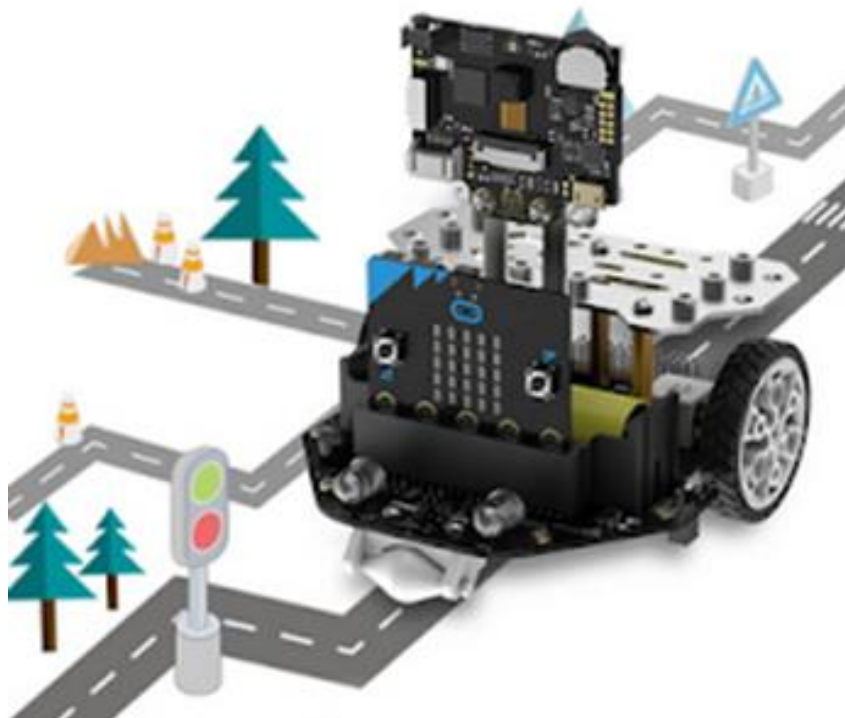


# Maqueen Plus + HuskyLens

## Tutorial con 6 proyectos



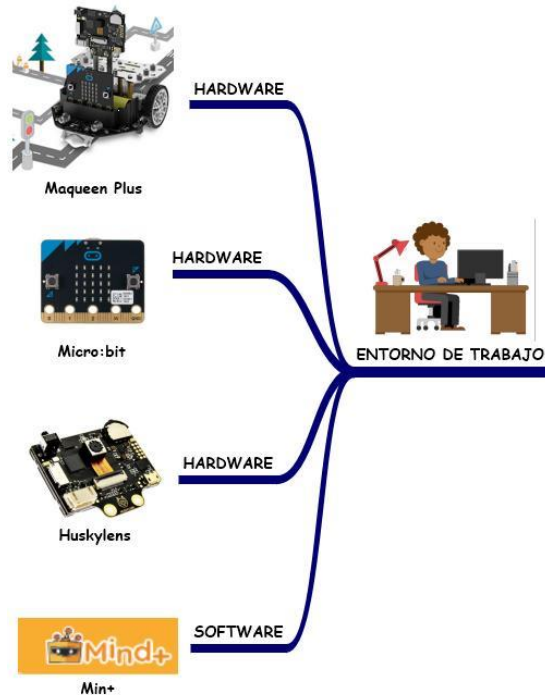
Prof. José Manuel Ruiz Gutiérrez

Agosto 2020 Ver: 1.0

## Objetivo de este libro.

Este libro se ha realizado pensando en difundir entre los usuarios de habla hispana el conocimiento de una herramienta muy útil para el mundo de la educación. Los textos están basados en los materiales que **DFRobot** pone a disposición de los usuarios en inglés y en chino.

La puesta en el mercado de la cámara “*inteligente*” **HuskyLens**, en mi opinión, establece un punto de inflexión en lo que se refiere al conocimiento práctico de las técnicas de “visión artificial” en el ámbito de la enseñanza y la experimentación.



Son numerosos los sistemas *Open Hardware* que hoy nos ofrece el mercado para poder usar la cámara, lo cual la hace más accesible para que se pueda integrar en sistemas como **Arduino**, **Micro:bit**. etc.

A lo largo de seis prácticas, bien escogidas, podemos evaluar las distintas potencialidades de la cámara. La programación de estas se realiza con la herramienta **Mind+** de la misma firma.

La tarjeta micro controladora seleccionada es **Micro:bit** que se conectará con la cámara a través del bus **I2C** que ya es bien conocido entre los usuarios de los sistemas basados en micro controladores. Además del sistema I2C, la conexión se puede realizar también a través de la **UART** de los

sistemas micro controladores.

En la configuración Hardware se ha añadido el robot recientemente aparecido de la familia Maqueen llamado **Maqueen Plus**.

### Fuentes de documentación utilizadas.

- [Wiki de Productos](#)
- [Tutorial para micro:Maqueen Plus y HUSKYLENS](#)
- [Descarga del software Mind+\(Scratch3.0\)](#)
- [Foro](#)
- [Comunidad](#)

### Agradecimientos

Deseo manifestar mi agradecimiento a DFRobot por la facilidad en el acceso a la documentación técnica y el uso de las imágenes.

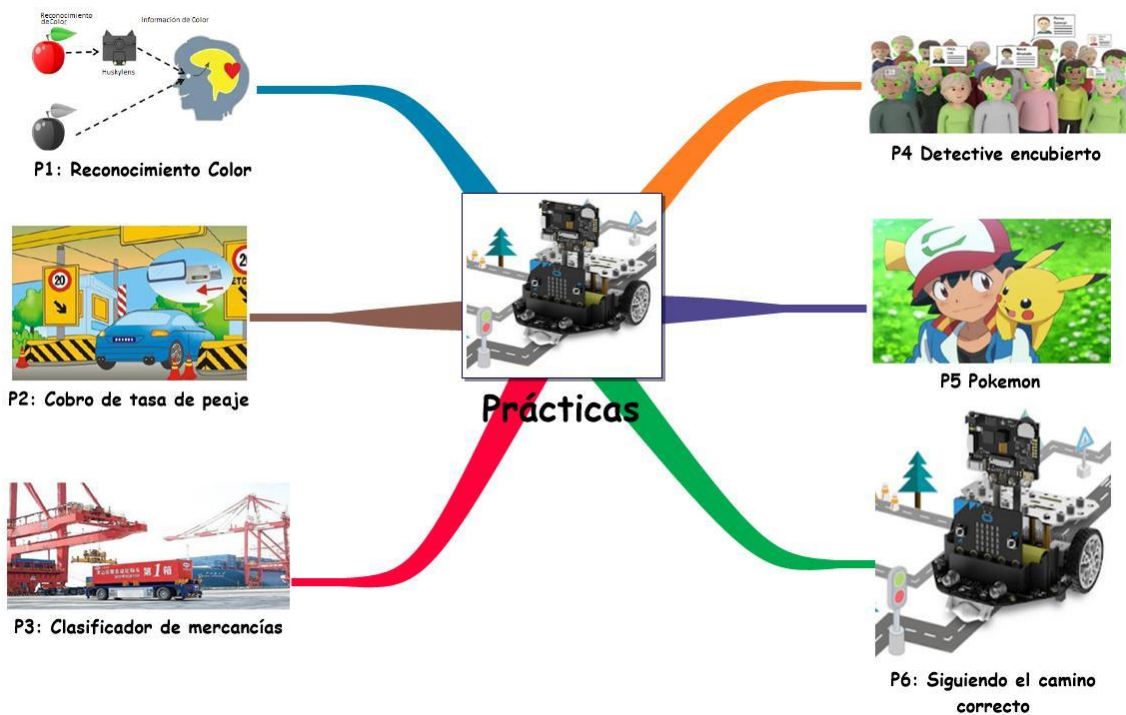
**Prof. José Manuel Ruiz Gutiérrez**

Agosto 2020

## Conceptos previos para abordar este libro y realizar sus prácticas.

Este manual está pensado para experimentar con la cámara **HuskyLens** y el robot **Maqueen Plus** utilizando el software **Mind+** para el desarrollo y generación de los programas que después cargaremos en la tarjeta **Micro:bit** que implementa el control del mencionado robot.

En el esquema de la siguiente figura se muestran las seis prácticas que realizaremos.



El desarrollo de cada práctica se hace mediante distintas tareas que se llevan a cabo para comprender los fundamentos teóricos y de las prestaciones de la cámara.

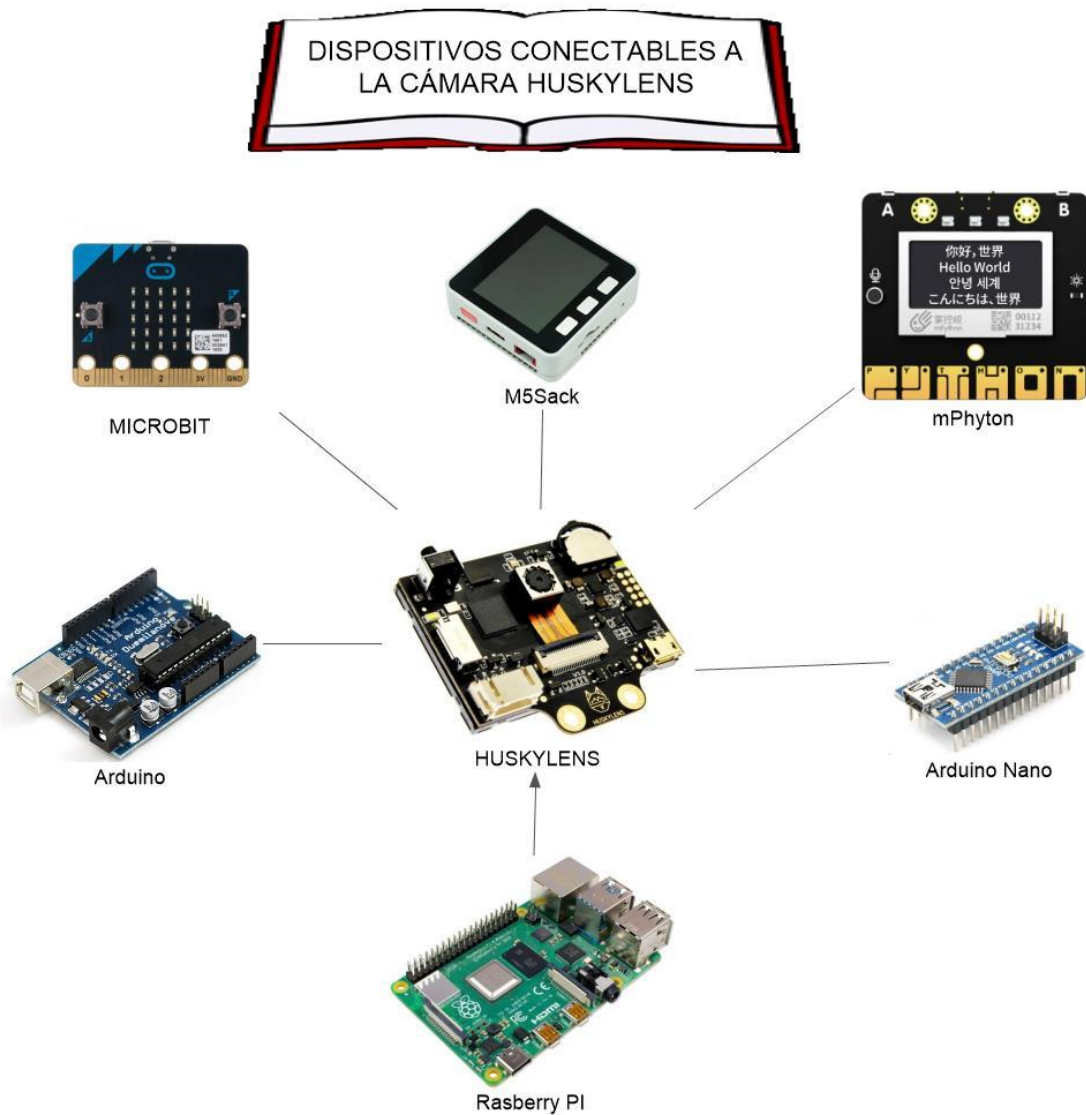
El contenido de cada práctica estará formado por:

- **Introducción:** Se explica lo que se pretende
- **Descripción del proyecto:** Se explica de manera concreta lo que se va a realizar
- **Lista de materiales:** Se enumeran los materiales a utilizar en la práctica.
- **Conceptos teóricos:** necesarios para entender la práctica
- **Tareas a realizar:** Acciones que se promueve realizar para la consecución final de la práctica.
- **Pasos a realizar:** Dentro de cada tarea se explican los pasos a realizar.
- **Realización práctica del proyecto:** se realiza la programación de los algoritmos del proyecto y se descarga la aplicación en la tarjeta Micro:bit
- **Resumen del proyecto:** Se resume el desarrollo del proyecto
- **Revisión del proyecto:** Se comentan aspectos relativos a la revisión real de la aplicación ya ejecutándose en la tarjeta Micro:bit

- Ampliaciones sugeridas al proyecto: Se sugieren ampliaciones al proyecto

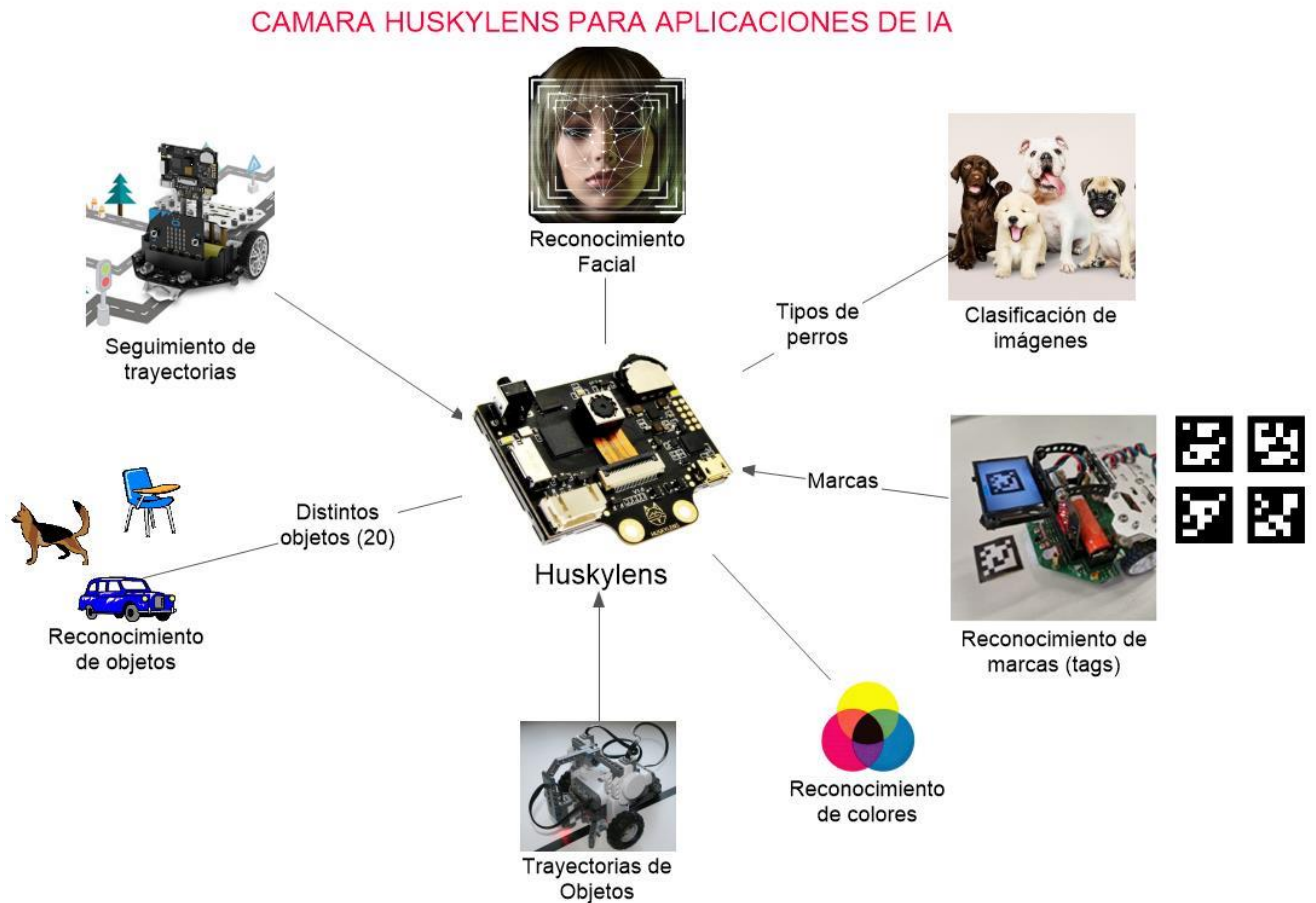
La intención de este manual es doble, por un lado la adquisición de destrezas con el uso del Maqueen Plus y por el otro desarrollar aplicaciones que nos permitan conocer mejor la cámara HuskyLens.

Se ha decidido trabajar con Micro:bit y Maqueen pero podríamos haber realizado estas prácticas con otra plataforma Open Hardware distinta a Micro:bit. En el cuadro siguiente se muestran posibles conexiones de la cámara HuskyLens a hardware:



## ¿Cuáles son las posibilidades que nos ofrece la cámara HuskyLens?

En la imagen siguiente se muestra un esquema de las posibilidades de la cámara. Como podemos observar son muchas las posibilidades, por lo que en mi opinión es muy válido este hardware para el desarrollo de prototipos educativos en torno a las STEM





¿Con que software programaremos nuestros proyectos?

Para poder implementar los algoritmos de cada uno de los proyectos se utilizará en este manual la herramienta **Mind+** de DFRobot.



Son muy numerosas las ventajas de esta poderosa herramienta y además incluye una extensa librería de funciones en forma de bloques que facilitan nuestra creatividad e innovación sobre los proyectos propuestos

## Proyecto1: IDENTIFICACION DE COLOR

### ¿La música está relacionada con el color?

Muchos músicos comparan la música con el color. El famoso músico estadounidense Maryon dijo una vez: "*El sonido es color audible, el color es música visible*". El uso de diferentes timbres en obras musicales es muy similar al uso de diferentes colores en obras de arte. Tanto el timbre como el color pueden dar a las personas sentimientos como claro, brillante, cálido o tenue.



Aunque no existe una teoría científica para demostrar la correlación entre la música y el color, podemos hacer divertidos dispositivos interactivos entre la música y el color. Lo más simple es conectar los siete tonos en la escala con siete colores para que cada color represente una escala. ¡Hagamos una partitura coloreada, a través de la cual se generó la maravillosa música!

### Descripción del proyecto

Este proyecto utiliza la función de reconocimiento de color de HuskyLens para reconocer bloques de color de diferentes colores. Usando Maqueen Plus para tocar diferentes escalas, su partitura musical no solo es atractiva sino también agradable de escuchar, lo que produce un efecto audiovisual absolutamente maravilloso.

### Lista de materiales:



## Conceptos teóricos

En la sociedad actual, la producción automatizada se ha convertido en la tendencia de desarrollo. Como los ojos de los "robots", la visión artificial es particularmente significativa.

Como una de las instrucciones técnicas más importantes, el reconocimiento del color ha experimentado muchas generaciones de actualización tecnológica. Este proyecto se basa en la función de reconocimiento de color del sensor HuskyLens.

### I ¿Qué es la identificación del color?

¿Qué es la identificación de color? Primero necesitamos saber de qué color es.

El color es un efecto visual sobre la luz generada por los ojos, el cerebro y nuestra experiencia de vida. La luz que vemos a simple vista es generada por ondas electromagnéticas con un rango de longitud de onda muy estrecho. Las ondas electromagnéticas con diferentes longitudes de onda muestran diferentes colores.

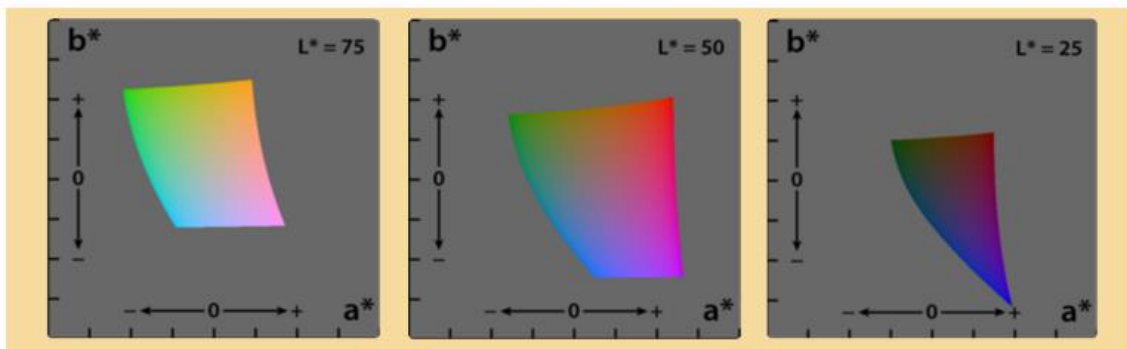


El reconocimiento de color se basa en atributos de color con diferente claridad para identificar y

### II El principio de funcionamiento del reconocimiento del color.

El reconocimiento de color se basa en el espacio de color Lab, donde la dimensión L representa la luminosidad, a y b representa la dimensión del valor de color opuesto y las coordenadas del espacio de color CIE XYZ basadas en compresión no lineal.

Comparando los parámetros Lab de los colores reconocidos y aprendidos, cuando los dos colores coinciden dentro de un cierto rango de error, se identifican como el mismo color.



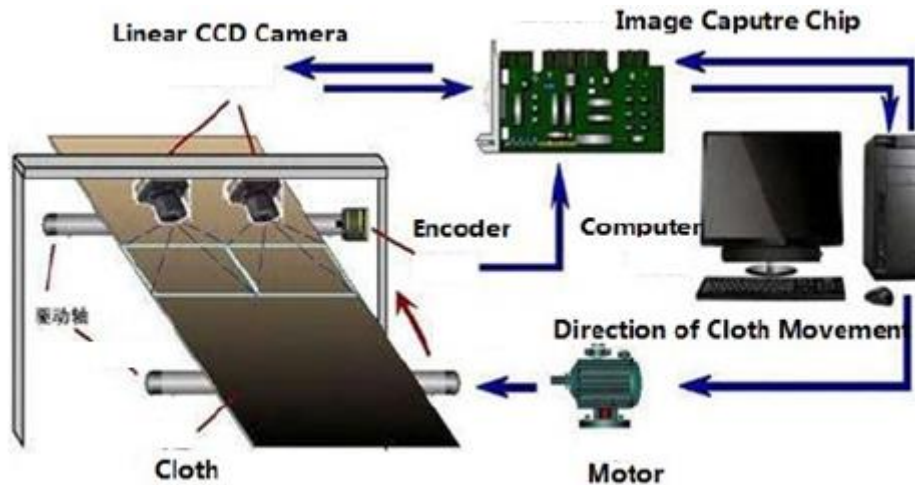
En nuestro uso habitual del reconocimiento de color, el tono y la saturación en el atributo de color del mismo módulo son fijos, pero la luminosidad cambiará bajo la influencia del



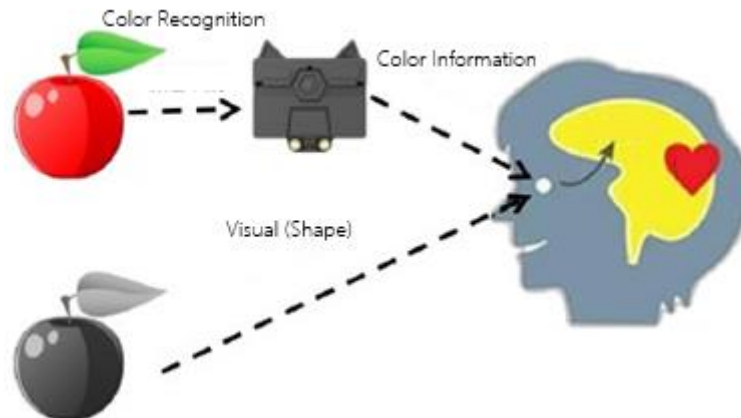
cambio de luz ambiental. Por lo tanto, al utilizar la función de reconocimiento de color de HuskyLens, debemos hacer todo lo posible para garantizar que la luz ambiental durante el aprendizaje y el reconocimiento sea coherente con el trabajo real.

### III. Campos de aplicación principales de reconocimiento de color

**Campo industrial:** el reconocimiento de color se usa ampliamente en el campo industrial, como la impresión, el revestimiento de pintura y el textil, para el monitoreo y calibración del color.



**Vida personal:** el uso como reconocimiento auxiliar para personas con debilidad de color o discapacidad visual puede mejorar su comprensión del color.

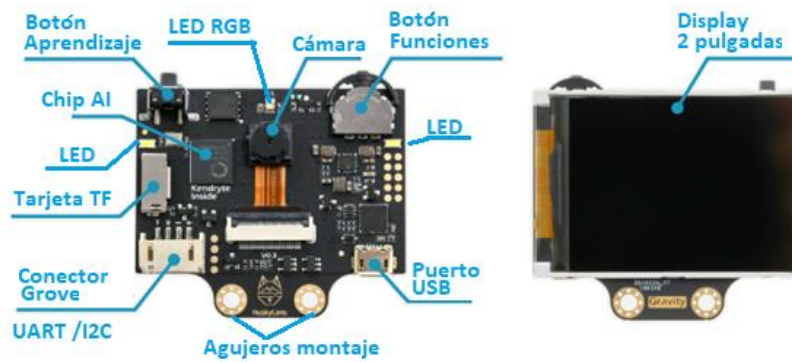


### IV. Demostración de la función de reconocimiento de color HuskyLens

La función de reconocimiento de color del sensor HuskyLens es utilizar el algoritmo incorporado, al aprender y grabar diferentes colores, la identificación de diferentes colores se puede identificar y retroalimentar a la placa base.

La configuración predeterminada en HuskyLens es aprender, reconocer y rastrear solo un color. Pero podemos cambiar la configuración predeterminada para que reconozca varios colores.

¿Cómo operarlo paso a paso? Primero, saca tu HuskyLens. Hagámoslo juntos.

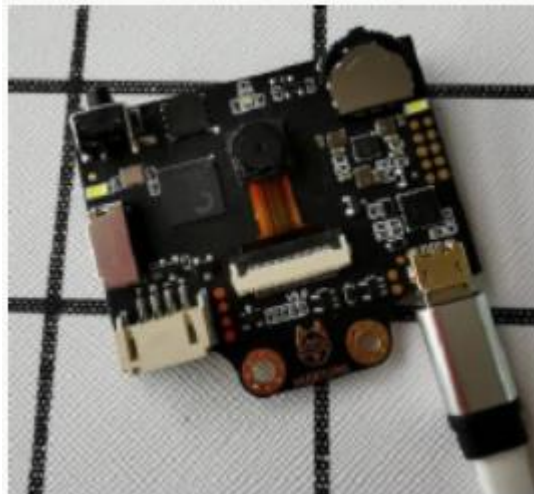


Para usuarios nuevos, consulte el sitio web de WIKI para la grabación de firmware y la configuración de idioma:

[http://wiki2.dfrobot.com.cn/HuskyLens\\_V1.0\\_SKU\\_SEN0305\\_SEN0336](http://wiki2.dfrobot.com.cn/HuskyLens_V1.0_SKU_SEN0305_SEN0336)

### **PASO 1 Encienda**

HuskyLens tiene su propio conector USB independiente, que se puede encender conectando el cable USB.



### **PASO 2 Operación y configuración - Aprenda múltiples**

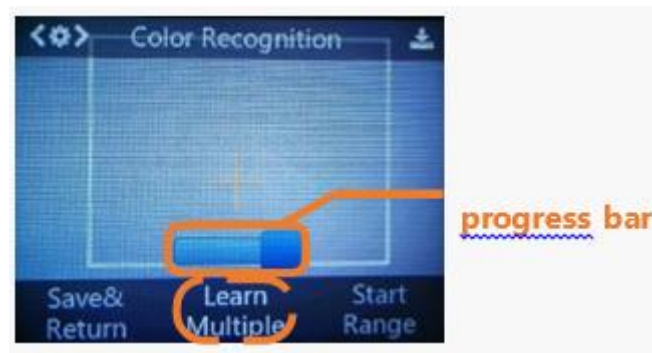
1. Marque el botón de función derecha o izquierda hasta que aparezca la palabra "Reconocimiento de color" en la parte superior de la pantalla.



