se adapte al orden de los movimientos.
- ✓ Se puede usar cualquier cosa para crear el modelo de Teachable Machine, como piedra papel o tijera, caras, objetos, etc. por lo que se sugiere dejar libertad a los alumnos para explorar esta herramienta por su cuenta y luego usar el modelo mencionado en el punto anterior.

### *Links adicionales:*

- Se tiene una [presentación](#) en la cual se da una pequeña introducción a machine learning de manera breve y concisa, además de que es importante usar ejemplos, posteriormente se va explicando cómo usar Teachable Machine, así como el resto de pasos a seguir (sin entrar en detalle).
- Todos los códigos mencionados en este manual se pueden encontrar en el siguiente [repositorio](#).