2. Mantenga presionada la función botón para ingresar la configuración de parámetros de la función de reconocimiento de color.



3. Marque el botón de función hasta que aparezca "Aprender múltiples", luego presione brevemente el botón de función y marque hacia la derecha para activar el interruptor "Aprender múltiples", es decir, la barra de progreso se vuelve azul y el icono cuadrado en la barra de progreso se mueve hacia la derecha. Luego presione brevemente el botón de función para confirmar este parámetro.



4. Marque el botón de función hacia la izquierda hasta que aparezca "Guardar y volver". Y la pantalla muestra "¿Desea guardar los parámetros? "Seleccione" Sí "en forma predeterminada, ahora presione brevemente el botón de función para guardar los parámetros y regresar automáticamente.

De esta manera, terminamos la configuración de la función "Aprender múltiples"

### **PASO 3 Operación y configuración: aprendizaje y detección**

#### **1. Detección de color**

Apunte el icono "+" en el centro de la pantalla de HuskyLens al bloque de color de destino, y aparecerá un cuadro blanco en la pantalla, que selecciona el bloque de color de destino automáticamente. Ajuste el ángulo y la distancia de HuskyLens al bloque de color para que el cuadro blanco encuadre todo el bloque de color objetivo lo más lejos posible



#### **2. Aprendizaje de color**

Después de detectar el color, mantenga presionado el "**botón de aprendizaje**" para aprender el primer color y luego suelte el "**botón de aprendizaje**" para finalizar el aprendizaje. Aparecerá un mensaje en la pantalla: "¡Haga clic nuevamente para continuar! Haga clic en otro botón para finalizar".

Presione brevemente el "**botón de aprendizaje**" antes de que finalice la cuenta regresiva si desea aprender el próximo color. De lo contrario, presione brevemente el "botón de función" antes de que finalice la cuenta regresiva, o no presione ningún botón para que finalice la cuenta regresiva.

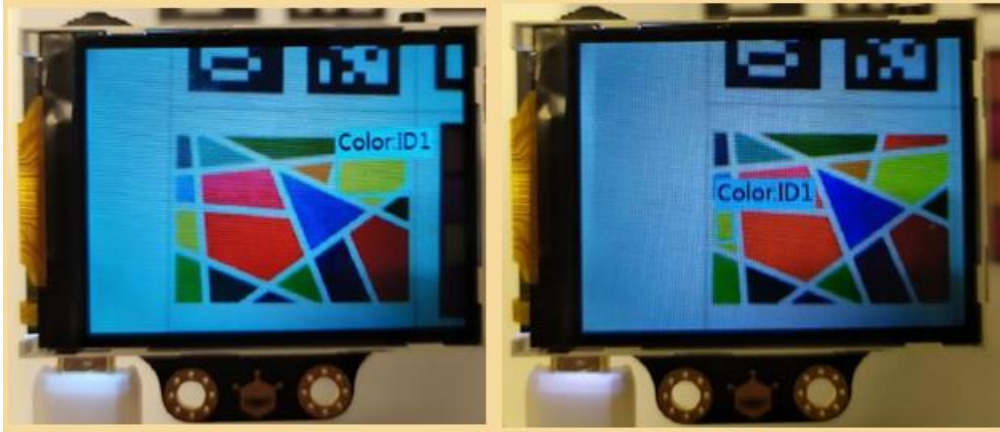
La identificación de color que muestra HuskyLens está en el mismo orden que el color de aprendizaje, es decir, la identificación se marcará como "**ID1**", "**ID2**", "**ID3**" y así sucesivamente. Y los colores del marco correspondientes a diferentes colores también son diferentes.

### 3. Aprendizaje del color



Cuando encuentre bloques de color iguales o similares, un marco de color con una ID se mostrará automáticamente en la pantalla, y el tamaño del marco de color es igual al tamaño de los bloques de color. El marco seguirá automáticamente el bloque de color. Se pueden identificar y rastrear diferentes colores al mismo tiempo, y los colores del marco correspondientes a diferentes colores también son diferentes.

Cuando aparecen varios bloques de color del mismo color en la imagen, solo se puede identificar un bloque de color a la vez.



\* El reconocimiento del color se ve muy afectado por la luz ambiental. A veces, HuskyLens puede identificar erróneamente colores similares. Intente mantener la luz ambiental constante y utilícela con iluminación moderada.

### Realización práctica del proyecto

¿Cómo se usa la función de reconocimiento de color HuskyLens? ¿Cómo podemos mapear los colores a escalas una por una? Divida todo el proyecto en varias tareas pequeñas y complete la notación musical numerada en color paso a paso.

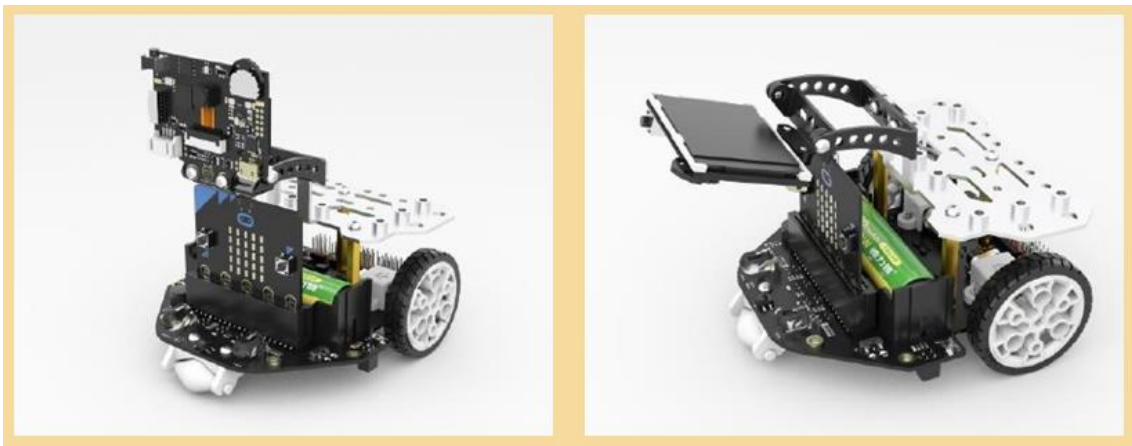
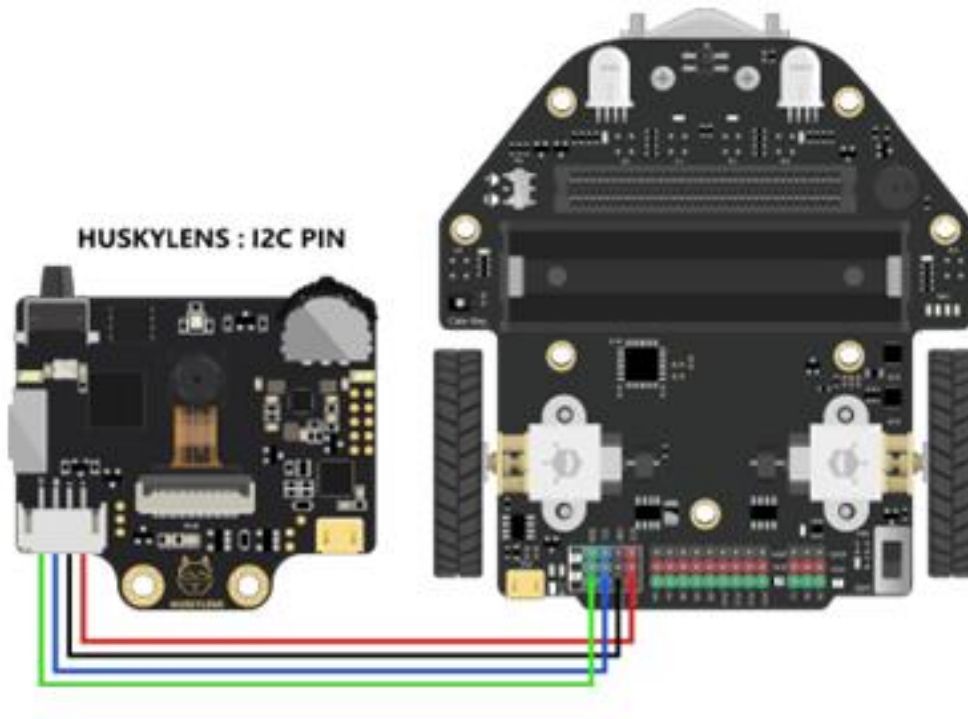
Este proyecto se dividirá en tres pasos para completar la tarea. Primero, aprenderemos a usar la función de reconocimiento de color de HuskyLens y generar el **ID** de color reconocido. Entonces podemos jugar las escalas de acuerdo con la identificación del color de salida.

### Tarea 1: Conozca el reconocimiento de color

#### 1. Configuración de hardware

Diagrama de conexión: el sensor HuskyLens utiliza un conector **IIC**. Por favor, preste atención a la secuencia del cable. No lo conecte de forma incorrecta o inversa.





**Diagrama de ensamblaje:** el sensor HuskyLens tiene su propia estructura de soporte, que puede ser fijada a Maqueen Plus mediante tornillos, y HuskyLens puede ajustarse varios ángulos.

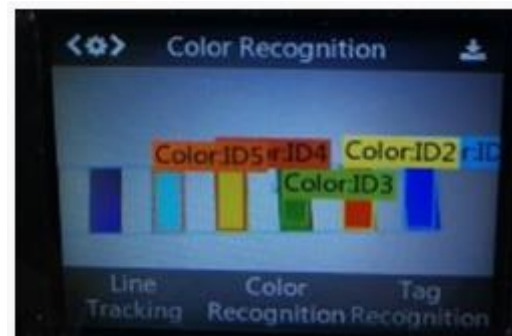
## 2. Diseño del programa

Aquí tomamos el sensor HuskyLens para aprender 3 colores como ejemplo, y emitimos ID de color a través del puerto serie para facilitar la visualización en tiempo real.

\* **Puerto serie:** el puerto serie es una forma de comunicación en tiempo real entre el ordenador y el hardware. Por ejemplo, en esta tarea, a través del puerto serie, los datos de HuskyLens en Maqueen Plus se pueden ver en tiempo real en el ordenador.

## PASO 1 Aprendizaje y reconocimiento

Antes de diseñar el programa, necesitamos el Sensor HuskyLens para aprender cada color. (Nota: Primero cambie a la función "Aprender múltiples")



## PASO 2 Configuración de Mind + Software

Software Open Mind + (Versión 163 o superior):

1. Cambie al "Modo de carga";



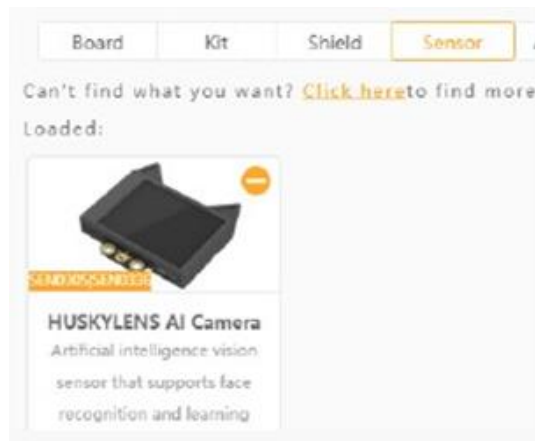
2. Haga clic en "Extensiones" y cargue "micro: bit" debajo de "Tablero";



3. Continuar haciendo clic y cargar "Edición doméstica de Maqueen Plus" bajo "Expansión";



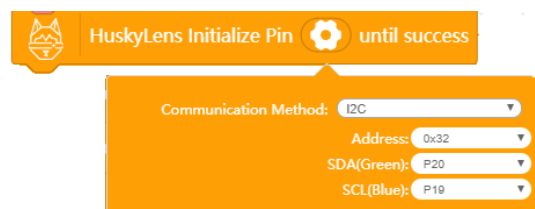
4. Haga clic y cargue "HuskyLens AI Cámara "debajo de" Sensor "para terminar.



### PASO 3 Instrucción Aprendizaje

Echemos un vistazo a algunas de las instrucciones principales.

1. Inicialización: colocada entre el inicio del programa principal y la ejecución del bucle, solo necesita ejecutarse una vez. Se puede seleccionar I2C o puerto serie, no es necesario cambiar la dirección I2C.

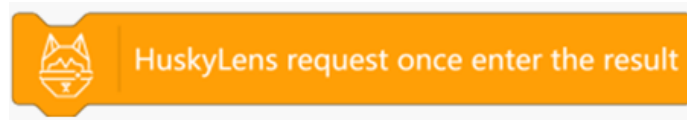


Nota: el lado de HuskyLens debe ajustar el "protocolo de salida" en la configuración para que sea coherente con el programa, de lo contrario, los datos no pueden ser rojos

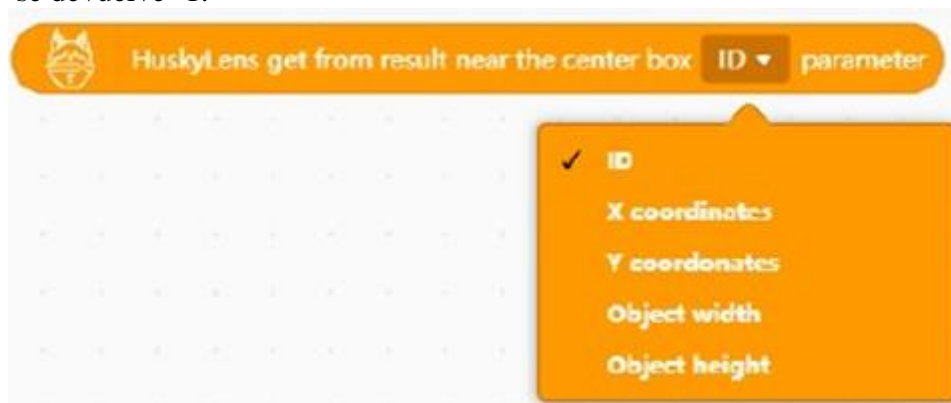
1. Algoritmos de conmutación: otros algoritmos pueden ser cambiado a en cualquier momento, y solo puede existir un algoritmo al mismo tiempo.  
Nota: El cambio de algoritmo lleva algo de tiempo.



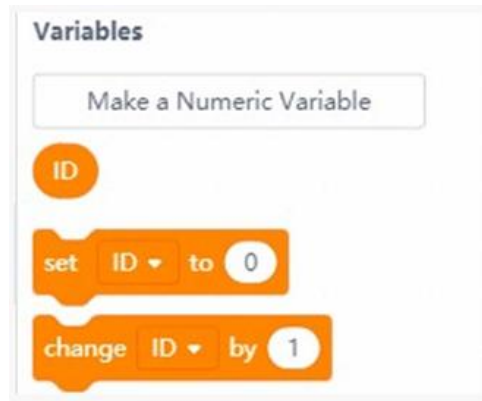
2. La placa principal solicita a HuskyLens que almacene los datos en el "**resultado**" una vez (datos almacenados en la memoria de la placa m, actualizados por cada solicitud), luego los datos se pueden obtener del "resultado", y se obtendrán los últimos datos en el "resultado" solo después de llamar a este módulo.



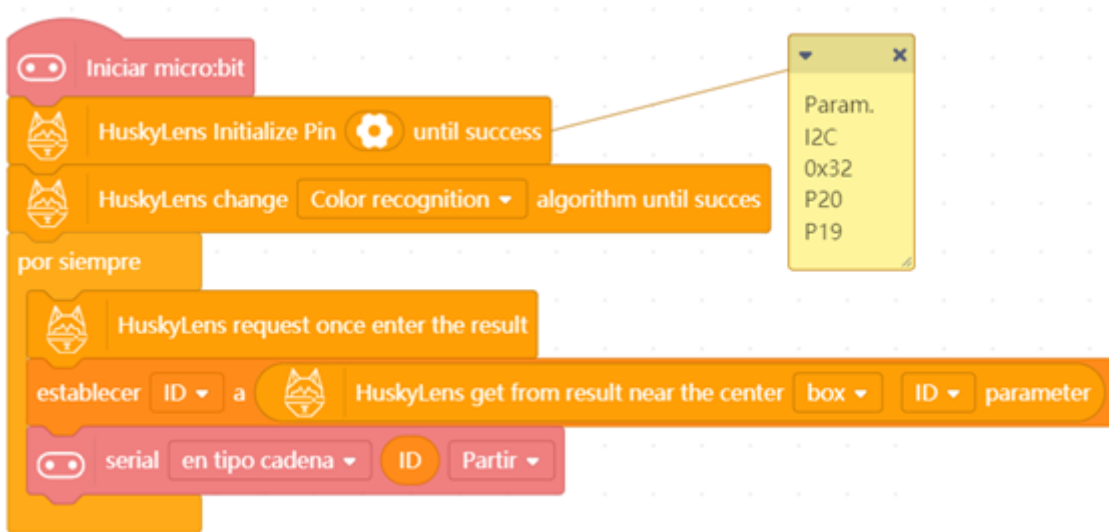
3. La información cerca del cuadro central en la interfaz actual se obtiene del "resultado" solicitado. El id de la caja que no se ha aprendido es 0, de lo contrario, se devuelve -1.



4. Variable: la variable se refiere a la cantidad de cambio, lo cual es conveniente para almacenar el número cambiado. Haga clic en "**Crear una variable numérica**" para crear una nueva variable. Puede establecer directamente el valor de la variable, así como aumentar o disminuir el valor de la variable3



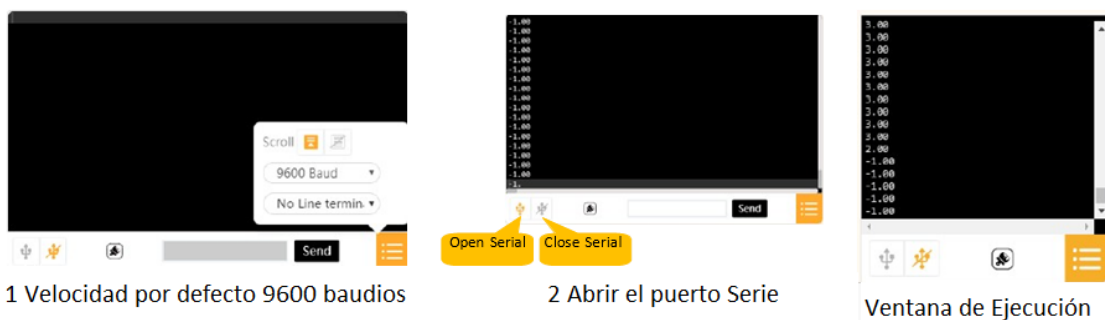
### 3. Programa Ejemplo



### 4. Resultados de la Ejecución

Cuando se utiliza un puerto serie, el ordenador y la placa principal deben conectarse mediante un cable USB. En Mind +, conecta el puerto COM correspondiente.

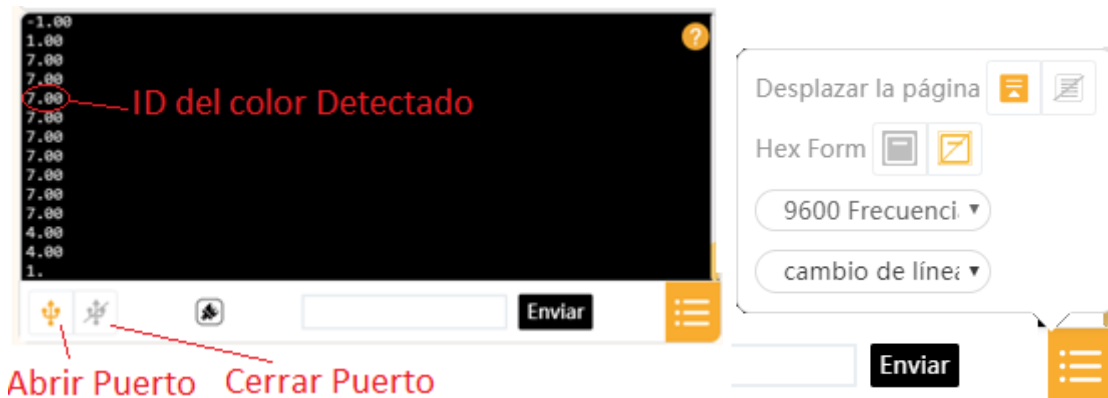
#### Método de operación del puerto serie:



Cuando el sensor HuskyLens ha aprendido tres colores, reconoce la identificación del color correspondiente y muestra el número correspondiente en el puerto serie. 1 denota el



primer color aprendido, 2 denota el segundo color aprendido, 3 denota el tercer color aprendido y -1 denota el color que no se ha aprendido



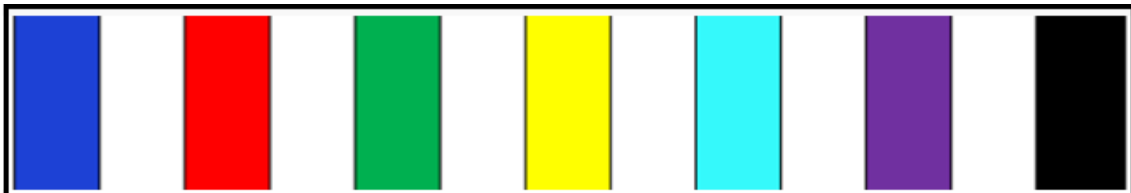
## Tarea 2: Definir escala para cada color

### 1. Diseño del Programa

#### PASO 1 Operación y configuración: aprendizaje y reconocimiento

Antes de diseñar el programa, necesitamos tener HuskyLens para aprender 7 colores. Para evitar falsos reconocimientos Además, se deben seleccionar 7 colores con una gran diferencia de color.

Los colores de ID1-ID7 se configuran respectivamente en azul, rojo, verde, amarillo, cian, morado y negro. Aprenda los colores en este orden para corresponder con los siguientes procedimientos. Después de aprender los 7 colores, HuskyLens dejó de aprender.

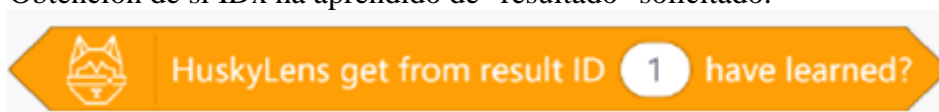


En combinación con las luces de Mcqueen Plus, identifique un color y deje que las luces muestren el color correspondiente. ¿Ver? ¿Es el efecto visual más glorioso?

#### PASO 2 Instrucción Aprendizaje

Echemos un vistazo a algunas de las instrucciones principales.

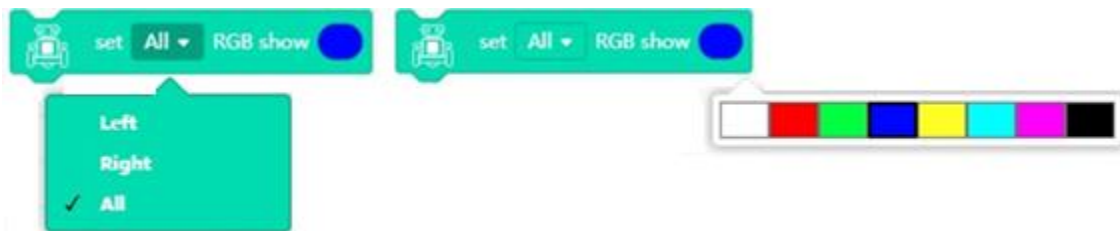
1. Obtención de si IDx ha aprendido de "resultado" solicitado.



- Obtención de si hay un cuadro o una flecha en el interfaz actual del "resultado" solicitado, i aprendido (id es mayor que 0) y no aprendido, si hay uno o más, devuelve 1.



- Ajuste las luces del robot a muestra diferentes colores, y el programa proporciona 8 colores.



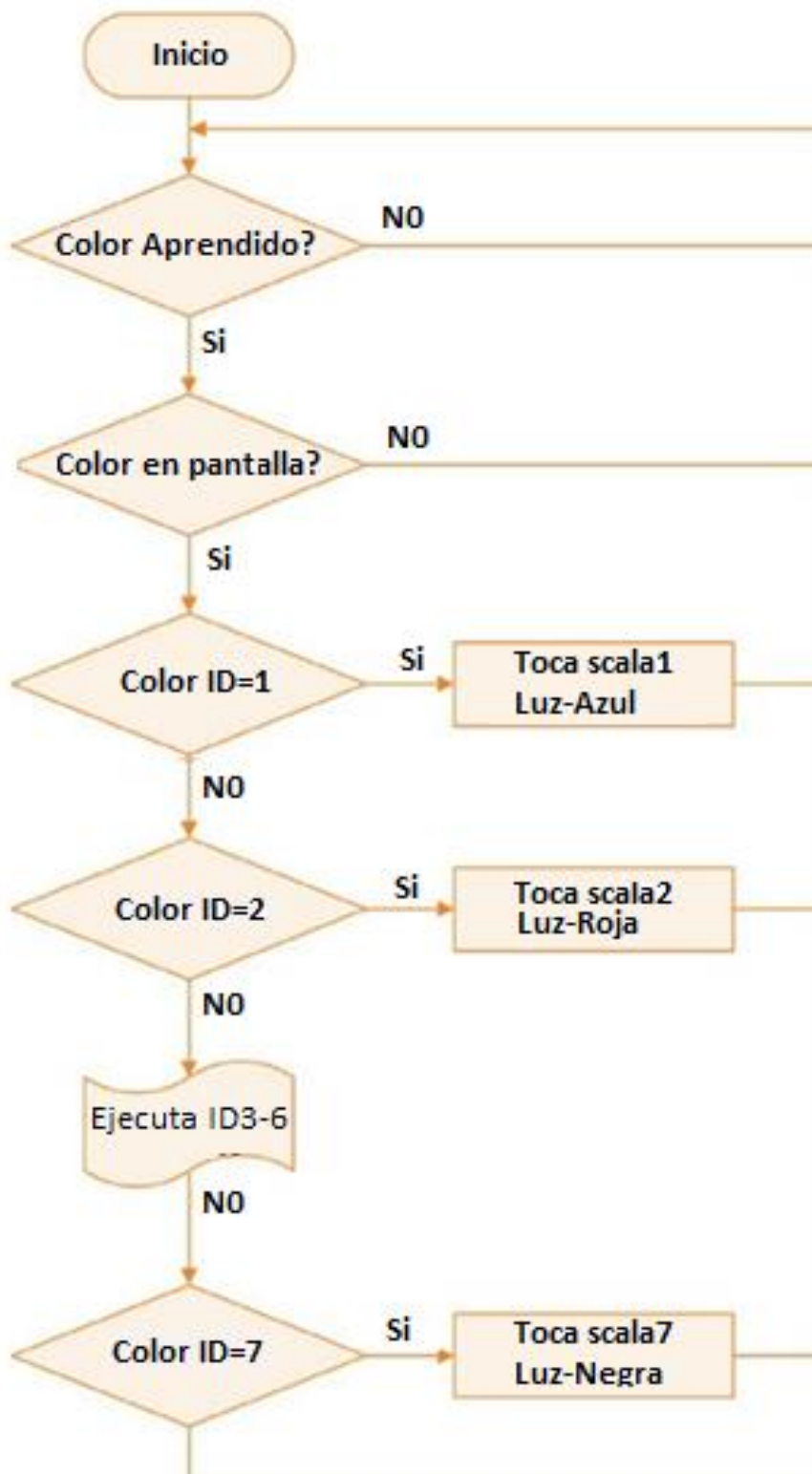
- Instrucciones de tocar notas: se dividen en notas bajas, medias y altas, y varios ritmos.



- Función:** una función puede entenderse como un programa que realiza una función especificada. Haga clic en "Hacer un bloque" para crear una nueva función.



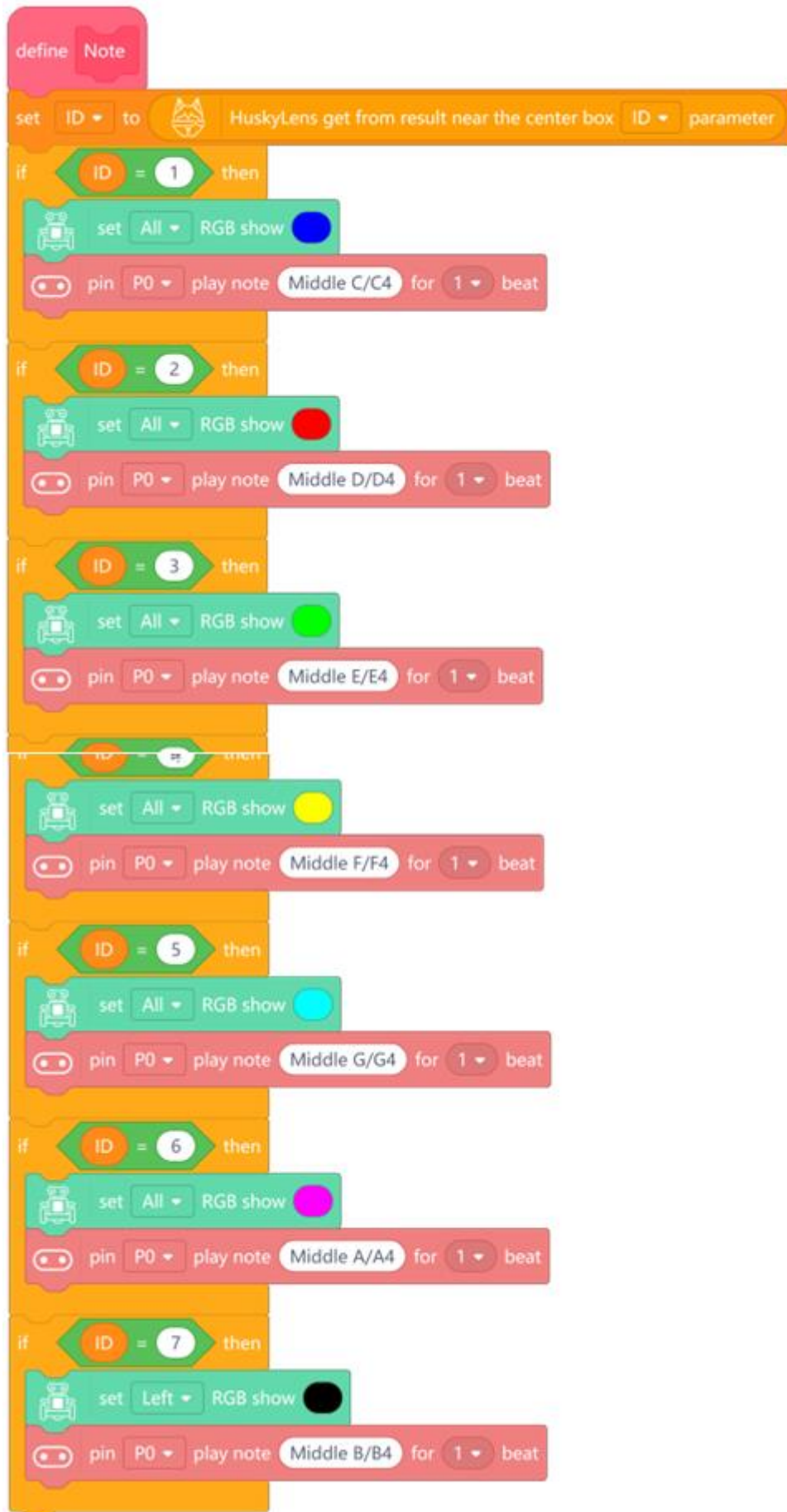
A veces, demasiadas instrucciones en el programa principal afectarán nuestra comprensión. Las funciones pueden simplificar el programa principal y facilitar la comprensión de todo el programa.

**PASO 3 Análisis de diagrama de flujo**

## Ejemplos de Programa

### El Programa

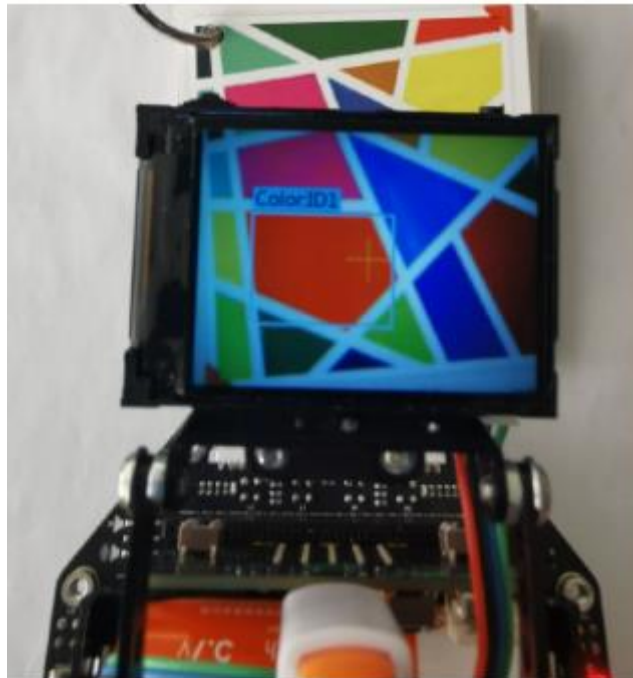






### 3. Resultado de ejecución

Hemos aprendido colores en el orden de azul, rojo, verde, amarillo, cian, morado y negro dado arriba. Luego ejecute el programa, cuando el centro de la pantalla de HuskyLens sea azul, toque la nota 1 y la luz sea azul; cuando es rojo, toque la nota 2, y la luz se vuelve roja, y así sucesivamente.



Si sus programas no funcionan correctamente, verifique los siguientes puntos: (1) Si han aprendido los colores en el orden de azul, rojo, verde, amarillo, cian y morado; (2) Si el modo se cambia a "**Stop Learning**" después de aprender los 7 colores anteriores; de lo contrario, siempre aparecerá un cuadro blanco en la pantalla de la cámara, lo que afectará la ejecución del programa.

### Tarea 3: Eliminar la interferencia errónea de pequeños bloques de color

#### 1. Diseño del Programa

##### PASO 1 Análisis de los problemas encontrados

El programa en la Tarea 2 básicamente puede realizar la función del proyecto, pero algunos pequeños bloques de color pueden convertirse en interferencia cuando hay múltiples bloques de color aparecieron inesperadamente en la pantalla de la cámara.

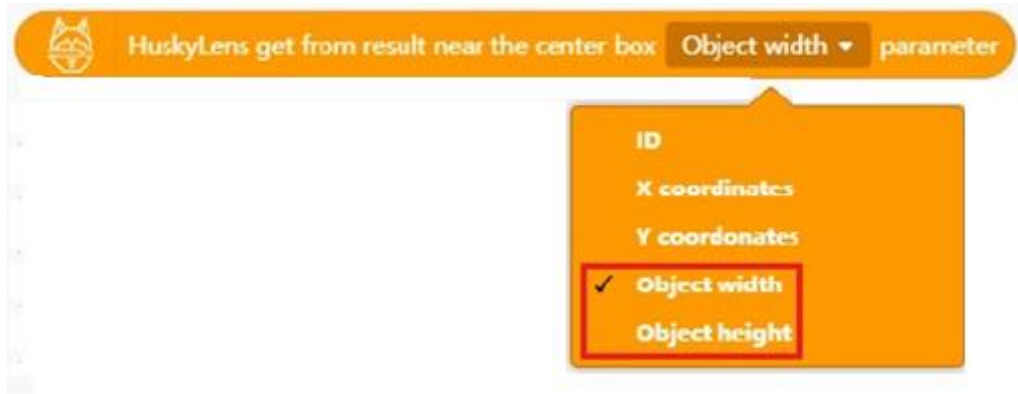
##### ¿Cómo eliminar tal interferencia?

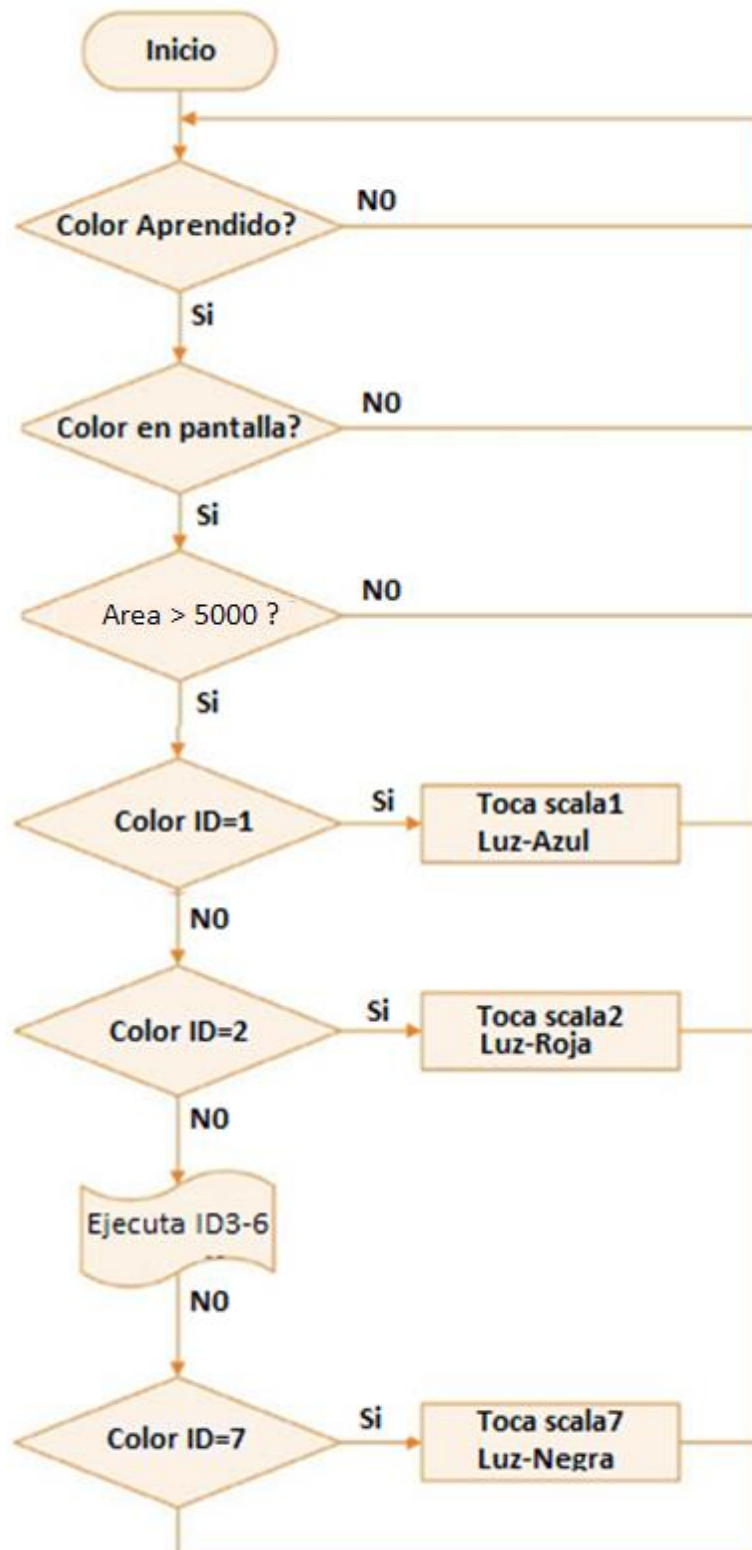
Una solución es juzgar si el bloque de color es el bloque de color objetivo por la proporción de área en la pantalla de la cámara. Podemos establecer un valor umbral, y el bloque de color por debajo del valor umbral se considerará automáticamente como interferencia.

## PASO 2 Instrucción Aprendizaje

Echemos un vistazo a algunas de las instrucciones principales.

1. Obtenga la información en el cuadro cerca del cuadro central del "resultado" solicitado en la interfaz actual, el área del cuadro se puede obtener multiplicando "Ancho del objeto" y "Altura del objeto".



**PASO3 Análisis de diagrama de flujo**

## 2. Ejemplos de programas

### El programa

Modifique el programa en la Tarea 2. La función "nota" permanece sin cambios. El programa principal se modifica de la siguiente manera.



## 3. Resultado de ejecución

Cuando el centro de HuskyLens se muestre en azul, toque la nota 1 y la luz sea azul; cuando se pone rojo, toque la nota 2, la luz se vuelve roja, y así sucesivamente

### Resumen del proyecto:

### Revisión del proyecto

Comprenda el principio de funcionamiento del reconocimiento de color en esta lección y aprenda la función de reconocimiento de color utilizando HuskyLens. El reconocimiento de color es una función muy importante en el reconocimiento de visión de IA y se usa ampliamente en campos industriales. ¿Qué otras funciones interesantes puede lograr el reconocimiento de color? Por favor piénsalo.

### Resumen de conceptos adquiridos

1. Comprender el principio de funcionamiento del reconocimiento de color;

2. Aprenda la función de reconocimiento de color de HuskyLens y el método de operación para reconocer múltiples colores

### Ampliaciones sugeridas al proyecto

Una vez completado este proyecto, puede buscar una notación musical numerada a través de Internet, deletrear el ¡Notación musical numerada con bloques de colores para que Maqueen Plus reproduzca música después de imprimirla! Sin embargo, definitivamente encontraremos un problema: el número de escalas es relativamente pequeño, y las notas medias, bajas y altas no se pueden realizar. Si queremos agregar bloques de color, con el aumento en el número de colores, habrá muchos colores similares, lo que puede conducir a un reconocimiento erróneo. Entonces, ¿hay alguna solución para ampliar el rango de sonido?

(Sugerencias: el botón AB en la placa micro: bit se puede utilizar para realizar las funciones de subir y bajar la tecla).



## Proyecto 2: Sistema de cobro electrónico de peajes

### 1. Introducción

Cuando conducimos en la autopista, nos daríamos cuenta de que no es nada especial en horarios normales, pero si se trata de vacaciones, ¡las estaciones de peaje siempre están llenas de vehículos!

Con la promoción de **ETC** (sistema electrónico de cobro de peaje), la congestión en las estaciones de peaje también se ha reducido. El llamado ETC es un sistema de cobro de peaje electrónico sin detenerse el vehículo. Adopta una forma de cobrar la tarifa del usuario electrónicamente, cuando los propietarios de automóviles con ETC conducen a través de estaciones de peaje a baja velocidad, los peajes se cobran automáticamente.

¿Puede Maqueen Plus realizar la función ETC? En este proyecto, haremos un sistema ETC simple y fácil, de modo que cuando el Maqueen Plus esté en la "autopista".



### 2. Descripción de Funcionamiento

Se aplica en este proyecto la función de “**reconocimiento de etiquetas**” de HuskyLens. Dos etiquetas representan la entrada y salida a las estaciones respectivamente. Al registrar el tiempo de intervalo, se calculan el kilometraje y el peaje, y así se realiza una función de Cobro Fácil con Maqueen Plus.

#### Materiales utilizados

- Tarjeta Micro:bit
- Robot Maqueen Plus
- Tarjeta HuskyLens

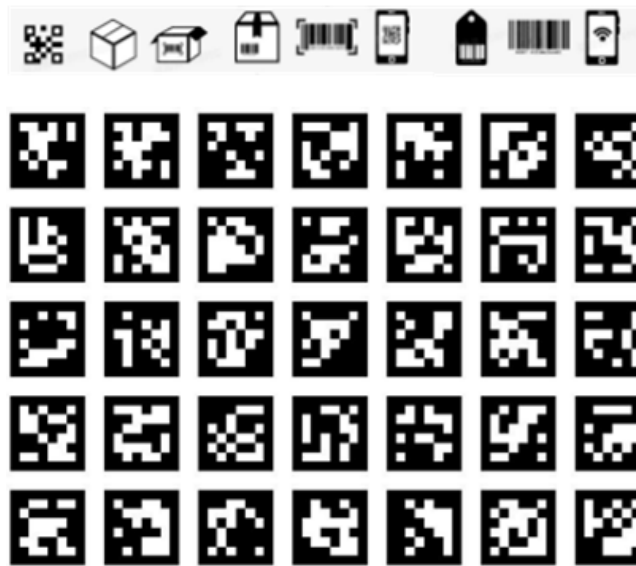
#### Conceptos teóricos a tener en cuenta

¿En qué consiste el reconocimiento de un TAG (marca, imagen, etc)?

La tecnología de reconocimiento de etiquetas se refiere a un método técnico de codificación e identificación efectiva y estandarizada para productos. En la actual sociedad, hay muchos tipos de etiquetas, como códigos de barras y códigos QR.

La biblioteca de referencia visual **AprilTag** será la que adopta HuskyLens. AprilTag puede detectar rápidamente etiquetas y calcular posiciones relativas a través de etiquetas específicas (similar al código QR, pero con complejidad reducida para cumplir con los requisitos en tiempo real).

HuskyLens solo admite el reconocimiento de las etiquetas de biblioteca de referencia visual AprilTag incorporadas, como se muestra en la siguiente imagen



### Principio de funcionamiento del reconocimiento de etiquetas

El algoritmo de AprilTag incluye principalmente los siguientes pasos:

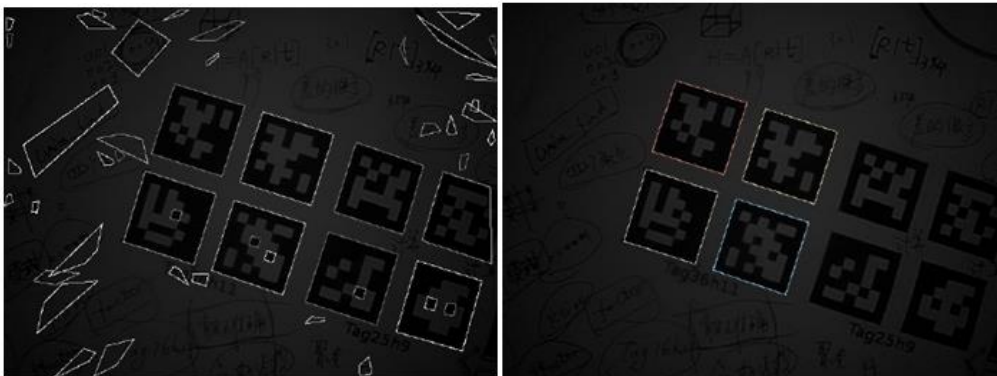
- I. Detección de bordes, buscando el contorno del borde en la imagen.



- II. Detección de cuadros, descubriendo cuadrados en todas las formas de contorno



III. Decodificar, emparejar y verificar los cuadrados detectados.



### 3. Demostración de la función de reconocimiento de etiquetas HuskyLens

#### 1. Detección de etiqueta

Cuando HuskyLens detecta las etiquetas de código QR, estas se marcan automáticamente con un marco blanco en la pantalla.



#### 2. Etiqueta de aprendizaje

Apunte el símbolo "+" a la etiqueta y presione brevemente el "botón de aprendizaje" para completar el aprendizaje de la primera etiqueta.

Después de soltar el "botón de aprendizaje", la pantalla mostrará: "*Haga clic nuevamente para continuar!* Haga clic en otro botón para finalizar". Presione

brevemente el "**botón de aprendizaje**" dentro de la cuenta regresiva si desea aprender la próxima etiqueta. De lo contrario, presione brevemente el "botón de función" antes de que finalice la cuenta regresiva, o no presione ningún botón para esperar a que finalice la cuenta regresiva.



En este proyecto, se deben aprender dos etiquetas, por lo tanto, antes de que finalice la cuenta regresiva, presione el "**botón de aprendizaje**", luego apunte el símbolo "+" a la siguiente etiqueta que se va a aprender y presione brevemente el "**botón de aprendizaje**" para aprender.

El número asociado al ID de etiqueta está en el mismo orden que la etiqueta de aprendizaje, es decir, la ID se marcará como "*etiqueta: ID1*", "*etiqueta: ID2*", "*etiqueta: ID3*", y así sucesivamente. Y los colores del marco para diferentes etiquetas también son diferentes.

### 3. Reconocimiento de etiqueta

Cuando HuskyLens encuentra las etiquetas que se han aprendido, se muestra un marco de color alrededor de la etiqueta aprendida y se indica su ID en la pantalla. El tamaño del marco cambiará de acuerdo con el tamaño del código QR, y los marcos rastrearán automáticamente este código QR.



**Definición de la tarea de reconocimiento:** ¿Cómo se usa el reconocimiento de etiquetas de HuskyLens? ¿Cómo calcular el intervalo de tiempo? Analicemos todo el proyecto en varias tareas pequeñas y completemos el ETC simple paso a paso.

Este proyecto se dividirá en tres pasos para completar la tarea. Primero, aprenderemos a usar la función de reconocimiento de etiquetas de HuskyLens y mostrar la identificación de la etiqueta en la pantalla LED de la placa base. Luego aprenderemos cómo usar el "tiempo de ejecución del sistema" para calcular la diferencia de tiempo entre dos reconocimientos. Finalmente, mejoraremos todo el proyecto y lograremos una función ETC simple.

#### 4. Cuestión 1: Conozca el reconocimiento de etiquetas

##### 1. Conexión de hardware

Al igual que en el proyecto de la notación musical dependiente del color, HuskyLens se fija en Maqueen Plus a través de un soporte y se conecta al conector I2C. Los pasos de conexión son los mismos en los siguientes proyectos, por lo que no se repetirán.

##### 2. Diseño del programa

#### PASO 1 Aprendizaje y reconocimiento

Aquí, puede elegir dos etiquetas a voluntad para que aprenda HuskyLens. (Nota: en primer lugar, cambie a "aprender múltiples" modos)



#### PASO 2 Configuración de Mind + Software

Al igual que el proyecto de reconocimiento de color, abra el software Mind + (versión 163 o superior) y cambie a "modo offline". Haga clic en "Extensiones" y cargue "micro: bit" debajo de "Tarjeta", haga clic para cargar "Maqueen Plus" en "Expansión" y haga clic para cargar "Cámara AI HuskyLens" en "Sensor". La configuración del software en los siguientes proyectos es la misma y no se repetirá

#### PASO 3 Instrucción Aprendizaje

Echemos un vistazo a algunas de las instrucciones principales.

Obtener si **IDx** está en la imagen del "resultado" solicitado. El cuadro corresponde al algoritmo seleccionado con el cuadro (*box*) en la pantalla, y la flecha (*arrow*) corresponde al algoritmo seleccionado con la flecha en la pantalla. Actualmente, el modo *arrow* solo se utilizará para el algoritmo de *seguimiento de línea*, y el cuadro (*box*) se seleccionará en *todos los demás algoritmos*



### 3. Programa Ejemplo

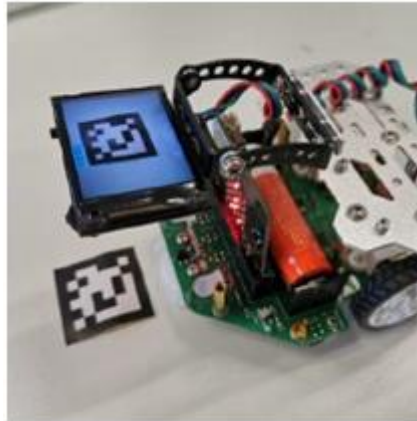
El Programa



### 4. Resultado de la ejecución

Cuando HuskyLens reconoce el código QR en **ID1**, la pantalla LED de la placa base muestra **1**; cuando HuskyLens reconoce el código QR en **ID2**, la pantalla muestra **2**.





Tarea 2: registrar el punto de tiempo de reconocimiento de dos códigos QR y calcular la diferencia horaria

## 1. Diseño del programa

### PASO 1 Análisis de funciones

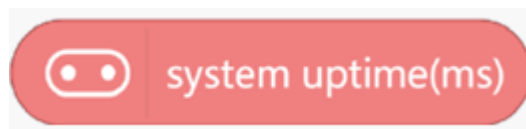
Según la experiencia de la vida, a un automóvil se le cobra el ingreso a una estación de peaje de la autopista y la carga se detiene cuando sale de otra estación de peaje.

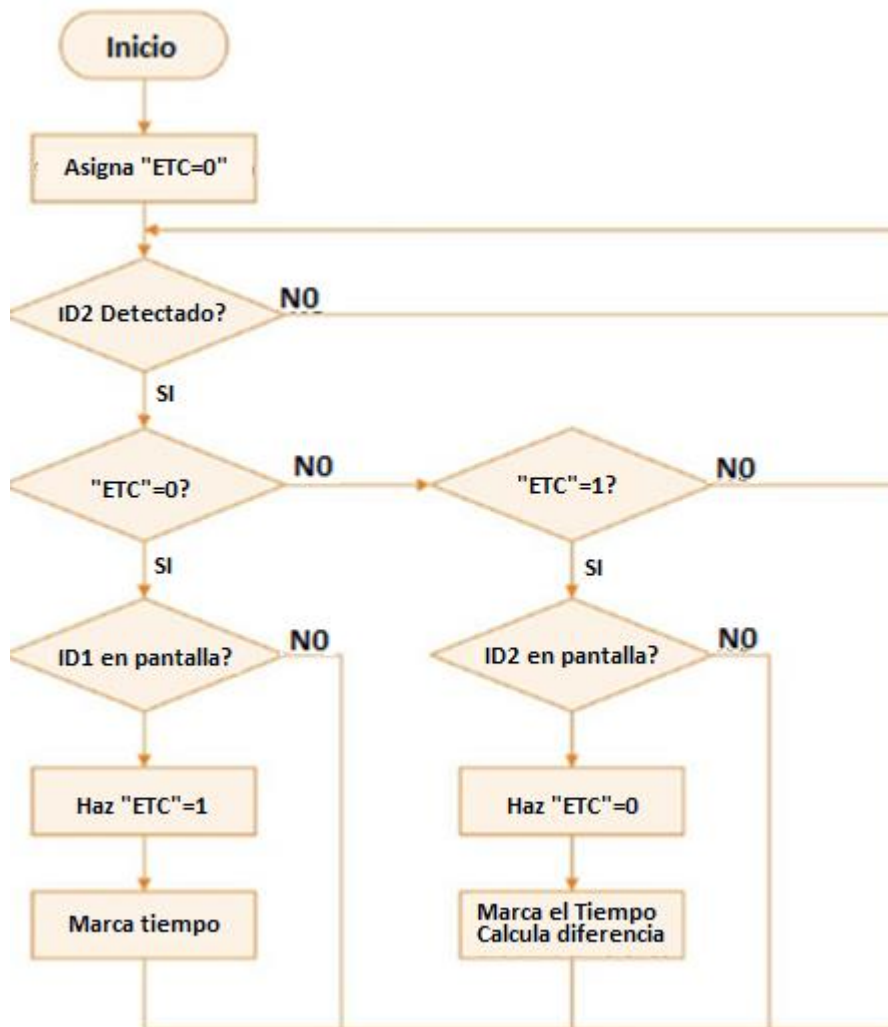
En correspondencia con nuestro proyecto, el punto de **tiempo de inicio** se registra cuando se identifica el código QR **ID1** y el **tiempo final** se registra cuando se identifica el código QR **ID2**. Después de eso, se calcula la diferencia horaria y se calcula la tarifa por milla. Puede haber un problema aquí: ¿qué sucede si el código QR **ID1** se reconoce *dos veces por accidente*?

Cada "*entrada*" se debe corresponder a una "*salida*" en la autopista. También podemos adoptar un método de correspondencia uno a uno. Cuando se reconoce **ID1** por primera vez, se registra la hora de inicio y, posteriormente, la hora de finalización se registra solo cuando se reconoce **ID2** posteriormente.

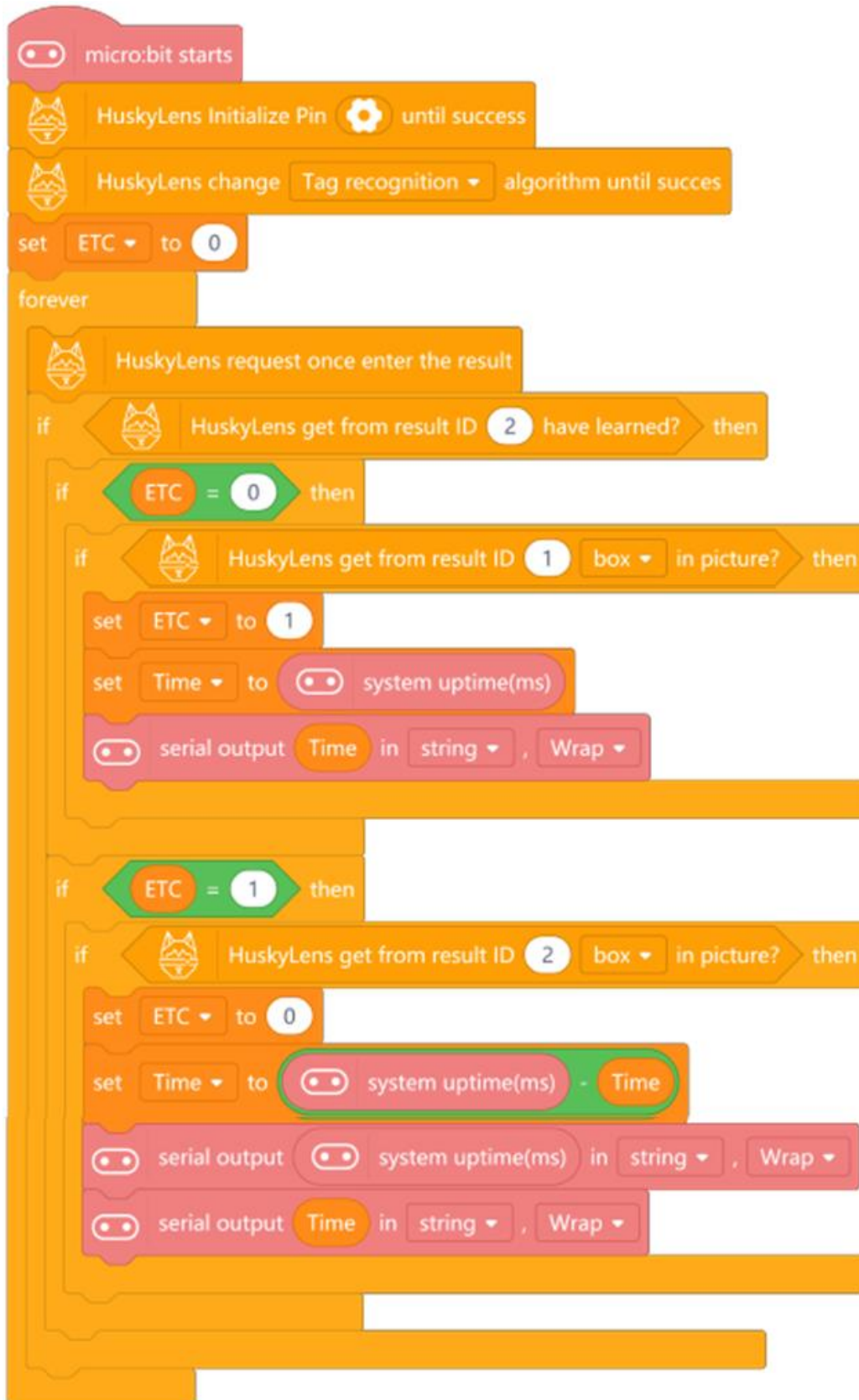
### PASO 2 Instrucción Aprendizaje

Echemos un vistazo a algunas de las instrucciones principales. Indicando el tiempo real después de que el programa comience a ejecutarse. **T** la unidad de tiempo para la lectura es *milisegundos*



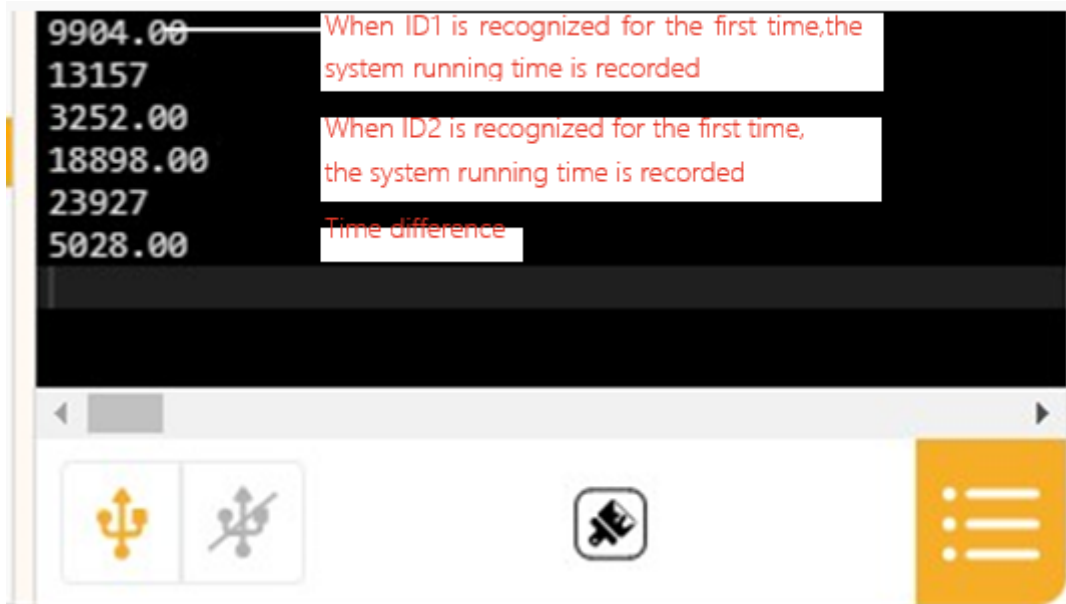
**PASO 3 Análisis de diagrama de flujo**

## 2. Ejemplo de programa



### 3. Resultado de ejecución

Cuando se reconoce **ID1**, el puerto serie emite el tiempo de ejecución del sistema una vez, y solo cuando se reconoce **ID2**, el puerto serie emite nuevamente el tiempo de ejecución del sistema y la diferencia horaria.



### Tarea 3: Inicie Maqueen Plus

#### 1. Diseño del programa

##### Paso 1: Análisis de la Función

Para ver el efecto fácilmente, podemos iniciar Maqueen Plus cuando se reconoce el código QR **ID1** y detenerlo cuando se reconoce el código QR **ID2**

##### Paso 2: Aprendizaje

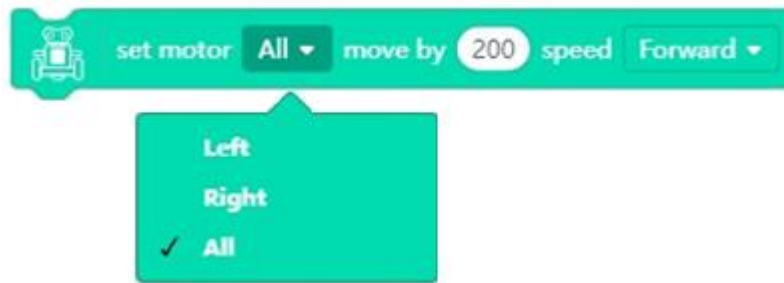
Echemos un vistazo a algunas de las instrucciones principales.

1. ID El PID de apertura puede controlar la velocidad de rotación de la rueda en un bucle cerrado, de modo que la velocidad de rotación real de la rueda no sea interferida por el entorno y esté cerca de la velocidad de rotación establecida.



Nota: El algoritmo PID tiene un cierto retraso y no se recomienda su uso junto con el seguimiento de línea en escala de grises.

2. Configure la velocidad y dirección del motor.



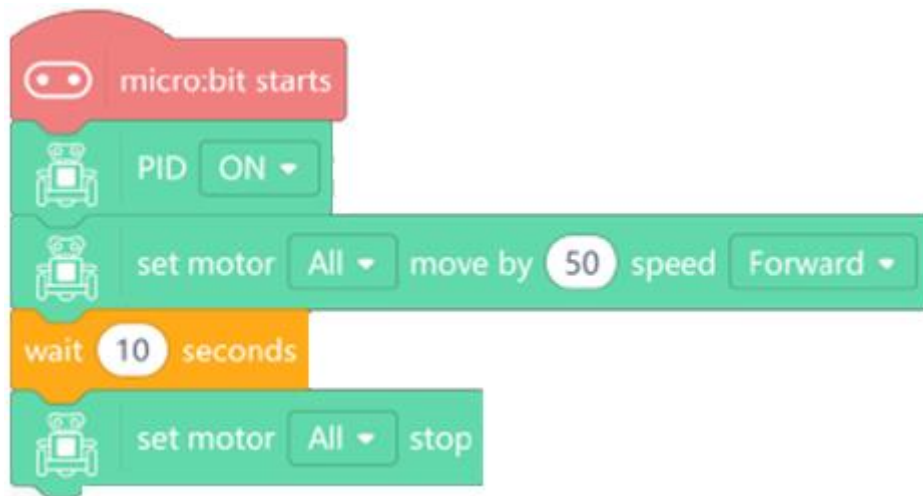
3. Configurar parada del motor



### Paso 3: Test de velocidad

El uso de PID para controlar la velocidad de la rueda ayuda a la precisión de la medición de distancia. Para calcular el kilometraje real de Maqueen Plus, la velocidad debe medirse primero.

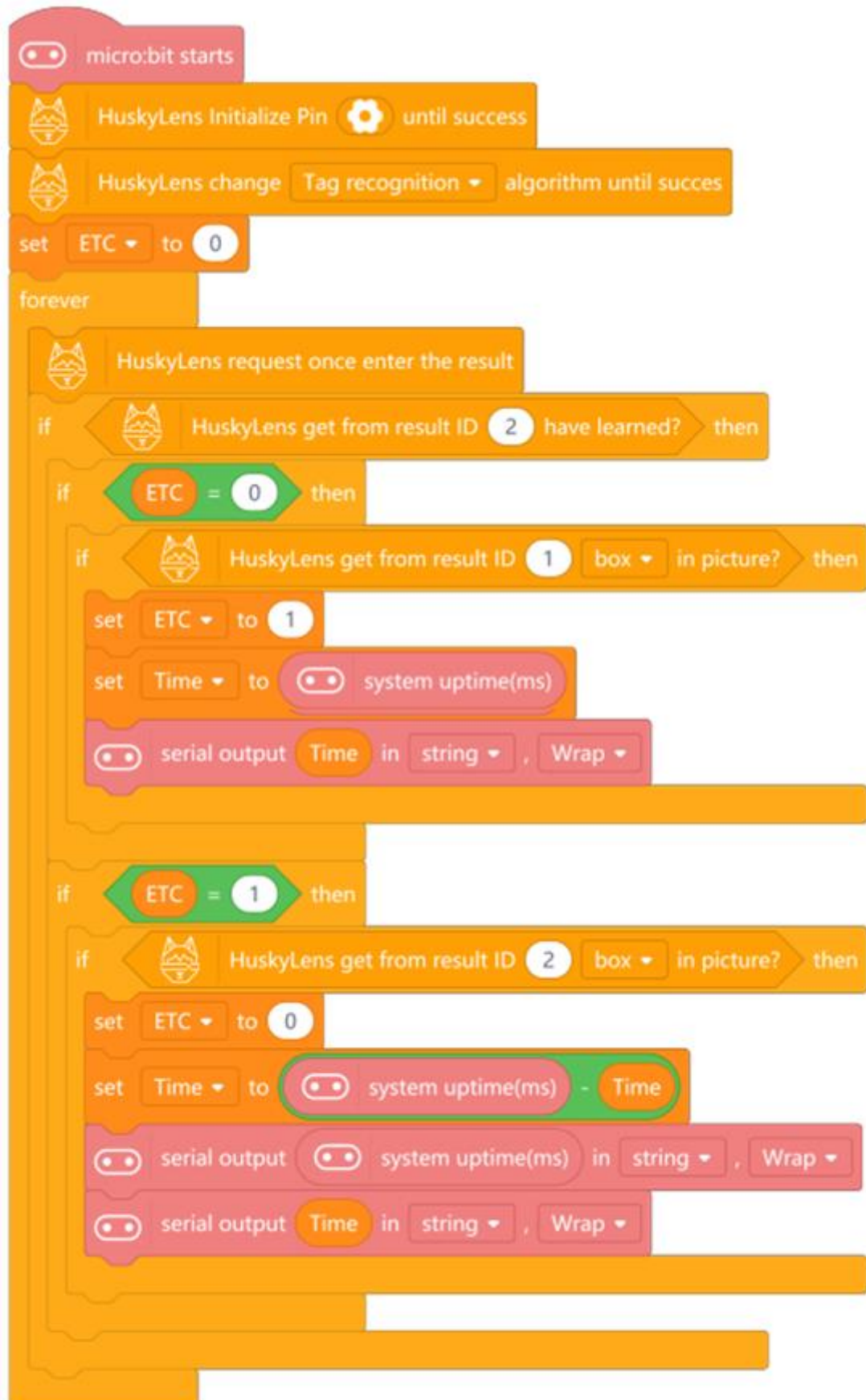
Programa de prueba:



**Método de prueba:** cuando Maqueen Plus ejecuta el programa anterior, avanzará durante 10 segundos a una velocidad de 50. Registre las posiciones inicial y final, mida la distancia  $S$  (unidad: cm), y luego  $S/10$  es la velocidad de kilometraje del automóvil (unidad: cm/seg).



## Ejemplo de programa



## 2. Ejecución y resultados

Cuando HuskyLens reconoce el código QR ID1, se inicia Maqueen Plus; después de eso, cuando HuskyLens reconoció el ID2 código QR, Maqueen Plus se detiene.

### Sumario del proyecto

#### Revisión del proyecto

Comprenda el principio de funcionamiento del reconocimiento de etiquetas en esta lección y aprenda la función de reconocimiento de etiquetas utilizando HuskyLens.

En combinación con la función PID de Maqueen Plus, se creó un ETC simple. El kilometraje y el peaje de la autopista no se calcularon al final del proyecto. Creyendo que puede terminar esta parte del cálculo de datos por su cuenta y mostrar el resultado final en la pantalla LED.

#### Resumen de ideas

1. Comprender el principio de funcionamiento del reconocimiento de etiquetas;
2. Aprender la función y el método de operación del reconocimiento de etiquetas de HuskyLens;
3. Aprenda el control PID de Maqueen Plus

### Propuesta de ampliaciones

Después de completar este proyecto, puede buscar las tarifas de peaje de la autopista local en Internet y finalizar el cálculo del peaje en este proyecto.

Piense en una pregunta: las tarifas de las autopistas varían según las provincias, pero ahora algunos ETC ya han implementado el cobro transprovincial, por lo que no es necesario ingresar o salir de las estaciones de peaje transprovinciales. ¿Cómo funciona? ¿Podemos simular también la carga interprovincial sin parar en nuestro proyecto?

(Consejos: se pueden usar más códigos QR para representar diferentes provincias. Cada provincia se identifica una vez. Finalmente, con el costo cobrado en cada sección, se puede calcular el costo total).

## Proyecto 3: Clasificador inteligente de mercancías

### Introducción

En el puerto de aguas profundas Yangshan de Shanghai, siempre podemos ver tales escenas. Las grúas de puente rojo brillante estiran sus enormes brazos; grúas torre se alinean a lo largo del puerto; Las grúas de ferrocarril bloquean los contenedores con precisión. Enormes contenedores se agarran rápidamente uno por uno y el vehículo guiado automatizado ya espera debajo de ellos; esos contenedores se cargan rápidamente y se remolcan a la posición designada. El flujo de tráfico en el área del puerto es incesante, pero difícilmente puede ver a un trabajador allí.

*Vehículo guiado automatizado*, cuya abreviatura es **AGV**, se refiere a un vehículo no tripulado autónomo que tiene guía automática electromagnética u óptica y puede correr a lo largo de una ruta de guía específica.

¿Puede Maqueen Plus actuar como AGV? ¿Incluso más inteligente que AGV? Por ejemplo, reconocer el tipo de mercancías transportadas y transportarlas a la posición designada de acuerdo con el resultado del reconocimiento.

¡Hagamos que Maqueen Plus sea un AGV con "Ojos"!



### Descripción de funcionamiento:

Este proyecto utiliza la función de reconocimiento de objetos de HuskyLens para identificar diferentes artículos y transportarlos a diferentes posiciones de acuerdo con los resultados del reconocimiento, para que Maqueen Plus pueda transformarse en un experto clasificador de IA.

Para realizar un transporte preciso, el sensor de seguimiento de línea incorporado de Maqueen Plus se utilizará para realizar el transporte de punto fijo a través de un algoritmo de seguimiento de línea simple

## Lista de materiales



Micro:bit



Maqueen Plus



HUSKYLENS

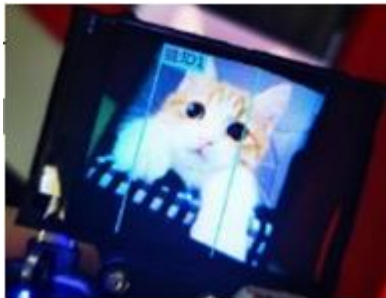
## Conceptos teóricos

Cuando hablamos de visión por ordenador, lo primero que viene a la mente es la clasificación de imágenes (reconocimiento de objetos). Sí, la clasificación de imágenes es una de las tareas más básicas de la visión por ordenador.

En este proyecto, la función de reconocimiento de objetos se aplicó para distinguir diferentes tipos de objetos.

### I. ¿Qué es el reconocimiento de objetos?

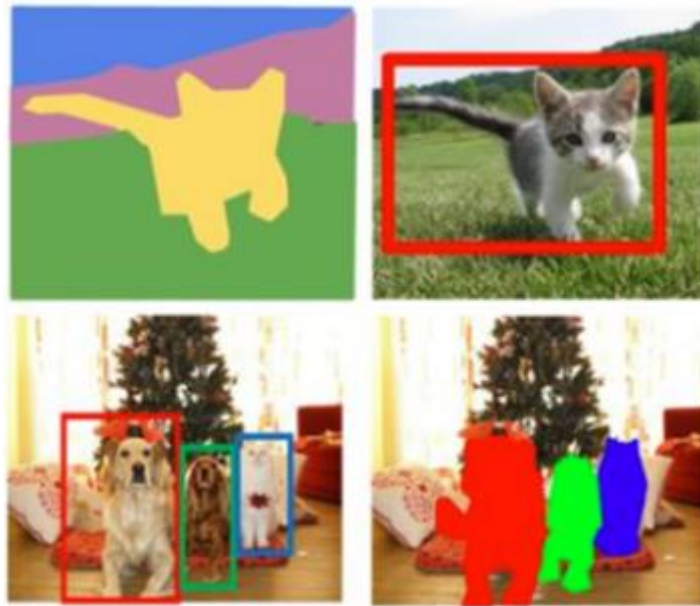
El reconocimiento de objetos se refiere principalmente a la percepción y comprensión de la entidad y el entorno en el mundo tridimensional, y pertenece a la categoría de visión avanzada por ordenador.



### II El principio de trabajo del reconocimiento de objetos

En el reconocimiento de objetos, hay muchas tareas complicadas e interesantes, como la detección de objetivos, la localización de objetos, la segmentación de imágenes, etc. La detección de objetivos puede verse como una combinación de clasificación de imágenes y localización de imágenes. Tomemos un ejemplo como una imagen dada, el sistema de detección de objetos debería poder identificar el objeto en la imagen y dar su ubicación.

En la actualidad, hay dos algoritmos para reconocer objetos, uno es el denominado **R-CNN** y el otro es **Yolo**. El primero, R-CNN es de velocidad más lenta. La siguiente explicación es una introducción a Yolo.



### Algoritmo Yolo:

**Yolo** es una abreviatura de "*solo miras una vez*", lo que significa que se puede hacer una predicción justo después de mirar una vez. La inspiración viene de los humanos, es decir, nosotros mismos. Porque cuando los humanos miramos una imagen, podemos conocer la ubicación de varios tipos de objetos en la imagen de un vistazo.

Yolo se destaca por su simplicidad y velocidad. Si la detección de objetivos se fuera como un proceso de pesca, tendría algoritmos como capturar a los peces con y lo haríamos uno por uno. Cuando se trata de Yolo, ¡solo esparce una red de pesca y los atrapa a todos!

Yolo realiza una predicción basada en la imagen completa, y generará toda la información del objetivo detectado a la vez, incluida la clasificación y la ubicación. El algoritmo de Yolo se implementa dividiendo la imagen de entrada en cuadrículas  $S \times S$ . Si el centro de un objeto cae en una cuadrícula, la cuadrícula correspondiente es responsable de predecir el tamaño y la clase del objeto. Como se muestra en la imagen a continuación, el centro del perro cae en la cuadrícula roja, por lo que la cuadrícula roja es responsable de la predicción de información de este objeto objetivo.

"La cuadrícula en la que cae el centro del objeto es responsable de predecir el objeto" El reconocimiento se divide en dos etapas, que incluyen la **etapa de entrenamiento** y la **etapa de prueba**.

En la etapa de entrenamiento, si el centro del objeto cae en esta cuadrícula, el algoritmo enseñará a esta cuadrícula a predecir los objetos en la imagen. En la etapa de prueba, la cuadrícula continuará haciéndolo naturalmente porque se ha



enseñado en la etapa de entrenamiento para predecir los objetos cuyos centros caen en la cuadrícula

### III. Demostración de la función de reconocimiento de objetos HuskyLens

La función de reconocimiento de objetos en HuskyLens puede identificar cual es el objeto y rastrearlo.

En la actualidad, se admiten 20 tipos de objetos: aviones, bicicletas, pájaros, barcos, botellas, autobuses, automóviles, gatos, sillas, ganado, mesas de comedor, perros, caballos, motocicletas, la humanidad, plantas en macetas, ovejas, sofás, trenes y televisores

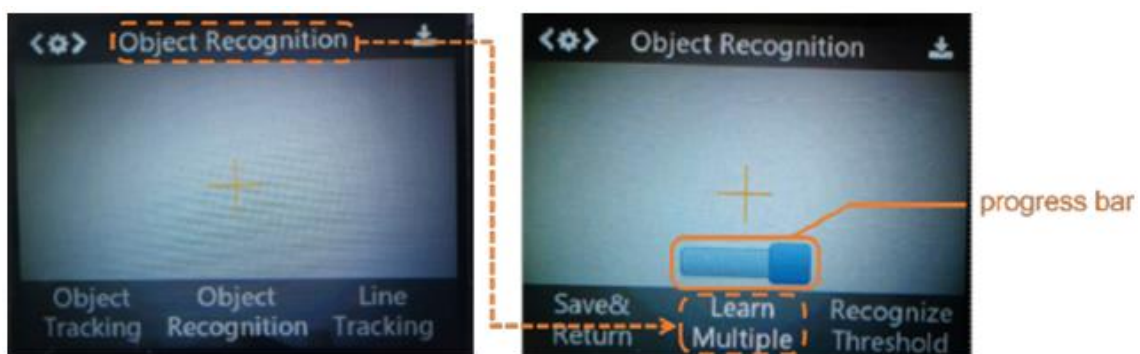
La configuración predeterminada es identificar un objeto, pero también se puede configurar aprender varios.

#### 1. Operación y configuración: aprender múltiples objetos

Marque el botón de función hacia la derecha o hacia la izquierda hasta que aparezca la palabra "*Object Recognition Reconocimiento de objetos*" en la parte superior de la pantalla. Mantenga presionado el botón de función para ingresar la configuración de parámetros de la función de reconocimiento de objetos.

Marque el botón de función hasta que aparezca "*Learn Multiple Aprender múltiples*", luego presione brevemente el botón de función y marque a la derecha para activar el interruptor "Aprender múltiples", es decir, la barra de progreso se vuelve azul y el icono cuadrado en la barra de progreso se mueve hacia Derecha. Luego presione brevemente el botón de función para confirmar este parámetro.

Marque el botón de función hacia la izquierda hasta que aparezca "**Guardar y volver**". Y la pantalla muestra "¿**Desea guardar los parámetros?**" Seleccione "Sí" de forma predeterminada, ahora presione brevemente el botón de función para guardar los parámetros y regresar automáticamente.



#### 2. Detectar objeto

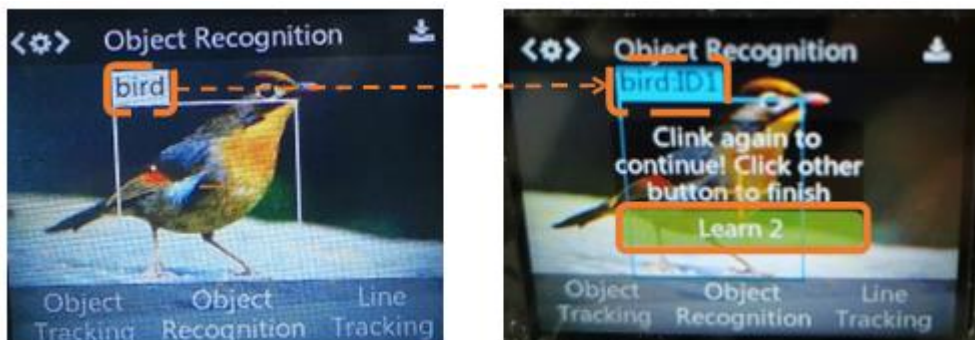
Al detectar objetos, HuskyLens lo reconocerá automáticamente. En la pantalla, los cuadros delimitadores blancos seleccionan el objeto con sus nombres. En la actualidad, solo se pueden reconocer 20 objetos integrados, y los objetos restantes no se pueden reconocer hasta ahora.





### 3. Objeto de aprendizaje

Apunte HuskyLens al objeto de destino. Cuando se detecta el objeto que se muestra en la pantalla y se muestra su nombre, apunte el símbolo "+" al objeto, luego presione brevemente el **"botón de aprendizaje"**. Al presionar, el color del cuadro delimitador cambia de blanco a azul, y el nombre del objeto y su número de identificación aparecerán en la pantalla. Al mismo tiempo, aparece un mensaje: **"¡Haga clic nuevamente para continuar! Haga clic en otro botón para finalizar"**. Presione brevemente el **"botón de aprendizaje"** antes de que finalice la cuenta regresiva si desea aprender el próximo objeto. De lo contrario, presione brevemente el **"botón de función"** antes de que finalice la cuenta regresiva, o no presione ningún botón para que finalice la cuenta regresiva.



Las ID de objeto que se muestran en HuskyLens están en el mismo orden que los objetos de marcado, es decir, las ID se marcarán como "ID1", "ID2", "ID3", etc. Y los colores del cuadro delimitador correspondientes a diferentes ID de objeto también son diferentes.

### 4. Reconocer objeto

Cuando HuskyLens se encuentre con los objetos que se han aprendido, habrá cuadros delimitadores de colores en la pantalla para seleccionar estos objetos y mostrar los nombres e ID de los objetos. El tamaño del cuadro que limita al objeto cambiará de acuerdo con el tamaño del objeto, y los cuadros delimitadores trazarán automáticamente estos objetos. La misma clase de objetos tiene el mismo color de caja, nombre e ID. Se admite el reconocimiento simultáneo de múltiples tipos de objetos, como el reconocimiento de botellas y pájaros al mismo tiempo.



Esta función se puede utilizar como un filtro simple para encontrar el objeto que buscamos de una pila de objetos y rastrearlo.

\* Esta función no puede diferenciar dentro de la misma clase de objetos. Por ejemplo, solo puede identificar que el objeto es gato, pero no puede identificar qué tipo de gato es.

### Práctica del proyecto:

¿Cómo se usa el reconocimiento de objetos de HuskyLens? ¿Cómo controlar Maqueen Plus para seguir la ruta designada?

Dividamos todo el proyecto en varias tareas pequeñas y completemos nuestro “maestro clasificador inteligente” paso a paso.

Este proyecto se dividirá en tres pasos.

- **Primero**, aprenderemos a usar la función de reconocimiento de objetos de Husky-Lens y enviaremos el nombre del objeto a través del puerto serie.
- **Segundo**, aprenderemos el algoritmo de seguimiento de línea en escala de grises para realizar el movimiento de punto fijo.
- **Tercero**, mejoraremos todo el proyecto y simularemos el escenario de clasificación y transporte.

### Tarea 1: Aprender sobre el reconocimiento de objetos

#### 1. Diseño del programa

**Aprendizaje y reconocimiento:** Seleccione 3 elementos aquí para que HuskyLens los aprenda. (Nota: Primero cambie a la función "Aprender múltiples")



## 2 Programa Ejemplo:



### 3. Resultado de la ejecución



Cuando HuskyLens reconoce la botella, la pantalla de la placa principal muestra 1 y saca en la salida del puerto serie “*botella*”. Cuando HuskyLens reconoce la bicicleta, la pantalla de la placa principal muestra 2 y el puerto serie emite “*bicicleta*”. Cuando HuskyLens reconoce la silla, la pantalla de la placa principal muestra 3 y el puerto serie emite “*silla*”.

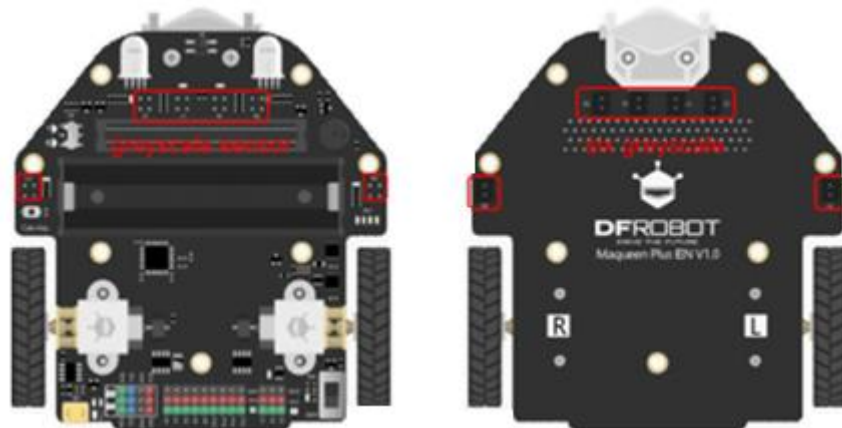
## Tarea 2: Seguimiento de línea en escala de grises - Algoritmo 2IO

### 1. Diseño del programa

#### PASO 1: Análisis de función

Como experto en clasificación de IA, Maqueen Plus utiliza la función de “**Reconocimiento de objetos**” de HuskyLens para identificar el tipo de objeto.

Pero la tarea de clasificación solo se puede completar cuando el objeto se transporta a la ubicación designada, al igual que el AGV en el puerto, que completa el transporte automatizado.



Para este tipo de transporte punto a punto, puede usar el sensor de escala de grises incorporado en Maqueen Plus e implementarlo a través del algoritmo de “seguimiento de línea”.

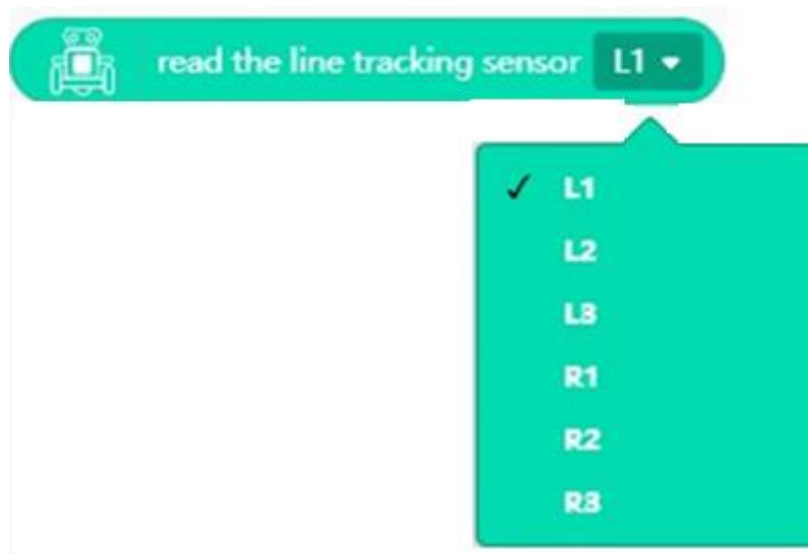
Maqueen Plus tiene 6 sensores de escala de grises integrados en la parte inferior, que se pueden usar para detectar la línea negra. Cuando el sensor de escala de grises se enfrenta a un fondo blanco, el LED indicador de escala de grises está

apagado y el detector del valor del sensor de escala de grises es 0; cuando el sensor de escala de grises se enfrenta a la línea negra, el LED indicador de escala de grises está encendido y el valor de detección del sensor de escala de grises es 1

## PASO 2 Instrucciones de aprendizaje

Echemos un vistazo a algunas de las instrucciones principales.

1. Lee el valor del sensor de seguimiento de línea y los valores de retroalimentación serán 0 o 1, 1 indica la línea negra. Seleccione escala de grises en el cuadro desplegable, L1, L2, L3, R1, R2, R3 se corresponden la etiqueta que cada uno tiene marcada en la parte inferior de Maqueen Plus



## PASO 3 Test de escala de grises

Salida del valor en escala de grises a través del puerto serie. Programa de prueba:



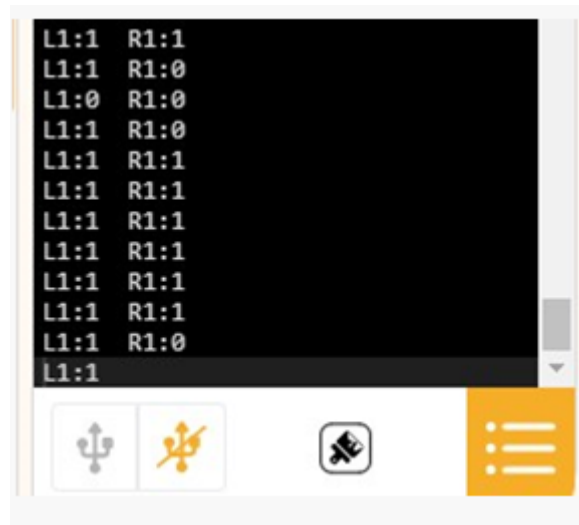
Resultados de la prueba:

Cuando los sensores L1 y R1 están ambos en la línea negra, las salidas del puerto serie 1, 1;

cuando solo L1 está en la línea negra, las salidas de puerto serie 1, 0;

cuando solo R1 está en la línea negra, las salidas de puerto serie 0, 1;

si ambas escalas de grises no están en la línea negra, se generarán 0 y 0



Si el programa no funciona correctamente, solucione los siguientes problemas:

1. El color negro impreso por la impresora puede no reconocerse correctamente. La cinta adhesiva negra y el mapa impreso se pueden usar normalmente.
2. La luz ambiental puede afectar el sensor de escala de grises. Cuando la luz cambia mucho, la escala de grises necesita ser recalibrada. Método de calibración en escala de grises: Maqueen Plus tiene una función de calibración en escala de grises de un botón. El botón de calibración se muestra en la siguiente imagen. Cuando esté en uso, asegúrese de que todos los sensores de seguimiento de línea estén en el área de calibración negra. Presione el botón de calibración durante 1 segundo. Las dos luces RGB delante de Maqueen Plus parpadean en verde, lo que indica que la calibración se ha completado. Suelta el botón para completar.

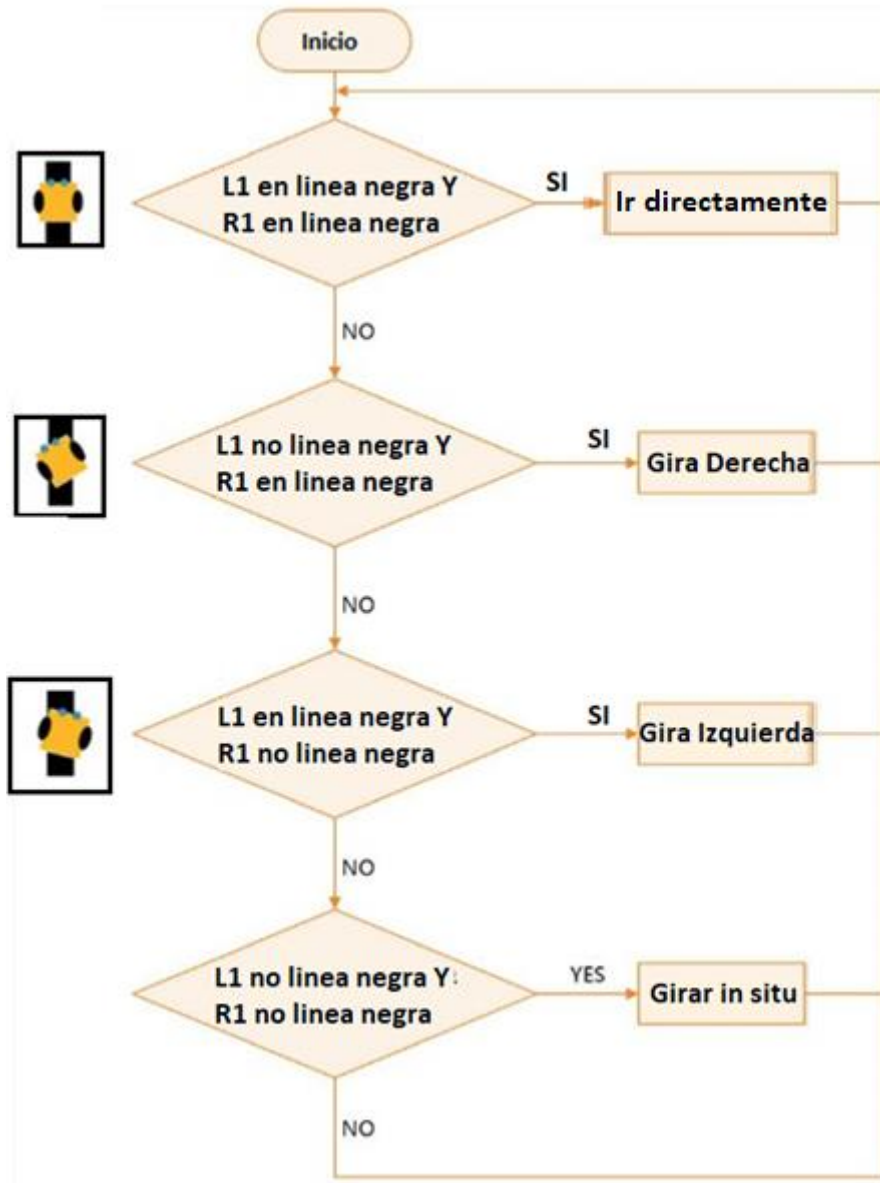
\* **Principio del sensor de escala de grises:** cada sensor de escala de grises consta de un emisor de infrarrojos y un receptor de infrarrojos. Debido a que a menudo se usa para controlar robots para caminar a lo largo de la línea, también se le llama sensor de seguimiento de línea. El transmisor infrarrojo emite continuamente luz infrarroja al suelo. Si se refleja la luz infrarroja (como encontrarse con planos blancos u otros de color claro), el receptor recibe la señal infrarroja, emite un valor de 0 y el LED indicador de escala de grises está apagado. Si la luz infrarroja se absorbe o no se puede reflejar, el receptor no



recibirá señales infrarrojas, emitirá un valor de 1 y el LED indicador de escala de grises estará encendido.

#### PASO 4 Análisis del Diagrama de flujo

Aquí usamos dos escalas de grises L1 y R1 para patrullar la línea, y el ancho predeterminado de la línea negra es 2 cm. Cuando L1 y R1 están en la línea negra, Maqueen Plus sigue recto



## 2 Programas Ejemplo.



## 3. Efecto de la operación

Si no tiene un mapa adecuado a la mano, se recomienda pegue un mapa de patrulla con cinta negra de 2 cm de ancho, como la siguiente imagen.



Al ejecutar el programa, el automóvil patrullará automáticamente la línea negra.



## Tarea 3: configuración y optimización de programas

### 1. Diseño del programa

#### **PASO 1 Análisis de función**

Suponiendo que hay que clasificar tres tipos de objetos,

Maqueen Plus completará el transporte a lo largo de una determinada pista después de reconocer la clase de objeto.

Para controlar con precisión Maqueen Plus, la unión en T, la unión izquierda y la unión derecha (como se muestra en la siguiente imagen) se agregan al mapa de seguimiento de línea, y el juicio de unión se agrega correspondientemente a los programas



Uso de escalas de grises L3 y R3 en Maqueen además para ayudar a juzgar los cruces.

#### **PASO 2 Programa de juicio de unión**

Modifique la función "leer en escala de grises" en la tarea 2 y cree las funciones "juicio de unión", "gire a la izquierda" y "dobla a la derecha".

```
define read greyscale
  set L1 to 0
  set R1 to 0
  set L3 to 0
  set R3 to 0
  if read the line tracking sensor L1 = 1 then
    set L1 to 1
  if read the line tracking sensor R1 = 1 then
    set R1 to 1
  if read the line tracking sensor L3 = 1 then
    set L3 to 1
  if read the line tracking sensor R3 = 1 then
    set R3 to 1
```

The image shows a Scratch script for a function named "read greyscale". The function starts with a pink "define" block. It then contains four "set" blocks, each setting a variable (L1, R1, L3, and R3) to the value 0. Following these are four "if" blocks. Each "if" block checks if the value of a specific line tracking sensor (L1, R1, L3, or R3) is equal to 1. If the condition is true, the corresponding "set" block sets that variable to 1. The sensors are represented by a small robot icon in the "if" blocks.

```

define junction judgment
if L3 = 1 and R3 = 1 then
set Junction type to "T"
else if L3 = 1 then
set Junction type to "L"
else if R3 = 1 then
set Junction type to "R"
else
set Junction type to ""
display Junction type

```

Se sugiere que al depurar el programa, las funciones anteriores "juicio de unión", "girar a la izquierda" y "girar a la derecha" se ejecuten por separado para probar si cada función puede ejecutar la función correspondiente

```

define turn right
set motor Left to move by 60 speed Backward
set motor Right to move by 60 speed Forward
wait 0.9 seconds
set motor All to stop

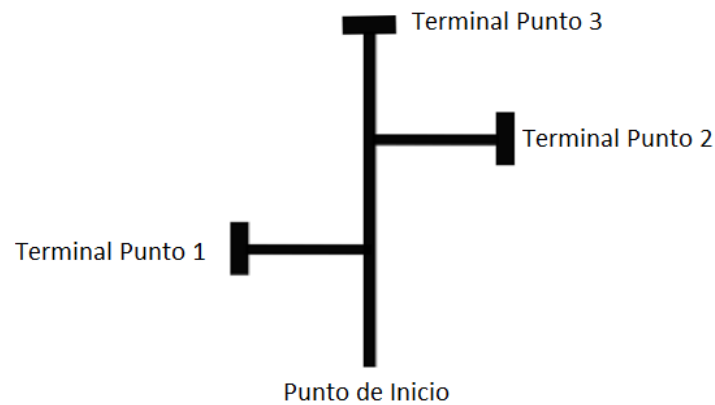
```

```

define turn left
set motor Left to move by 60 speed Backward
set motor Right to move by 60 speed Forward
wait 0.9 seconds
set motor All to stop

```

### PASO 3 Ejemplo de mapa



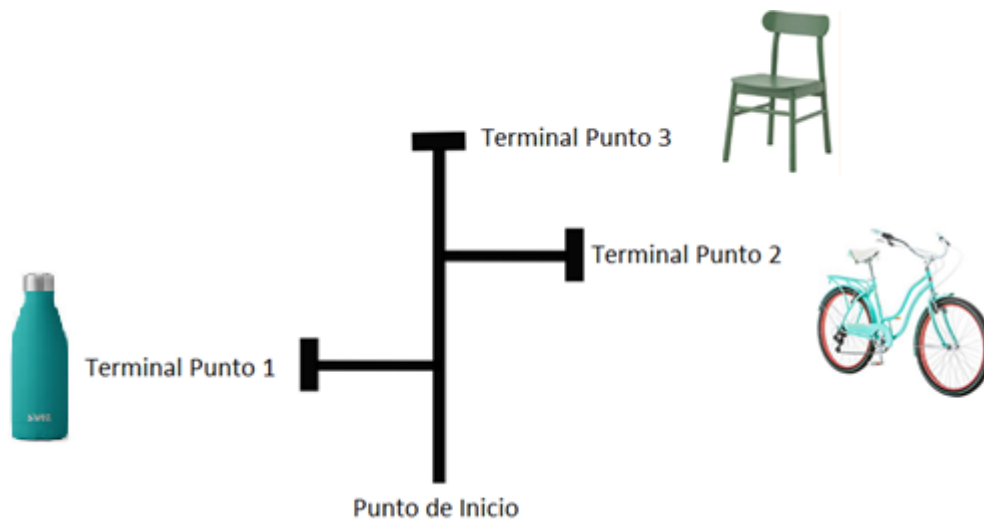
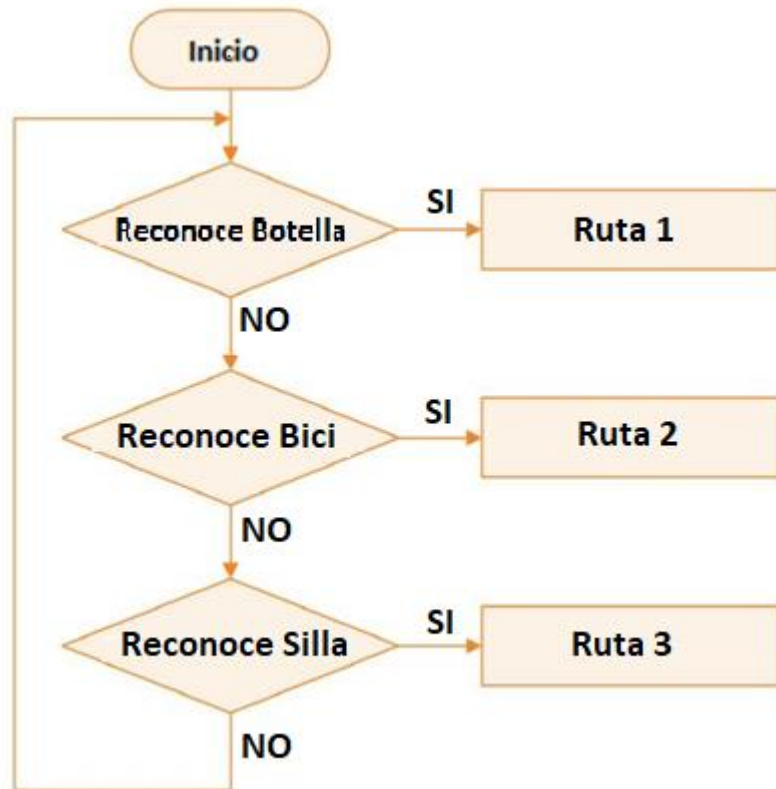
### PASO 4 Análisis de diagrama de flujo

Los diferentes mapas necesitan una planificación de ruta diferente. Tome el mapa de arriba como ejemplo, la planificación de la ruta desde el punto de partida hasta el punto final 1 es: si Maqueen Plus patrulla a lo largo de la línea hasta que se encuentra con el cruce izquierdo, Maqueen Plus gira a la izquierda y continúa moviéndose a lo largo de la línea hasta se encuentra con la unión T, llega al punto terminal 1. Si HuskyLens reconoce el objeto de ID1, irá al punto terminal 1 de acuerdo con la planificación de la ruta. Punto terminal 2, 3 y así sucesivamente.

La secuencia de reconocimiento de objetos en la Tarea 1 se copia aquí.



El diagrama de flujo muestra lo siguiente



## 2. Ejemplo de programa

Modifique el programa principal y agregue las funciones "ruta 1", "ruta 2" y "ruta 3". Cuando realmente esté programando, consulte su propio mapa para hacer un plan de ruta. El siguiente programa usa el mapa de ejemplo anterior



```

micro:bit starts
HuskyLens Initialize Pin until success
HuskyLens change Object recognition algorithm until success
set ID to 0
repeat until ID > 0
  HuskyLens request once enter the result
  if HuskyLens get from result ID 1 have learned? then
    if HuskyLens get from result ID 1 box in picture? then
      set ID to 1
    if HuskyLens get from result ID 2 box in picture? then
      set ID to 2
    if HuskyLens get from result ID 3 box in picture? then
      set ID to 3
  if ID = 1 then
    route 1
  if ID = 2 then
    route 2
  if ID = 3 then
    route 3

define route 3
  repeat until Junction type contains "T"?
  line tracking
  junction judgment
  set motor All stop
  display "ID3"

define route 1
  repeat until Junction type contains "L"?
  line tracking
  junction judgment
  set motor All stop
  turn left
  repeat until Junction type contains "T"?
  line tracking
  junction judgment
  set motor All stop
  display "ID1"

define route 2
  repeat until Junction type contains "R"?
  line tracking
  junction judgment
  set motor All stop
  turn right
  repeat until Junction type contains "T"?
  line tracking
  junction judgment
  set motor All stop
  display "ID2"
    
```

## Proyecto 4: Detective encubierto

Después de años de desarrollo, la tecnología de reconocimiento facial ha comenzado a implementarse. Por ejemplo, cuando al ingresar a la estación de tren, la información de la tarjeta de identificación se compara a través de la cámara. Y el reloj de escaneo facial de entrada / salida en el trabajo, el pago de escaneo facial de **Alipay**, el desbloqueo facial en el teléfono inteligente, etc.



La tecnología de reconocimiento facial también se ha aplicado de manera efectiva en el campo de la seguridad pública, como arrestar a fugitivos. En los últimos años, muchos fugitivos han sido arrestados con éxito muchas veces en conciertos de estrellas cantantes. Usando la tecnología de reconocimiento facial de IA, cuando un fugitivo pasa por un control de seguridad, la cámara captura la información de la cara del fugitivo y la compara con la base de datos de fondo, luego el sistema emitirá un mensaje de

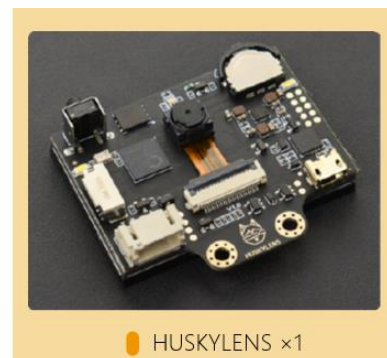
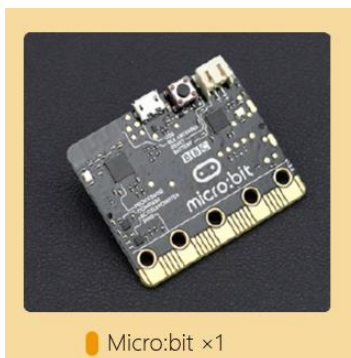
advertencia.

**HuskyLens** tiene una función de reconocimiento facial incorporada. ¿Se puede combinar **Maqueen Plus** con **HuskyLens** para atrapar a los malos? En la vida real, los malos a menudo tienen que tomar la iniciativa de ir a lugares con reconocimiento facial antes de que puedan ser identificados y atrapados. Entonces, ¿cómo pueden atrapar a los malos en esos lugares sin cámaras de reconocimiento facial?

### Descripción del proyecto:

Este proyecto utiliza la función de “**reconocimiento facial**” de la cámara **HuskyLens** para hacer que **Maqueen Plus** se convierta en un detective encubierto. En tiempos normales, puede moverse libremente y tiene una función básica para evitar obstáculos. A través del reconocimiento facial, la cara en la pantalla se compara con la base de datos de rostros de personas y se activará una alarma una vez que se encuentre al chico malo.

### Lista de materiales



## Conceptos Teórico

El rostro humano, como otras características biológicas del cuerpo humano (la huella digital, el iris, etc.), es singularmente único y no es fácil ser duplicado. Las imágenes faciales pertenecen a la primera categoría de imágenes estudiadas y también son las más utilizadas en el campo de la visión por ordenador. Este proyecto utiliza la **función de reconocimiento facial** HuskyLens

### I ¿Qué es el reconocimiento facial?

El reconocimiento facial es un tipo de tecnología de identificación biométrica basada en información de rasgos faciales humanos. Al recopilar imágenes o videos que contienen rostros humanos con una cámara de imagen o cámara de video, detectar automáticamente la información de la imagen y rastrear rostros humanos, se realiza una serie de análisis técnicos sobre los rostros humanos detectados



### II El principio de trabajo del reconocimiento facial

Hay cuatro pasos clave en el proceso de reconocimiento facial:

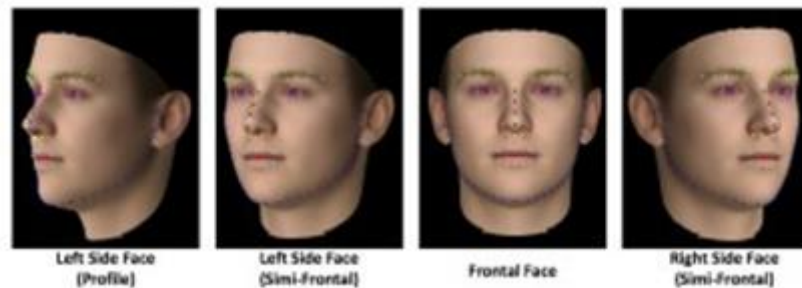


La siguiente es una breve descripción de estos 4 pasos.

1. **Detección de rostro:** encuentre la ubicación del rostro en la imagen, generalmente marcada con un límite



2. **Alineación de caras:** identifique caras desde diferentes ángulos localizando puntos de características en la cara

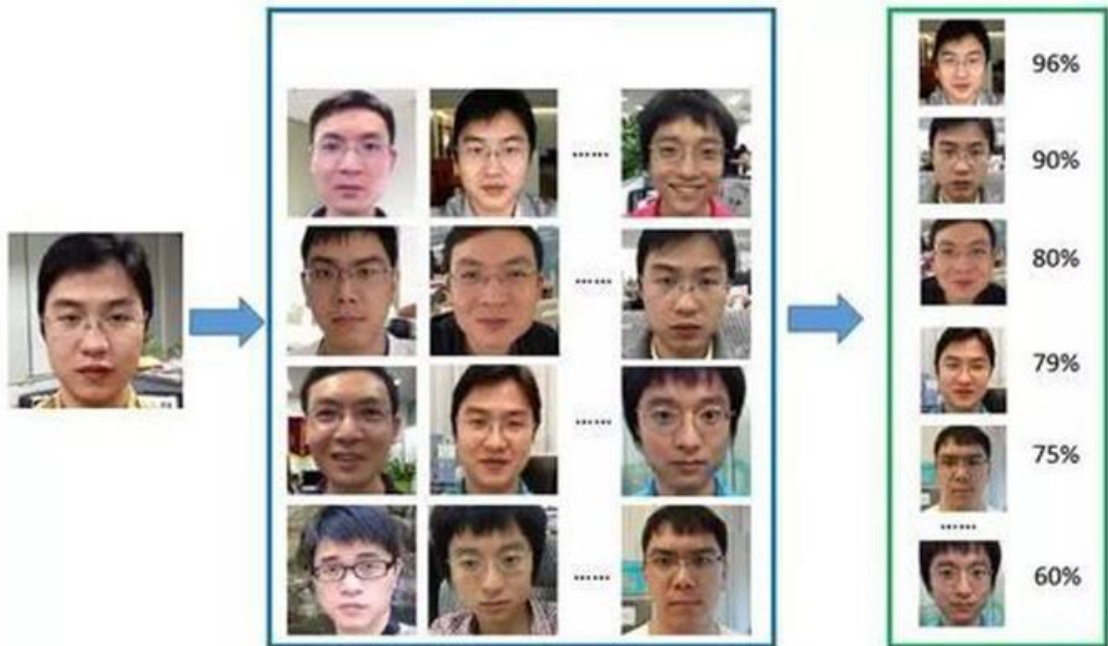


3. **Codificación facial:** se puede entender simplemente como extraer información facial y convertirla en información comprensible para los ordenadores



$[-0.23, -0.54, \dots, 0.27]$

4. **Comparación de caras:** compare la información de la cara con la base de datos existente para obtener una puntuación de comparación y dar un resultado en función de esta comparación



El reconocimiento facial también se considera uno de los temas de investigación más difíciles en el campo del reconocimiento biométrico o incluso el área de inteligencia artificial. La dificultad del reconocimiento facial se atribuye principalmente a las características del rostro humano como entidad biológica.

**Similitud:** Hay poca diferencia entre cada individuo. La estructura de todos los rostros humanos es similar, incluso la estructura y la apariencia de los órganos del rostro humano son muy similares. Esta característica es beneficiosa para localizar el rostro humano, pero no es beneficioso para distinguir el individuo humano por el rostro humano.

**Cambiabilidad:** la apariencia de un rostro humano es inestable. Las personas pueden producir muchas expresiones a través de cambios en la cara, y las imágenes visuales de la cara varían mucho en diferentes ángulos de observación. Además, el reconocimiento facial también está influenciado por varios factores, incluidas las condiciones de iluminación (como día y noche, interiores y exteriores, etc.), cubiertas en la cara (como máscaras, gafas de sol, cabello, barba, etc.), edad, etc.

### III Escenarios de aplicación de reconocimiento facial

**Sistema de control de acceso:** las áreas bajo protección de seguridad pueden identificar la identidad de aquellos que intentan ingresar a través del reconocimiento facial, como prisiones, centros de detención, áreas residenciales, escuelas, etc.

**Sistema de videovigilancia:** Monitoreo de multitudes en lugares públicos como bancos, aeropuertos, estadios, centros comerciales, supermercados, etc. para lograr el propósito de identificación. Por ejemplo, se instala un sistema de monitoreo en el aeropuerto para evitar que los terroristas aborden aviones.





**Aplicación de red:** Uso del reconocimiento facial para ayudar al pago en línea con tarjeta de crédito para evitar que los propietarios de tarjetas que no son de crédito usen tarjetas de crédito o para prevenir reclamos falsos de seguro social, etc.

El reconocimiento facial se usa ampliamente en todos los ámbitos de la vida en la actualidad, como el sistema de asistencia de estudiantes, la cámara, el desbloqueo del teléfono móvil y el sistema de autenticación de personas.

#### IV. Demostración de la función de reconocimiento facial de HuskyLens

##### 1. Seleccione la función "Reconocimiento facial"

Marque el botón de función hacia la izquierda hasta que aparezca la palabra "*Face Recognition Reconocimiento facial*" en la parte superior de la pantalla.



##### 2. Aprendizaje facial

Orientar la cámara HuskyLens al área con caras. Cuando se detecta una cara, se seleccionará automáticamente mediante un cuadro delimitador blanco en la pantalla, con la palabra "*Face Cara*" arriba.



Apunte el símbolo "+" a una cara, presione brevemente el **"botón de aprendizaje"** para aprender la cara. Si HuskyLens detecta la misma cara, aparecerá un cuadro azul con las palabras **"Cara: ID1"** en la pantalla, lo que indica que HuskyLens ha aprendido la cara antes y puede reconocerla ahora.

\* Mantenga presionado el "botón de aprendizaje", se puede aprender una cara desde diferentes ángulos.

\* Si no hay un símbolo "+" en el centro de la pantalla antes de aprender, significa que HuskyLens ya ha aprendido la cara en la función actual. En este momento, presione brevemente el "botón de aprendizaje", la pantalla mostrará "haga clic de nuevo para olvidar". Antes de que finalice la cuenta regresiva, presione brevemente el "botón de aprendizaje" nuevamente para eliminar la información de la cara aprendida.

### Proyecto práctico

¿Cómo se usa el reconocimiento facial HuskyLens? ¿Cómo implementar la ejecución de Maqueen Plus para evitar obstáculos? ¿Cómo detectar al chico malo? Dividamos todo el proyecto en varias tareas pequeñas y completemos paso a paso la tarea de detective encubierto.

Este proyecto se completará en tres pasos.

1. En primer lugar, aprenda a usar la función de reconocimiento facial de HuskyLens y muestre la ID de cara correspondiente en la pantalla de la placa base.
2. Luego, aprenda a implementar la ejecución de Maqueen Plus para evitar obstáculos.
3. Finalmente, mejore todo el proyecto permitiendo que Maqueen Plus se ejecute mientras evita los obstáculos y detecta a los malos. Si se encuentra al chico malo, se activará una alarma.



## Tarea 1: Conozca el reconocimiento facial

### 1. Diseño del programa

#### PASO 1 Aprendizaje y reconocimiento

Aquí puede elegir algunas imágenes de la cara a voluntad y dejar que HuskyLens aprenda. (Nota: Primero cambie a la función "**aprender múltiples**")



#### PASO 2 Instrucción Aprendizaje

Echemos un vistazo a algunas de las instrucciones principales.

1. Obtenga cuántos objetivos se han aprendido bajo el algoritmo actual del "**result**" solicitado. Tenga en cuenta que para aprender *múltiples rostros* de personas debemos configurar este parámetro desde HuskyLens presionando prolongadamente el botón de función para ingresar la configuración de parámetros.



#### PASO 3 Función de análisis

Suponiendo que solo hay 3 datos faciales, el programa es fácil de implementar. Como se muestra en el siguiente imagen, solo necesito juzgar uno por uno.

Suponiendo que solo hay 3 datos faciales, el programa es fácil de implementar. Como se muestra en la siguiente imagen, solo es necesario juzgar uno por uno.



Sin embargo, cuando hay 5, 10 o más caras aprendidas, ¿cómo pueden ser implementadas por los programas? Juzgar uno por uno es bastante prolijo. ¿Has encontrado algún patrón del programa anterior?

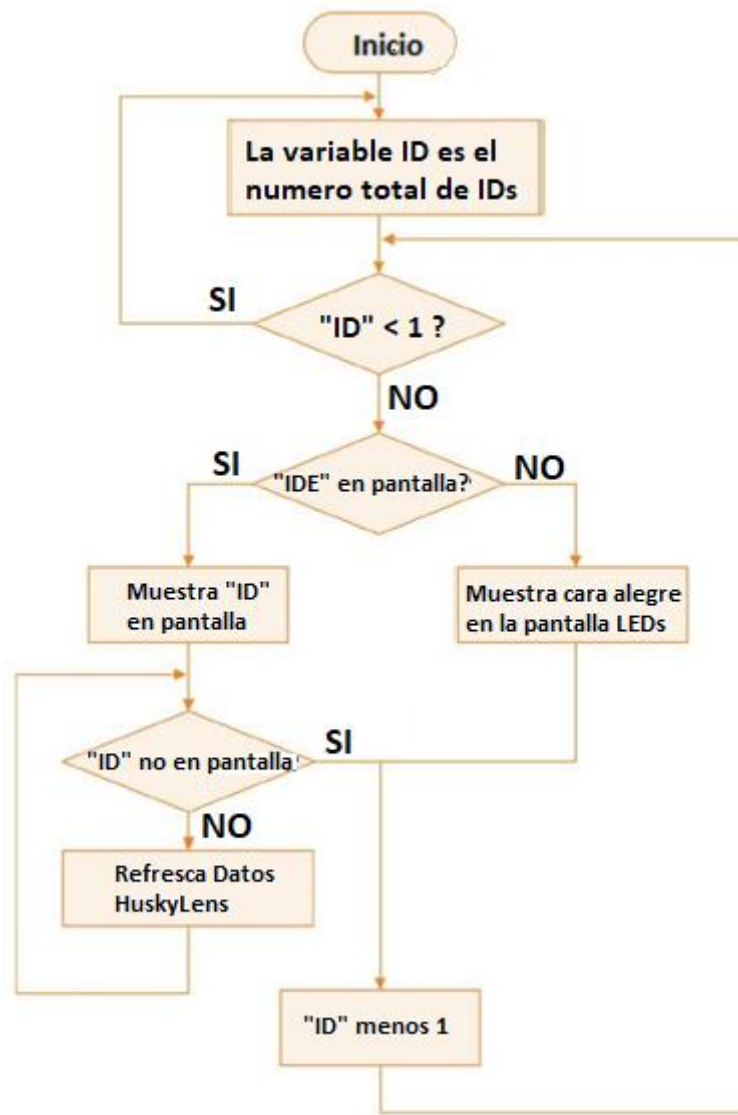
De hecho, siempre que se agregue una variable, las partes con alta repetibilidad en el programa anterior puede ser simplificado El programa es el siguiente, de modo que no importa cuántas ID se hayan aprendido, se pueden juzgar una por una.



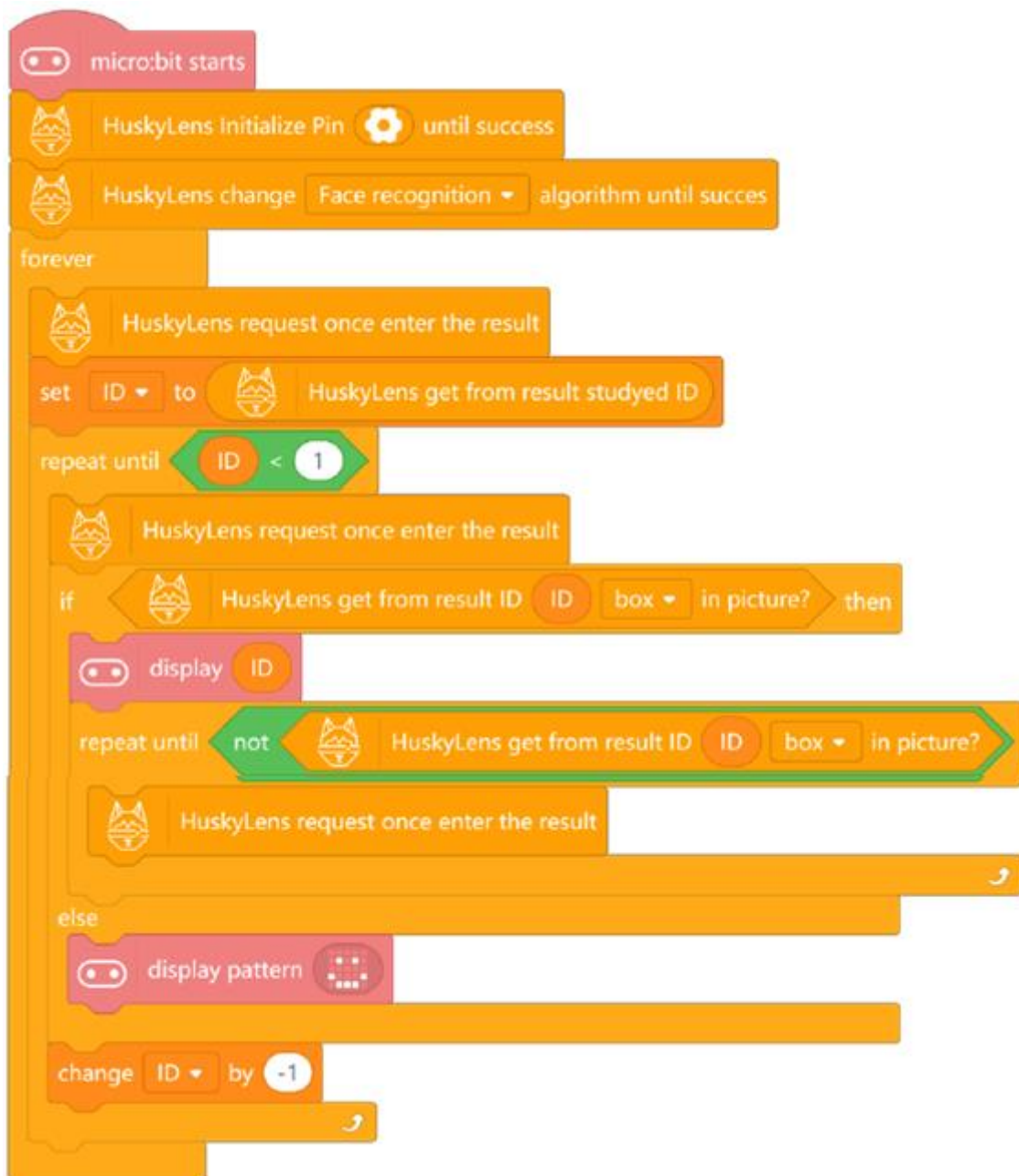
## PASO 4 Análisis de diagrama de flujo

Optimizar la función de programa, cuando no se detectan caras aprendidas, la pantalla LED muestra una cara sonriente, y cuando hay una cara aprendida, se muestra su ID.

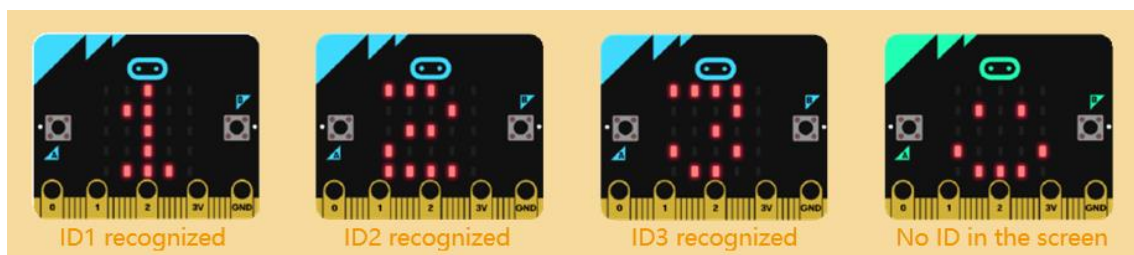
El análisis del diagrama de flujo del programa es el siguiente. La lógica puede ser un poco complicada. Puede usar el diagrama de flujo para ordenar la lógica del programa, o puede usar el programa para comprender el método de realizar la función del diagrama de flujo.



## 2. Ejemplo de programa



## 3. Resultado de la ejecución



## Tarea 2: ejecución de evitación de obstáculos

### 1. Diseño del programa

#### PASO 1 Análisis de funciones

Maqueen Plus, como detective encubierto en la gente, necesita tener una función básica de independencia conducción. ¿Cómo lidiar con los obstáculos? ¿Cómo evitar obstáculos automáticamente?

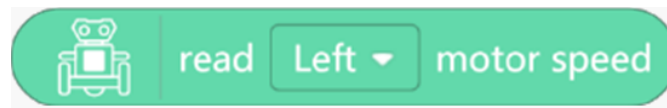
Puede pensar en cámaras para evitar obstáculos. De hecho, la función básica para evitar obstáculos no necesita una cámara, y el Maqueen Plus puede manejarla por sí mismo. ¡Echemos un vistazo!

El motor Maqueen Plus tiene su propio codificador, con el que se puede leer la velocidad del motor en tiempo real. Cuando el automóvil encuentra obstáculos durante la conducción, se ve obligado a detenerse y la velocidad del motor está cerca de 0. Por lo tanto, la velocidad del motor se puede leer para determinar si se encuentra con obstáculos. Si encuentra obstáculos, el automóvil implementará algunas acciones para evitar obstáculos, como retirarse y luego desviarse

#### PASO 2 Instrucción Aprendizaje

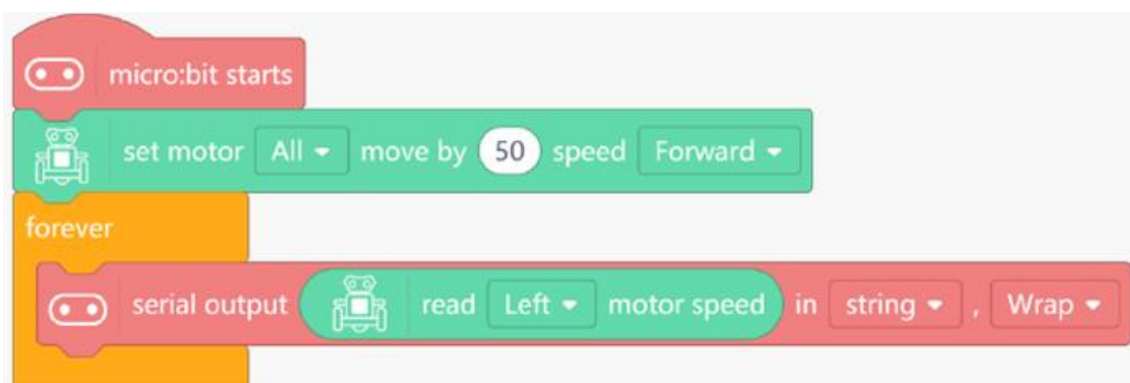
Echemos un vistazo a algunas de las instrucciones principales.

① Lea la velocidad actual del motor y seleccione la izquierda o la derecha



#### PASO 3 Programa de prueba

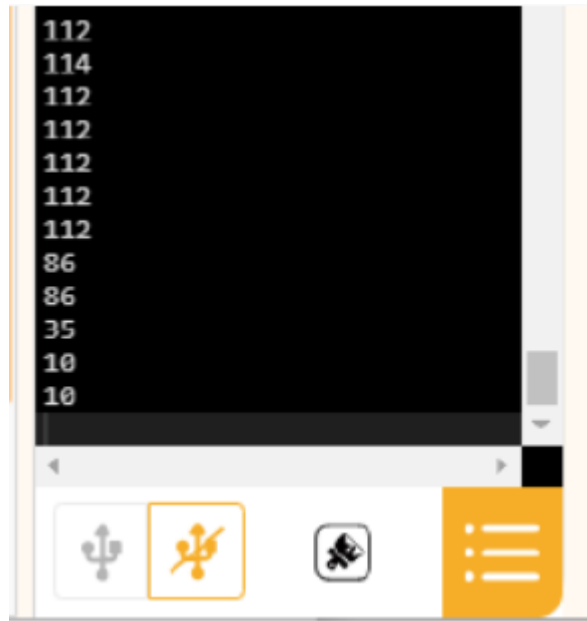
Velocidad del motor de salida a través del puerto serie. Tome el motor izquierdo como ejemplo. Programa de prueba:



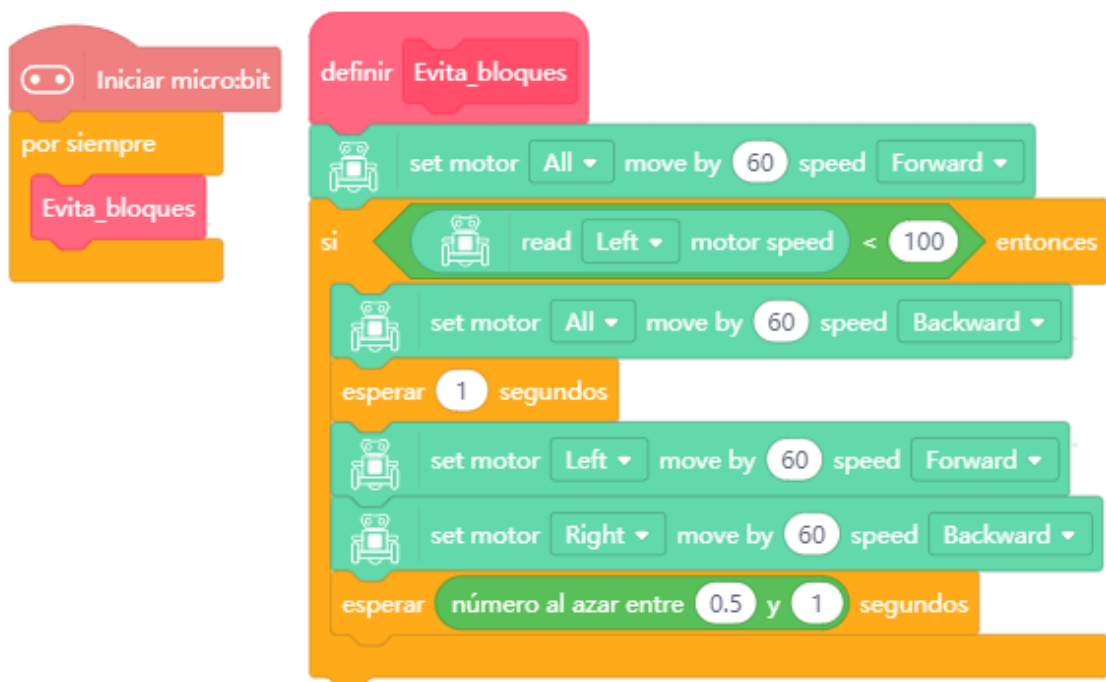
Resultados de la prueba: el área del puerto serie muestra la velocidad del motor izquierdo. Dele un poco de resistencia a la rueda izquierda con la mano, y la velocidad del motor puede obviamente reducirse.



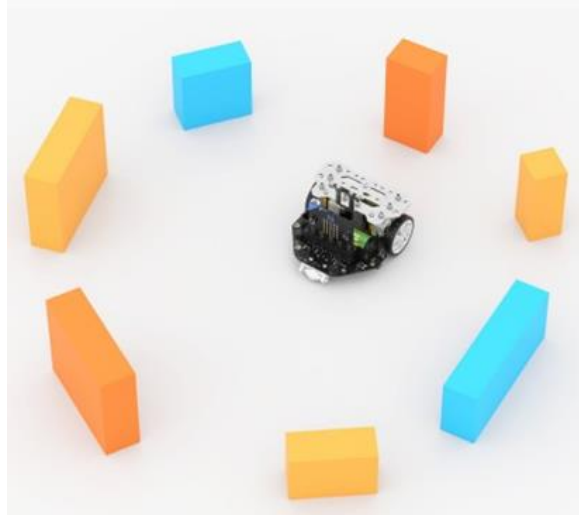
\* cuando miras la velocidad, encontrarás que la velocidad leída por el motor tiene una cierta desviación de la velocidad establecida por el programa, que está relacionada con la potencia de la batería y la fuerza de fricción en las ruedas. ¿Todavía recuerdas el control PID aprendido en el último proyecto? De hecho, PID es ajustar la velocidad en tiempo real a través del algoritmo después de leer la velocidad del motor a través del codificador, de modo que la velocidad del motor sea lo más cercana posible a la velocidad establecida.



## 2. Ejemplos de programa



## 3. Resultados de la ejecución



Maqueen Plus evitará obstáculos y correrá automáticamente. Al encontrar obstáculos, Maqueen

Además se retirará y cambiará de dirección para evitar obstáculos.

\* Cuando se inicia el programa, ya que la velocidad del motor no aumenta inmediatamente, la acción de evitar obstáculos se ejecutará primero. Esto solo sucedió cuando recién se inició.

### Tarea 3: alerta si se reúne con una persona mala

#### 1. Programación

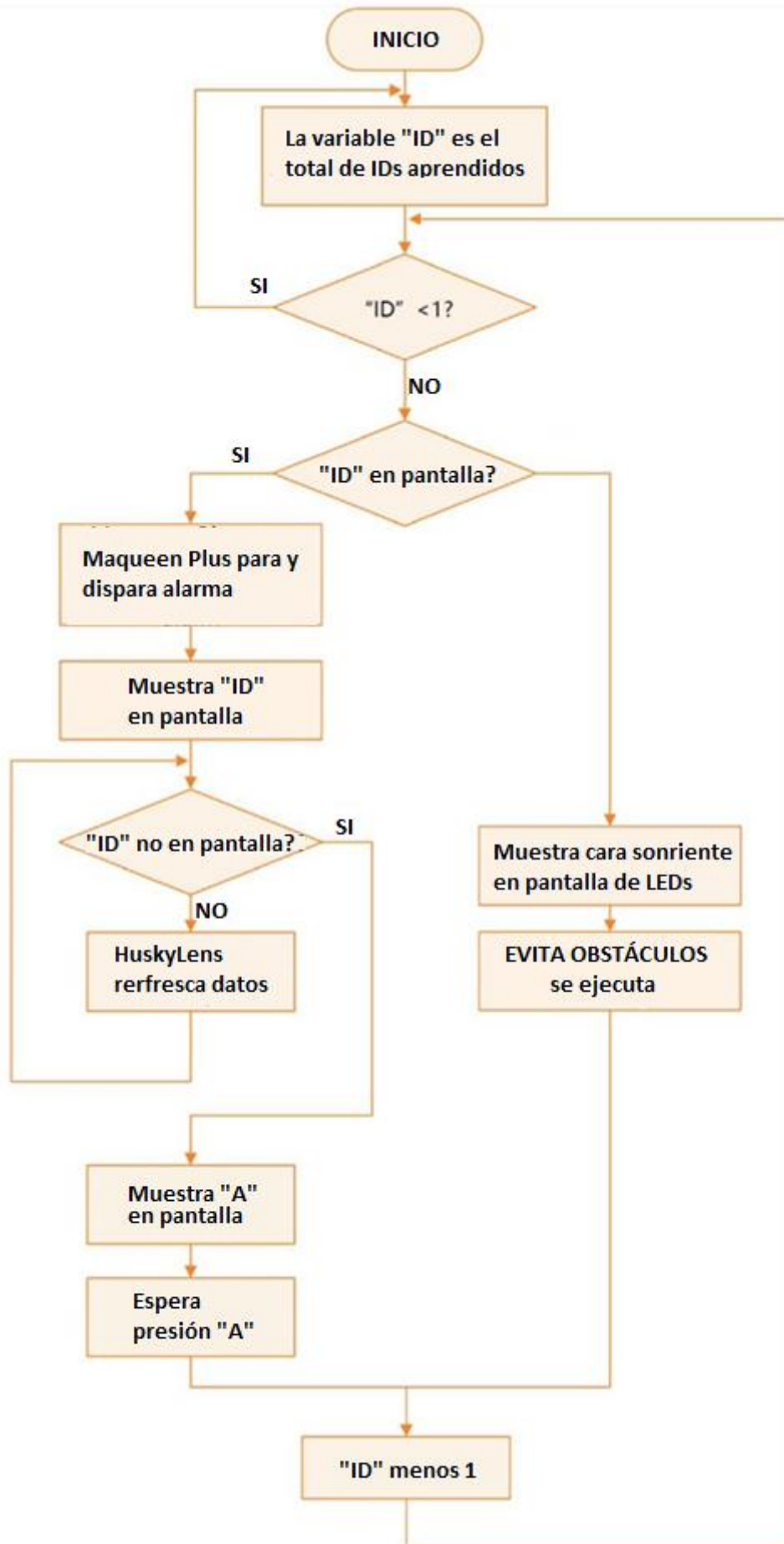
##### PASO 1 Análisis de funciones

En la tarea 1 implementamos la función de reconocimiento facial, y en la tarea 2 implementamos la función prevención de obstáculos. ¡Combinando las dos funciones y es lo que queremos para el detective encubierto!

Cuando HuskyLens no reconoció la cara designada, Maqueen Plus siguió evitando obstáculos. Cuando se reconoce la cara designada, Maqueen Plus deja de moverse, activa una alarma in situ y muestra la identificación de la cara reconocida en la pantalla de la placa principal.

##### PASO 2 Análisis de diagrama de flujo

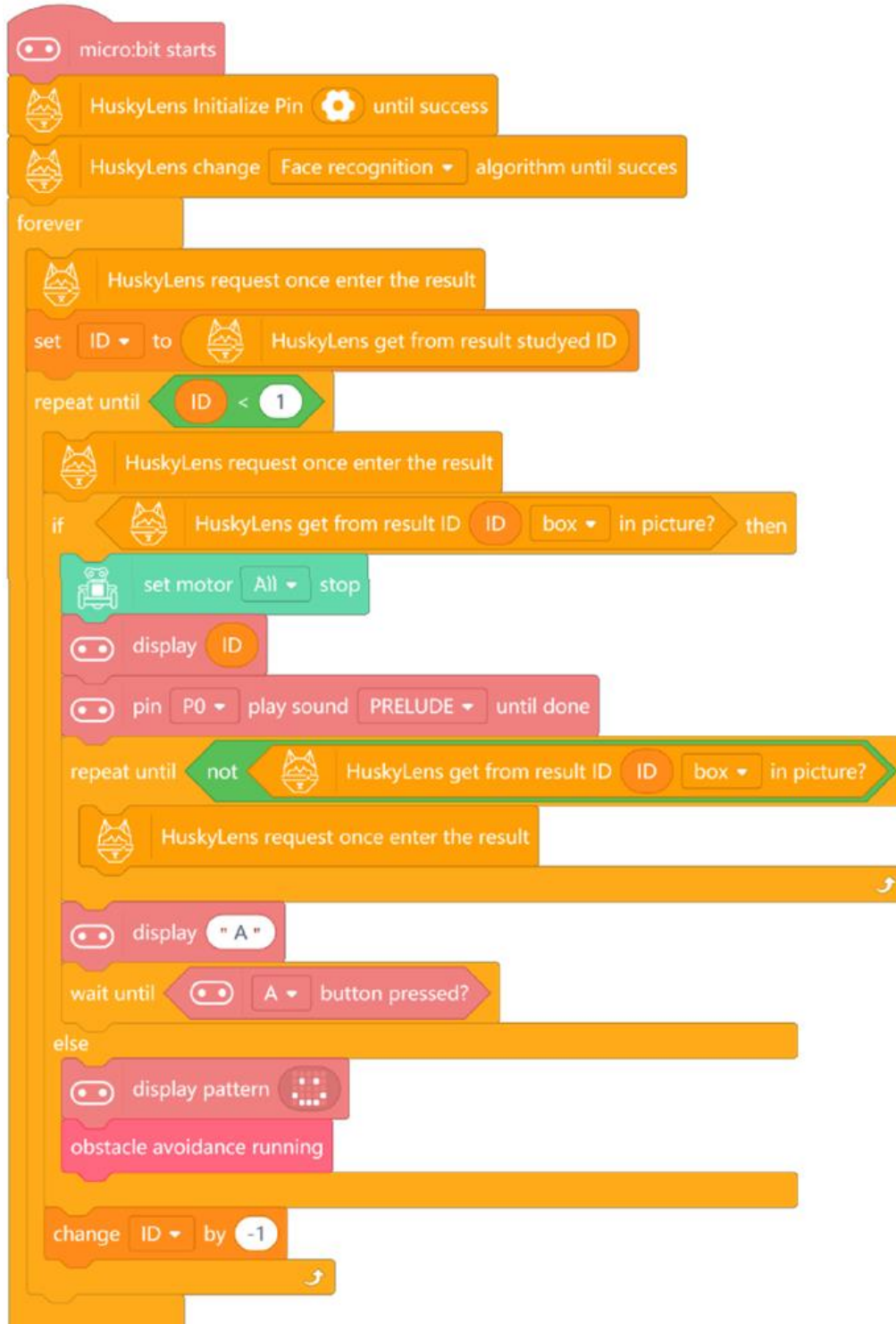
Para mantener el programa ejecutándose continuamente, después de reconocer la cara, si la cara es no durante más tiempo en la imagen, presione la tecla A en el tablero de control principal para reiniciar el programa. El diagrama de flujo es el siguiente:

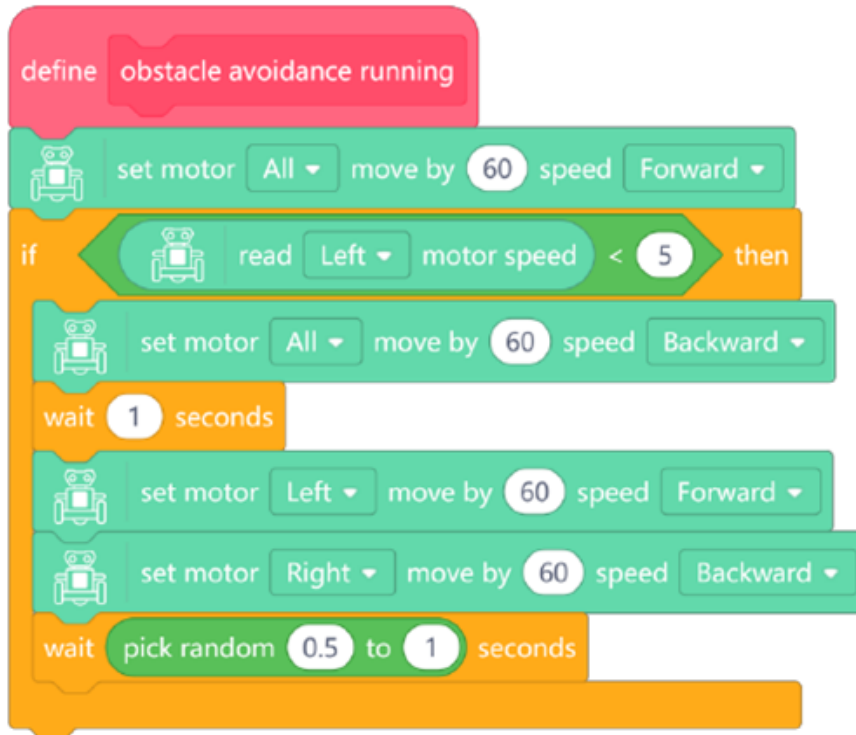


## 2. Programas ejemplo

### El programa

La función "Evitar obstáculos" no cambia y el programa principal se modifica. El programa completo es el siguiente.





### 3. Resultado de ejecución

Primero, deje que HuskyLens aprenda varias caras por la función de reconocimiento facial. Al ejecutar el programa, coloque

Maqueen Plus en cualquier posición y el automóvil evitará automáticamente los obstáculos.

Si HuskyLens ve la cara aprendida durante la conducción, el automóvil deja de moverse, activa una alarma y la pantalla de la placa principal muestra la identificación de la cara correspondiente. Después de eso, cuando HuskyLens no detecta ninguna cara aprendida, la pantalla de la placa principal muestra A, lo que indica que presionar el botón A puede reiniciar Maqueen Plus y continuar buscando malos.

### Resumen del proyecto

#### Revisión del proyecto

Comprenda el principio de funcionamiento del reconocimiento facial en esta lección y aprenda el método de operación del reconocimiento facial mediante HuskyLens.

La combinación de HuskyLens con el codificador de motor Maqueen Plus hace que Maqueen Plus se convierta en un detective encubierto y pueda reconocer constantemente la información facial recopilada mientras ejecuta obstáculos para evitar la conducción.

#### Resumen de nodos de conocimiento

1. Comprender el principio de funcionamiento del reconocimiento facial;

2. Aprender el método de operación de la función de reconocimiento facial de HuskyLens;
3. Aprender la función del codificador de motor Maqueen Plus

### **Extensión del proyecto**

En este proyecto, nuestro detective Maqueen Plus puede encontrar al chico malo escondido en la multitud, pero, ¿podemos seguirlo automáticamente después de capturarlos, haciendo que los malos no tengan a dónde escapar bajo el sistema de reconocimiento facial de AI?

Si cree que el rastreo (seguimiento) es difícil, hablaremos sobre cómo rastrearlos en el próximo proyecto. Sigamos aprendiendo juntos.



## Proyecto 5: Pokémon

Hablando de Pikachu, todos están familiarizados con él. Es el primer Pokémon de Ketchum, el personaje héroe de la animación "Pokémon". Después de ser domesticado por Ash Ketchum, siempre lo ha acompañado y crecido con él.

Tal compañía es realmente envidiable. Aunque no hay vida real de Pokémon, todos quieren tener un Pokémon que siempre nos pueda acompañar.

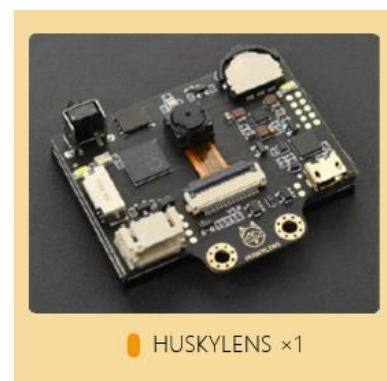
Pensemos en ello, ¿podemos crear un Pokémon por nosotros mismos?



### Descripción del proyecto

Este proyecto utiliza la **función de seguimiento de objetos** HuskyLens para convertir el Maqueen Plus en tu Pokémon exclusivo, que puede moverse con flexibilidad y siempre seguir detrás de ti

### Lista de verificación de materiales:



### Conceptos teóricos

Cuando necesitamos rastrear un objeto en movimiento, necesitamos usar tecnología de **rastreo de objetos visuales** además de operación manual. Esta tecnología es ampliamente utilizada en nuestra vida, como la *video vigilancia*, el *seguimiento de drones*, etc. Este proyecto utilizará la función de **seguimiento de objetos** HuskyLens.

## I. ¿Qué es el seguimiento de objetos?

El seguimiento de objetos es una tarea importante en la visión por ordenador. Se refiere al proceso de inferir continuamente el estado de los objetos en secuencias de video, dicho de otro modo, identificar y rastrear objetos específicos.





## II El principio de funcionamiento del seguimiento de objetos


La imagen es recolectada por una sola cámara y la información de la imagen se transmite a una computadora. Después de analizar y procesar, se calcula la posición relativa del objeto en movimiento. Al mismo tiempo, la cámara se controla para girar y rastrear el objeto en tiempo real.

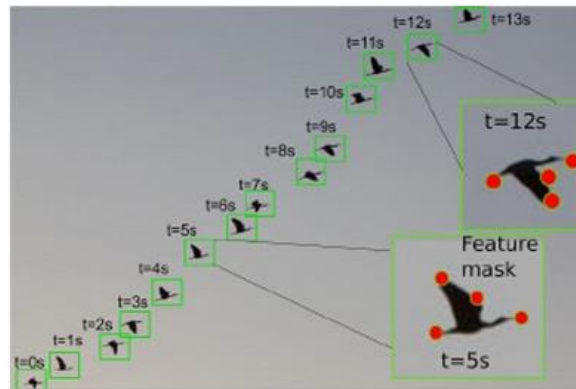



Cuando el sistema de seguimiento de objetos realiza la función de seguimiento, se divide principalmente en cuatro pasos: reconocimiento de objetos, seguimiento de objetos, predicción de movimiento de objetos y control de la cámara.

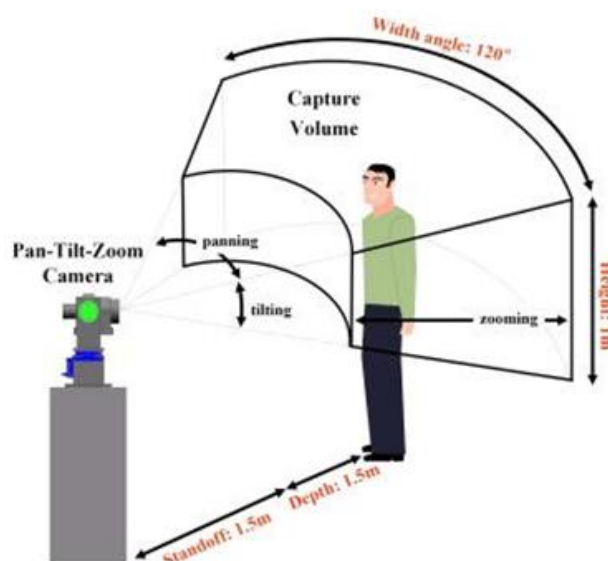
1.  **Reconocimiento de objetos:** bajo el fondo estático, se puede obtener información precisa sobre objetos en movimiento a través de algunos algoritmos de procesamiento de imágenes.
2.  **Rastreo de objetos:** de acuerdo con la posición del objeto en movimiento obtenido en el paso anterior, el algoritmo se usa para rastrear la siguiente secuencia de imágenes de acuerdo con la distribución de probabilidad de color, y se puede seguir aprendiendo en el seguimiento posterior para realizar el seguimiento cada vez más preciso.



3.  **Predicción del movimiento del objeto:** para mejorar la eficiencia, el algoritmo se usa para calcular y predecir la posición de la imagen del objeto en movimiento en el siguiente cuadro. El algoritmo puede optimizarse y la eficiencia puede mejorarse.



4.  **Control de la cámara:** mueva la cámara mientras recopila información de la imagen para ajustar la dirección de la cámara junto con la dirección de movimiento del objeto, que generalmente requiere cooperar con PTZ u otros módulos de movimiento.



### III. Campo de aplicación de seguimiento de objetos

- **Video vigilancia inteligente:** Permite el desarrollo de aplicaciones como: *Reconocimiento de movimiento* (reconocimiento humano, detección automática de objetos, etc.), *monitorización automatizada* (monitoreo de una escena para detectar comportamientos sospechosos); *monitorización de tráfico* (recopilación de datos de tráfico en tiempo real para dirigir el flujo de tráfico).



- **Interacción humano-ordenador:** la interacción tradicional humano-ordenador se lleva a cabo a través del teclado y el ratón del ordenador. Para permitir que el ordenador tenga la capacidad de reconocer y comprender las posturas, acciones y gestos de las personas, la clave es la tecnología de seguimiento.



- **Realidad virtual:** la interacción 3D y la simulación de movimiento de personajes virtuales en un entorno virtual se benefician directamente de los resultados derivados de la investigación y análisis de videos sobre el movimiento humano, que puede proporcionar a los usuarios formas más abundantes de interacción, y el análisis del seguimiento del cuerpo humano. Esta es una tecnología clave.



#### IV. Demostración de la función de seguimiento de objetos HuskyLens

HuskyLens tiene incorporada una función de **seguimiento de objetos**, que puede rastrear objetos aprendiendo sus características y retroalimentando la **información de posición** a la placa base.

A diferencia de otras funciones, como el reconocimiento de color o el reconocimiento facial, *el seguimiento de objetos puede aprender y reconocer un objeto* (o persona) por completo. El reconocimiento de color es solo para colores, mientras que la cara es solo una parte del cuerpo humano. *El seguimiento de objetos es aprender y rastrear las características generales de este objeto.*

La función de seguimiento de objetos solo puede rastrear un objeto y no admite el seguimiento de varios objetos por el momento. Es deseable que el objeto aprendido tenga contornos claros y notables para que puedan ser más fáciles de identificar.

##### 1. Seleccione la función "Seguimiento de objetos"

Marque el botón de función hacia la izquierda o hacia la derecha hasta que aparezca la palabra "*Object tracking* **Seguimiento de objetos**" en la parte superior de la pantalla.





## 2. Aprendizaje de objetos

Apunte HuskyLens al objeto de destino, ajustando la distancia y hasta que el objeto esté contenido en el cuadro delimitador naranja del centro de la pantalla. También es aceptable que solo parte del objeto esté incluido en el cuadro pero dentro de características distintas.

Luego mantenga presionado el "botón de aprendizaje" para aprender el objeto desde varios ángulos y distancias. Durante el proceso de aprendizaje, se mostrará en la pantalla el cuadro naranja con las palabras "Aprendizaje: ID1".



Cuando HuskyLens pueda rastrear el objeto en diferentes ángulos y distancias, suelte el "botón de aprendizaje" para completar el aprendizaje.

**\* Si no hay un cuadro naranja en el centro de la pantalla, significa que HuskyLens ya ha aprendido un objeto. Seleccione "Olvidar objeto aprendido" y vuelva a aprender.**

## 3. Sigue aprendiendo

Bajo la función de seguimiento de objetos, HuskyLens puede seguir aprendiendo, es decir, mientras la cámara vea el objeto aprendido, seguirá aprendiendo el estado actual del objeto, lo que es propicio para capturar objetos dinámicos.

Método de operación: mantenga presionado el botón de función para ingresar la configuración de parámetros de la función de seguimiento de objetos. Marque el botón de función hacia la derecha para seleccionar "Habilitar aprendizaje", luego presione brevemente el botón de función y márkelo hacia la derecha para activar "Habilitar aprendizaje", es decir, el icono cuadrado en la barra de progreso se gira hacia Derecha. Luego presione brevemente el botón de función para confirmar este parámetro.

## 4. Guardar el modelo

Al reiniciar HuskyLens, el último objeto aprendido no se guarda de forma predeterminada, y puede activar el interruptor para guardar los modelos automáticamente.

Método de operación: igual que el anterior, después de ingresar la configuración de parámetros, active "Guardar automáticamente". De esta manera, solo necesita aprender el objeto una vez. Al reiniciar la cámara, se guardará el objeto que aprendió la última vez.

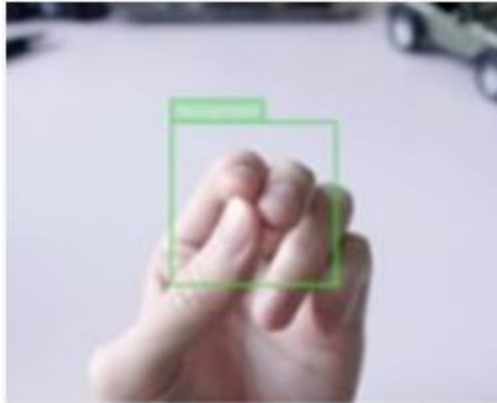
## Práctica del proyecto:

Tarea 1: Conozca el seguimiento de objetos

Diseño del programa

## PASO1 Aprendiendo y reconociendo

Aquí puede seleccionar un objeto con características de contorno obvias, como gestos



## PASO 2 Instrucción Aprendizaje

Echemos un vistazo a algunas de las instrucciones principales.

1. Parameter El parámetro de IDx se obtiene del "resultado" solicitado, y se devolverá -1 si este ID no se ha o no se ha aprendido en la pantalla

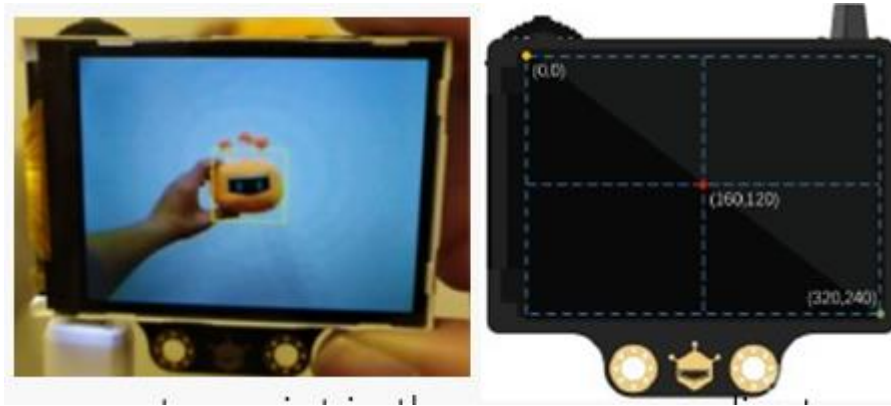


## PASO 3 Análisis de coordenadas

La resolución de la pantalla del sensor HuskyLens es 320 \* 240, como se muestra en

La siguiente imagen. Las coordenadas del punto central del objeto obtenidas a través del programa también están dentro de este rango. Por ejemplo, si los valores de coordenadas obtenidos son (160, 120), el objeto que se está rastreando está en el centro de la pantalla.





"Coordenadas X" y "Coordenadas Y" se refieren a la posición del punto central de la caja en la coordenada de la pantalla. "Ancho del objeto" y "Altura del objeto" se refieren al tamaño del marco. Debajo de la función de seguimiento de objetos,

El marco es cuadrado, por lo que el ancho y la altura son iguales.

#### PASO 4 Análisis de funciones

Verifique cada parámetro de la trama a través del puerto serie

#### 2. Programa ejemplo



### 3. Resultado de ejecución

En el área del puerto serie de Mind +, active el interruptor del puerto serie para ver los parámetros. Intente mover el objeto hacia la izquierda y hacia la derecha para observar el cambio numérico del centro X. Mueva el objeto hacia arriba y hacia abajo para observar el cambio numérico del centro Y. Mueva el objeto hacia adelante y hacia atrás para observar el cambio numérico de ancho y alto.

```

X: 162 Y: 154 width: 86 height: 86
X: 162 Y: 154 width: 86 height: 86
X: 160 Y: 161 width: 87 height: 87
X: 160 Y: 156 width: 86 height: 86
X: 160 Y: 156 width: 86 height: 86
X: 159 Y: 152 width: 85 height: 85
X: 159 Y: 152 width: 85 height: 85
X: 156 Y: 156 width: 85 height: 85
X: 138 Y: 159 width: 87 height: 87
X: 138 Y: 159 width: 87 height: 87
X: 191 Y: 111 width: 100 height: 100
X: 191 Y: 111 width: 100 height: 100
X: 184 Y: 106 width: 100 height: 100
X: 174 Y: 95 width: 98 height: 98
X: 174 Y: 95 width: 98 height: 98
X: 167 Y: 81 width: 99 height: 99
X: 158 Y: 60 width: 97 height: 97
X: 154 Y: 48 width: 97 height: 97

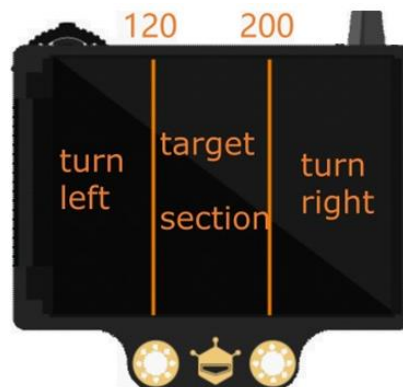
```

## Tarea 2: ajustar izquierda y derecha

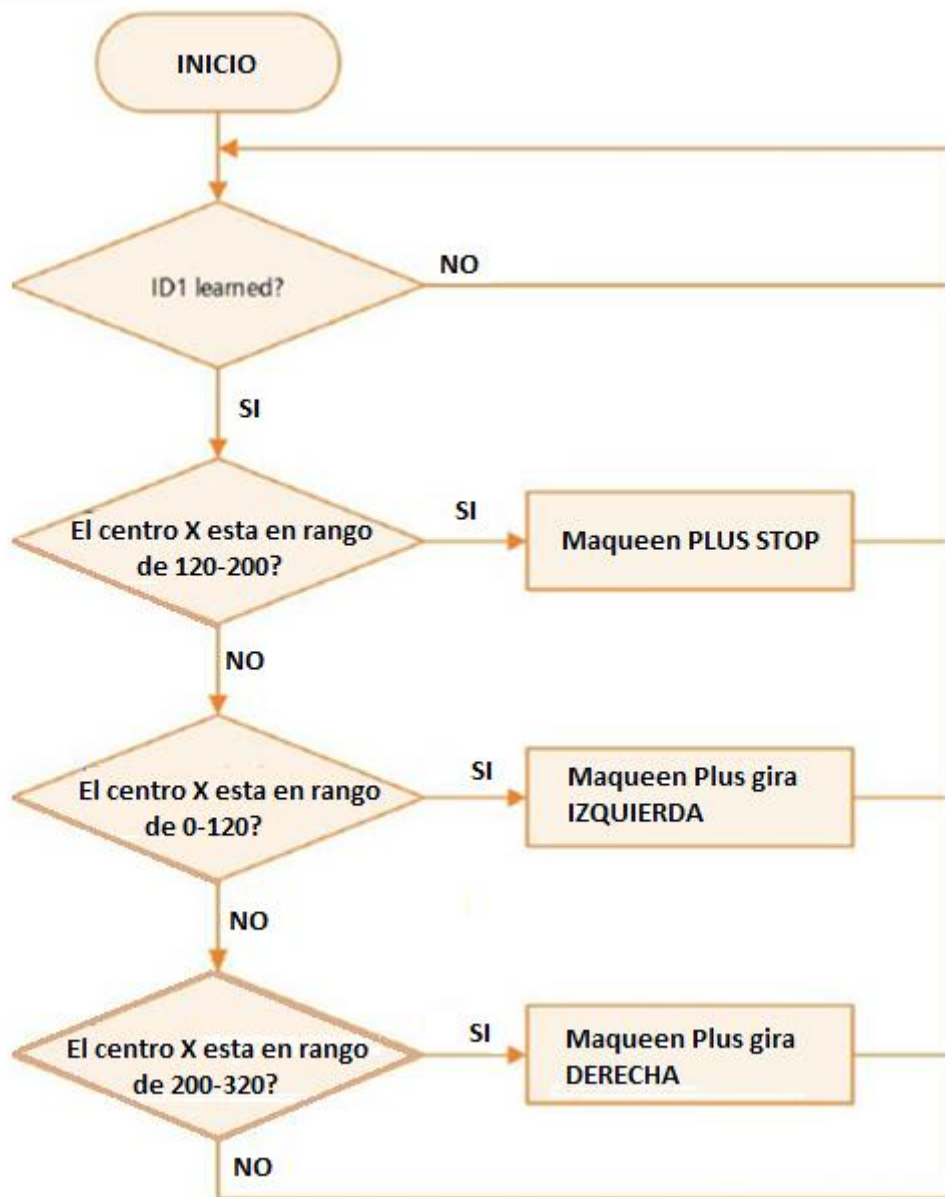
### 1. Diseño del programa

#### PASO 1 Análisis de funciones

Como se muestra en la siguiente imagen, la pantalla se divide en 3 secciones de acuerdo con el eje X del sistema de coordenadas de la pantalla de la cámara, y la sección central es nuestra sección objetivo. Cuando la cámara detecta continuamente el estado del objeto objetivo en la imagen, su centro X es 120-200, lo que significa que el objetivo está en el centro del campo de visión y Maqueen Plus no necesita ajustar su posición; su centro X es 0-120, Maqueen Plus necesita ajustarse girando a la derecha; su centro X es 200-320, Maqueen Plus necesita girar a la izquierda para ajustar.



## PASO 2 Análisis de diagrama de flujo



## 2. Programa Ejemplo



```
micro:bit starts
  set motor All stop
  HuskyLens Initialize Pin until success
  HuskyLens change Object tracking algorithm until success
  forever
    State detection

define Adjust Left and Right
  if X > 0 and X < 120 then
    set motor Left move by 30 speed Forward
    set motor Right move by 50 speed Forward
  if X > 200 and X < 320 then
    set motor Left move by 50 speed Forward
    set motor Right move by 30 speed Forward

define State detection
  HuskyLens request once enter the result
  if HuskyLens get from result ID 1 have learned? then
    if HuskyLens get from result ID 1 box in picture? then
      set X to HuskyLens get from result ID 1 box parameter X coordinates
      if X >= 120 and X <= 200 then
        set motor All stop
      else
        Adjust Left and Right
    else
      set motor All stop
```

The image displays a Scratch script for a Maqueen Plus robot using HuskyLens for object tracking. The script is organized into three main sections: initialization, a movement adjustment function, and a state detection function.

- Initialization:** The script starts with a "micro:bit starts" block, followed by "set motor All stop", "HuskyLens Initialize Pin until success", and "HuskyLens change Object tracking algorithm until success". A "forever" loop contains a "State detection" block.
- Adjust Left and Right:** This function block handles two specific X-coordinate ranges. If  $X > 0$  and  $X < 120$ , it sets the left motor to move by 30 and the right motor to move by 50, both in the forward direction. If  $X > 200$  and  $X < 320$ , it sets the left motor to move by 50 and the right motor to move by 30, both in the forward direction.
- State detection:** This function block requests data from HuskyLens. It checks if the robot has learned an object. If yes, it checks if the object's box is in the picture. If so, it retrieves the X-coordinate. If the X-coordinate is between 120 and 200, it stops all motors. Otherwise, it calls the "Adjust Left and Right" function. If the object is not in the picture, it stops all motors.

### 3. Resultado de ejecución

Después de que Maqueen Plus termine de aprender un objeto, lo seguirá automáticamente para moverse hacia adelante, hacia atrás, hacia la izquierda y hacia la derecha, manteniendo la caja de objetos en el centro de la pantalla y a una distancia adecuada.

#### 1. Resumen del proyecto:

##### Revision del proyecto

¿Comprende el principio de funcionamiento del **seguimiento de objetos** y el método de operación del **reconocimiento facial**? mediante el uso de HuskyLens.

El sensor HuskyLens puede generar el centro X, el centro Y, la altura y el ancho del cuadro de destino. Estos parámetros se pueden usar para juzgar la posición relativa y la distancia del objeto objetivo.

Cuando el Maqueen Plus se usa como un auto de rastreo, estos parámetros pueden ayudar a localizar el objeto objetivo, que todavía es válido bajo otras funciones de HuskyLens. Por ejemplo, puede convertir este proyecto en un seguidor de rostros humanos y dejar que Maqueen Plus siga el reconocido rostro humano.

##### Resumen de nodos de conocimiento

1. Comprender el principio de funcionamiento del seguimiento de objetos;
2. Aprenda el método operativo de la función de seguimiento de objetos HuskyLens;
3. Aprenda a usar HuskyLens para que Maqueen Plus siga el objetivo.

##### Extensión de proyecto:

Al igual que la animación en la que el personaje héroe domestica a su Pokémon para hacerlo más fuerte, ¿puedes hacer que tu Maqueen Plus tiene más funciones? Por ejemplo, agregue control de gestos. Cuando su mano dibuja un círculo a la izquierda, el automóvil girará a la izquierda. Cuando tu mano empuja hacia adelante, el auto se moverá hacia adelante. ¿Se puede implementar con la programación?



## Proyecto 6: Siguiendo el "camino correcto"

Hoy en día los robots son cada vez más poderosos. Pueden hacer casi cualquier cosa en lo alto de la montaña tranquilo y profundo en el mar. ¡Míralos! Algunos de ellos sirven bien como camareros en restaurantes. Algunos trabajan duro como "**cohermanos** (dispositivos que colaboran)" en talleres y/o fábricas. Algunos trabajan como inspectores de seguridad en centros de la red eléctrica con un alto grado de responsabilidad ...



Si observa detenidamente, encontrará que hay líneas en el terreno donde trabajan estos robots. Y ellos están siguiendo las líneas. Seguimiento de línea? ¿Cómo lo hicieron?

HuskyLens también tiene la función de seguimiento de línea. Imagínese, si algunas líneas de guía se pegan en casa, puede Maqueen

¿Además te conviertes en nuestro asistente de vida como el robot de entrega de comida en el restaurante robot?

¡Exploremos por bricolaje un Maqueen Plus que sigue el "camino correcto"!

### Descripción del proyecto:

Este proyecto utiliza la función de seguimiento de línea HuskyLens para hacer que Maqueen Plus siga el seguimiento de línea en el suelo.

### Lista de materiales:



### Conceptos teóricos


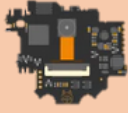
Si queremos que Maqueen Plus se mueva de acuerdo con las líneas en el suelo, necesitamos algunos sensores para identificar las líneas. Según los diferentes sensores, existen varios tipos de métodos de seguimiento de línea. Este proyecto utiliza la función de seguimiento de línea del sensor visual HuskyLens para implementar el efecto de seguimiento de línea.

#### I. ¿Qué es el seguimiento de línea?

El seguimiento de línea se refiere al proceso de movimiento de objetos a lo largo de una ruta designada. Un robot de rastreo de línea completamente funcional utiliza un robot móvil como portador, una cámara de luz visible, una cámara termográfica infrarroja y otros instrumentos de detección como un sistema de carga, una fusión de información de campo múltiple de visión artificial, campo electromagnético, GPS y SIG. como sistema de navegación para el movimiento autónomo y el seguimiento del robot, y un ordenador integrado como plataforma de desarrollo de software y hardware para el sistema de control

### II La comparación de dos métodos de seguimiento de línea comúnmente utilizados

En la tabla siguiente se comparan las características de los dos sistemas más comunes de seguimiento de líneas en una ruta (mapa).

Comparación de Productos	Sensor de línea por infrarrojos 	Sensor Visual 
<b>Costo</b>	Bajo	Alto
<b>Rango de visión</b>	El sensor debe estar cerca del suelo y tiene un rango de visión pequeño.	El rango de visión es amplio, y el movimiento se puede ajustar el estado de antemano de acuerdo con los cambios.
<b>Adaptación al medio ambiente</b>	Cuando se cambia el entorno de uso, la sensibilidad del sensor debe ajustarse y el proceso de ajuste es complicado.	Cuando se cambia el entorno de uso, solo es necesario volver a aprender las líneas y la operación es simple.
<b>Adaptación del mapa</b>	Generalmente solo es adecuado para mapas simples con líneas de fondo claras, líneas en blanco y negro o líneas continuas	Adecuado para mapas con líneas de fondo claras, líneas multicolores, líneas continuas, líneas punteadas y otras condiciones complejas.

### III. El principio de la función de seguimiento de línea

La función de seguimiento de línea de HuskyLens se basa en **Pixy**, un proyecto de código abierto de la **Universidad Carnegie Mellon**.

El algoritmo de Pixy puede reconocer el color de las imágenes. Su idea básica es utilizar el espacio de color para eliminar el fondo en el que no están interesados todos los usuarios y extraer el primer plano (como las líneas).



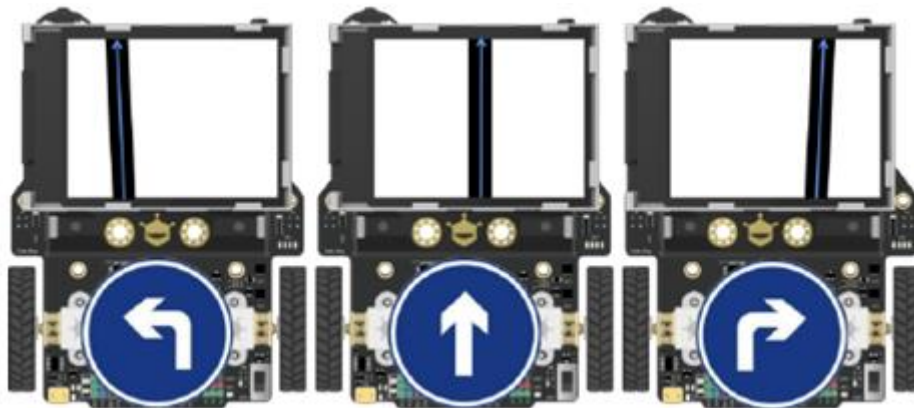


Líneas/fondo, extraer líneas y barras de color, eliminar el color de fondo

¿Cómo puede Maqueen Plus seguir la línea negra en el mapa de seguimiento (que tiene líneas negras sobre fondo blanco)? En

De hecho, solo necesitamos conocer la posición relativa de Maqueen Plus con respecto a la línea negra. Se puede dividir en las siguientes tres situaciones:

1. Cuando Maqueen Plus está en el lado derecho del línea negra, controle Maqueen Plus para girar a la izquierda;
2. Cuando Maqueen Plus está relativamente alineado al centro con la línea negra, controle Maqueen Plus para ir derecho;
3. Cuando Maqueen Plus esté en el lado izquierdo de la línea negra, controle Maqueen Plus para girar a la derecha.



¿Cómo se debe implementar esto? Eliminamos la información que HuskyLens muestra en la pantalla durante el seguimiento de línea y la resumimos en el modelo matemático geométrico que se muestra en la imagen a continuación.

La resolución de la pantalla HuskyLens es **320×240**. El punto **O** en la esquina superior izquierda de la pantalla es el origen de las coordenadas de la pantalla **(0, 0)**, la dirección horizontal derecha es la dirección positiva del eje **X** y la dirección vertical hacia abajo es la dirección positiva del eje **Y**, por lo que las coordenadas en la esquina inferior derecha de la pantalla son **(320,240)**. La línea naranja punteada en la imagen de arriba es el eje central de la pantalla, y el valor de abscisa de esta línea es 160. la pantalla de arriba es la línea del mapa "vista" por la cámara HuskyLens. La flecha azul es la dirección de la línea calculada por HuskyLens. Las coordenadas del punto inicial de la flecha azul son  $(x_1, y_1)$  y las coordenadas del punto final son  $(x_2, y_2)$ .

Para resumir, solo necesitamos juzgar la posición del punto de partida de la flecha azul con respecto al eje central para implementar el seguimiento de línea.

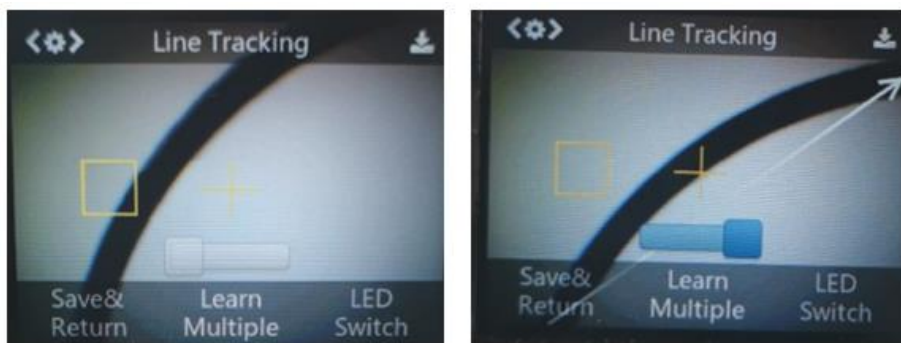
#### IV. Demostración de la función de seguimiento de línea HuskyLens

##### 1. Aprendizaje de objetos

Esta función puede rastrear líneas de colores especificados y hacer predicciones de ruta. La configuración predeterminada es rastrear líneas de un solo un color. Este proyecto se explicará tomando como ejemplo el seguimiento de línea de un color.

### Operación y configuración:

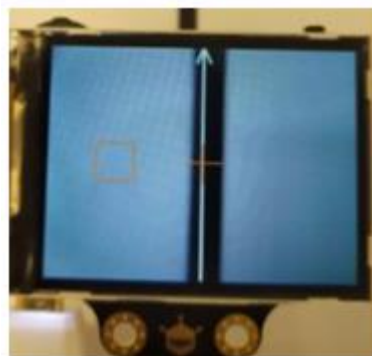
1. Marque el botón de función hacia la izquierda o hacia la derecha hasta que aparezca la palabra "Seguimiento de línea" en la parte superior de la pantalla.
2. Mantenga presionado el botón de función para ingresar la configuración de parámetros de la función de seguimiento de línea.
3. Marque el botón de función hacia la derecha o hacia la izquierda hasta que se seleccione "Aprender múltiples", luego presione brevemente el botón de función y márkelo hacia la izquierda para desactivar el interruptor "Aprender múltiples", es decir, el icono cuadrado en la barra de progreso está giró a la izquierda. Luego presione brevemente el botón de función para completar este parámetro.
4. También puede encender el LED configurando "Interruptor LED". Esto es muy útil en el ambiente oscuro. Referente con el método anterior, encienda el "Interruptor LED".



5. Marque el botón de función hacia la izquierda hasta que se seleccione "Guardar y volver", y presione brevemente el botón de función para guarde los parámetros y volverá automáticamente.

### 2. Aprendizaje y seguimiento

**Aprendizaje en línea:** Apunte el símbolo "+" a la línea, luego apunte el cuadro naranja en la parte posterior de fondo. Se recomienda que dentro del campo de visualización de HuskyLens, solo permanezcan líneas para aprender y no haya líneas cruzadas. Intente mantener HuskyLens paralelo a la línea objetivo, entonces HuskyLens detectará automáticamente la línea y aparecerá una flecha blanca. Luego presione brevemente el "botón de aprendizaje" y la flecha blanca se convertirá en una flecha azul.



**Aprendizaje en línea:** Apunte el símbolo "+" a la línea, luego apunte el cuadro naranja al área de fondo. Se recomienda que dentro del campo de visualización de HuskyLens, solo

permanezcan líneas para aprender y no haya líneas cruzadas. Intente mantener HuskyLens paralelo a la línea de destino, entonces HuskyLens detectará automáticamente la línea y aparecerá una flecha blanca. Luego presione brevemente el "**botón de aprendizaje**" y la flecha blanca se convertirá en una flecha azul.



1. Al aprender la línea, necesitamos ajustar la posición de HuskyLens para que sea paralela a la línea.
2. HuskyLens puede aprender la línea en cualquier color que tenga un contraste de color obvio con el fondo, pero estas líneas deben ser líneas monocromas con un contraste de color obvio del fondo para mantener estable el seguimiento de línea.
3. HuskyLens puede aprender y rastrear varias líneas de acuerdo con los diferentes colores de las líneas, pero las líneas deben ser líneas monocromáticas con un contraste de color obvio del fondo. En la mayoría de los casos, solo use la línea de un color durante el seguimiento de línea. Por lo tanto, para garantizar la estabilidad, recomendamos seguir la línea en un solo color.
4. El color de las líneas tiene mucho que ver con la luz ambiental. Al patrullar la línea, intente mantener la luz ambiental lo más estable posible.

### Práctica del proyecto:

Implementaremos el proyecto de acuerdo con la lógica de seguimiento de línea y optimizaremos continuamente el seguimiento de línea efecto en varios pasos para que Maqueen Plus pueda pasar el mapa de forma rápida y estable. Primero, aprenderemos a usar la función de seguimiento de línea de HuskyLens, leer los datos de abscisa de las líneas, escribir un programa simple para ajustar el estado de movimiento para cumplir con los requisitos de seguimiento de línea y luego mejorar nuestro programa de proyecto de acuerdo con el resultado de la depuración.

### Tarea 1: Algoritmo de seguimiento de línea 1

La siguiente imagen es un mapa de seguimiento de línea de referencia.



## 1. Construcción de la estructura

Usando tornillos para fijar HuskyLens y Maqueen Plus, debe tenerse en cuenta que nosotros necesitamos ajustar la cámara oblicuamente hacia abajo para el seguimiento de línea, de modo que Maqueen Además, puede ver la línea negra más cerca y el efecto de seguimiento sería mejor.



## 2. Diseño del programa

Cuando HuskyLens detecta que la línea negra está en tu lado izquierdo de la pantalla, es decir, el valor de abscisa  $x1 < 160$  desde el punto de partida de la flecha azul, controle Maqueen Plus para girar a la izquierda; Cuando la línea negra está en el lado derecho de la pantalla,  $x1 > 160$ , controle Maqueen Plus para girar a la derecha; Cuando la línea negra se encuentra en el centro de la pantalla,  $x = 160$ , controla Maqueen Plus para ir en línea recta.



### 3. Ejemplo de programa



### 3. Resultado de ejecución

HuskyLens aprende botella, bicicleta y silla bajo la función de reconocimiento de objetos. Cuando el programa se esté ejecutando, coloque Maqueen Plus en el punto de partida. Cuando HuskyLens reconoce la botella, Maqueen Plus rastreará la línea hasta el punto terminal 1. Cuando HuskyLens reconozca la bicicleta, Maqueen Plus rastreará la línea hasta el punto terminal 2. Cuando HuskyLens reconozca la silla, Maqueen Plus rastreará la línea hasta el punto terminal 3.

#### Resumen del proyecto:

#### Revisión del proyecto

Comprenda el principio de funcionamiento del reconocimiento de objetos en esta lección y el método de operación del reconocimiento de objetos mediante el uso del sensor HuskyLens AI.

HuskyLens se combina con el sensor de seguimiento de línea incorporado Maqueen Plus, transformando Maqueen Plus en una “Sorteador de obstáculos Inteligente” que es capaz de clasificar y transportar de manera automatizada.

Imagine que Maqueen Plus pudiera cargar o descargar carga con un brazo mecánico, toda la aplicación sería más inteligente.

## Resumen de nodos de conocimiento

1. Comprender el principio de funcionamiento del reconocimiento de objetos.
2. Aprenda el método operativo de la función de reconocimiento de objetos de HuskyLens.
3. Aprenda el control de seguimiento de línea de Maqueen Plus.

## Extensión de proyecto:

En este proyecto, implementamos la función de clasificación y transporte, pero ¿cómo podemos volver al punto de partida después de que Maqueen Plus es transportado a la ubicación designada? ¿Puede Maqueen Plus continuar rastreando la línea y regresar? Intenta implementarlo con un programa.

## 4. Resultado de la ejecución

Bajo la función de seguimiento de línea HuskyLens, aprenda la línea y el fondo, y la función de seguimiento de línea funcionará siempre como la cámara hacia abajo.

Maqueen Plus puede completar las tareas básicas de seguimiento de línea, pero al mismo tiempo también expone los siguientes problemas:

1. Maqueen Plus obviamente se balancea hacia la izquierda y hacia la derecha mientras avanza, los cambios de velocidad no son consistentes y el movimiento recto no es estable;
2. La velocidad no puede ajustarse demasiado rápido o es fácil desviarse en la esquina;
3. Las diferentes curvas requieren diferentes velocidades de giro. Cuando hay varias curvas en el mapa de seguimiento, es fácil desviarlo.

**\* Si Maqueen Plus sale de la línea durante el giro, necesita ajustar la velocidad de los motores izquierdo y derecho continuamente.**

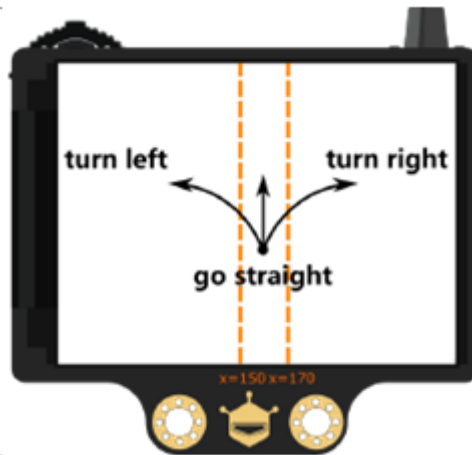
¿Cómo optimizarlo? ¡Echemos un vistazo!

## Tarea 2: Algoritmo de seguimiento de línea 2

### 1. Diseño del programa

En la Tarea 1, Maqueen Plus se balancea hacia la izquierda y hacia la derecha en su movimiento hacia adelante y no puede moverse constantemente en línea recta. ¿Por qué pasa esto? Debido a que el dominio del eje horizontal del movimiento recto Maqueen Plus es solo una línea, y hay inercia en el proceso de movimiento. Por lo tanto, mantenerse en movimiento en  $x = 160$  es difícil de lograr.



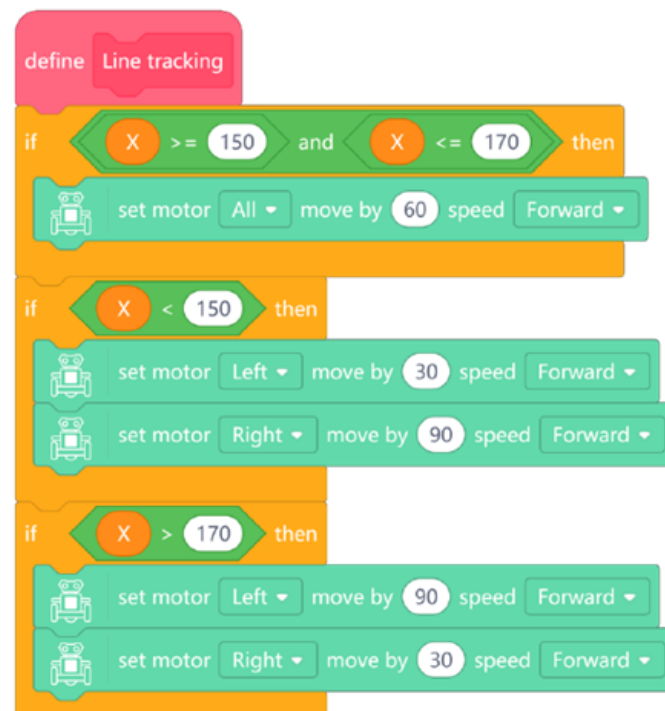


La idea de optimización es expandir el dominio de movimiento directo. Como se muestra en la siguiente imagen, configuramos el dominio  $[150,170]$  como el dominio de movimiento recto Maqueen Plus. Cuando la coordenada de punto de partida  $x_1$  está dentro de este dominio, controle el Maqueen Plus para ir en línea recta. Cuando  $x_1 < 150$ , controla Maqueen Plus para girar a la izquierda; Cuando  $x_1 > 170$ , controla Maqueen Plus para girar a la derecha.

## 2. Ejemplo de programa

### El programa

Sobre la base del programa de la tarea 1, el programa principal no cambia y la función "seguimiento de línea" se modifica de la siguiente manera



### 3. Resultado de la ejecución

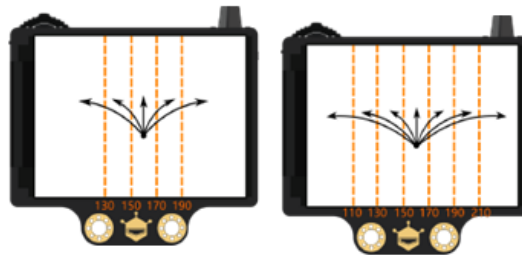
A través de estos ajustes, encontramos que el efecto de seguimiento de línea es mejor, y cuando encuentra un arco, puede casi permanecer en línea. Sin embargo, aún es fácil desviarse en el seguimiento de mapas con grandes cambios de curva.

#### Tarea 3: Algoritmo de seguimiento de línea 3

Intente evitar que el automóvil se desvíe cuando ingrese a la curva.

##### 1. Diseño del programa

Sigamos ajustándonos. Como podemos dividir el negro detectado línea en tres dominios, ¿podemos continuar dividiendo los dominios en cinco como lo hicimos?



Cuanto más cerca esté la línea negra de los extremos, mayor será la velocidad de giro. Cuanto más cerca está la línea negra del centro, más tiende a moverse en línea recta.

O, además, ¿se puede dividir directamente en 7 dominios de regulación de velocidad?

##### 2. Ejemplo de programa

Sobre la base del programa de la tarea 2, el programa principal permanece sin cambios y la función "línea tracking" se modifica de la siguiente manera.

The image shows a Scratch-style code editor with a function definition for "Line tracking". The function uses a series of conditional blocks to control a robot's motors based on the value of a variable 'X'.

```
define Line tracking
if X >= 150 and X <= 170 then
  set motor All move by 60 speed Forward
if X >= 130 and X < 150 then
  set motor Left move by 40 speed Forward
  set motor Right move by 80 speed Forward
if X >= 110 and X < 130 then
  set motor Left move by 40 speed Forward
  set motor Right move by 100 speed Forward
if X < 110 then
  set motor Left move by 40 speed Forward
  set motor Right move by 120 speed Forward
if X > 170 and X <= 190 then
  set motor Left move by 80 speed Forward
  set motor Right move by 40 speed Forward
if X > 190 and X <= 210 then
  set motor Left move by 100 speed Forward
  set motor Right move by 40 speed Forward
if X > 210 then
  set motor Left move by 120 speed Forward
  set motor Right move by 40 speed Forward
```

The code consists of the following blocks:

- define Line tracking** (pink block)
- if**  $X \geq 150$  and  $X \leq 170$  **then** (orange block)
  - set motor** All move by 60 speed Forward (green block)
- if**  $X \geq 130$  and  $X < 150$  **then** (orange block)
  - set motor** Left move by 40 speed Forward (green block)
  - set motor** Right move by 80 speed Forward (green block)
- if**  $X \geq 110$  and  $X < 130$  **then** (orange block)
  - set motor** Left move by 40 speed Forward (green block)
  - set motor** Right move by 100 speed Forward (green block)
- if**  $X < 110$  **then** (orange block)
  - set motor** Left move by 40 speed Forward (green block)
  - set motor** Right move by 120 speed Forward (green block)
- if**  $X > 170$  and  $X \leq 190$  **then** (orange block)
  - set motor** Left move by 80 speed Forward (green block)
  - set motor** Right move by 40 speed Forward (green block)
- if**  $X > 190$  and  $X \leq 210$  **then** (orange block)
  - set motor** Left move by 100 speed Forward (green block)
  - set motor** Right move by 40 speed Forward (green block)
- if**  $X > 210$  **then** (orange block)
  - set motor** Left move by 120 speed Forward (green block)
  - set motor** Right move by 40 speed Forward (green block)

### 3. Resultado de la ejecución

La velocidad de seguimiento de línea de Maqueen Plus ahora puede ser más rápida, y el cambio de velocidad es mucho más suave sin importar si gira o se mueve en línea recta.

**Resumen del proyecto:**

**Revisión del proyecto**

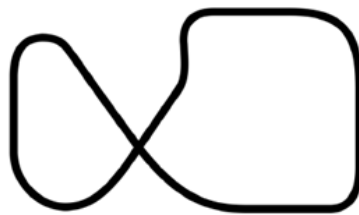
En este proyecto, implementamos el algoritmo de seguimiento de línea de simple a complejo, haciendo que el efecto de seguimiento de línea sea más rápido y más estable.

Resumen de nodos de conocimiento

1. Aprenda las ideas principales del seguimiento de línea
2. Aprenda el método de operación de la función de seguimiento de línea HuskyLens;
3. Método de optimización del algoritmo de seguimiento de línea

**Extensión de proyecto:**

Después de completar el seguimiento de la línea sin cruzar, puede ser necesario manejar el seguimiento de la línea de cruce como se muestra en la siguiente imagen. ¿Se puede usar el reconocimiento de etiquetas aquí para identificar y dejar que Maqueen Plus elija la ruta correcta en la intersección?



**Expansión del conocimiento:**

El estado de movimiento o los dominios de seguimiento de línea de Maqueen Plus se pueden dividir en 2 engranajes, 3 engranajes, 5 engranajes y 7 engranajes durante el seguimiento de línea. ¿Todavía es posible subdividir más? 9 engranajes? 11 engranajes?... hasta la subdivisión infinita. Cuanto más subdividido esté el dominio de regulación de velocidad, mejor será el efecto de seguimiento de línea. Pero los programas serán cada vez más largos. ¿Alguna solución?

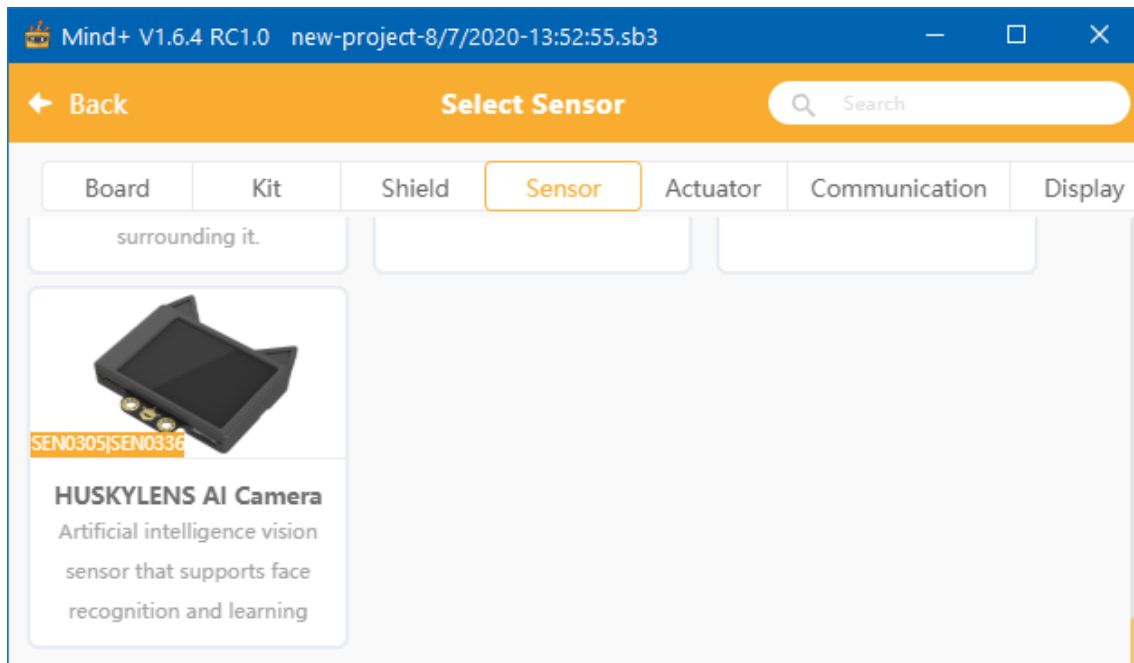
El algoritmo de regulación de velocidad PID puede ayudarnos a resolver este problema. PID significa proporción, control integral y diferencial. Es un sistema de control de circuito cerrado. El control de circuito cerrado es un método de control que corrige de acuerdo con la retroalimentación del objeto controlado. Se puede corregir de acuerdo con un determinado estándar de acuerdo con el error entre el valor real medido y el valor planificado.

Si está interesado en él, puede buscar información relevante en Internet y continuar optimizando nuestro efecto de seguimiento de línea.

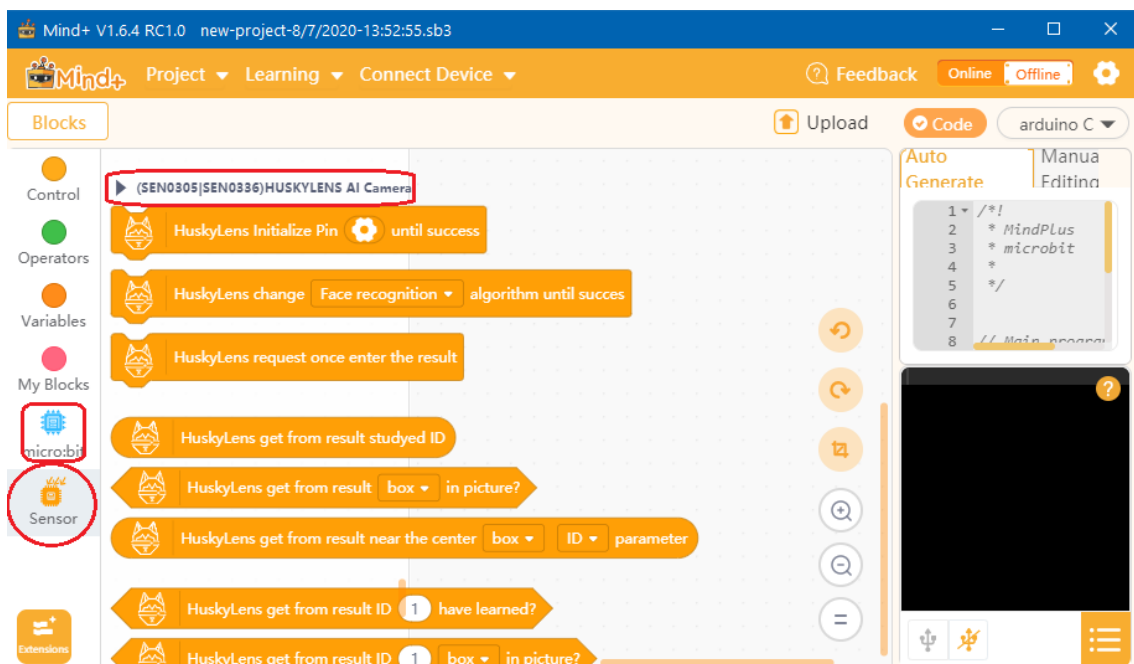
## ANEXO I

### Descripción de bloques de la librería HuskyLens

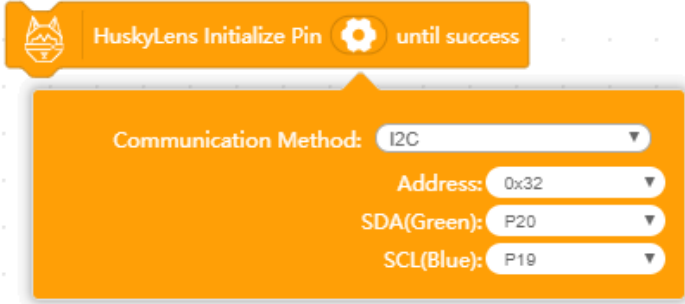
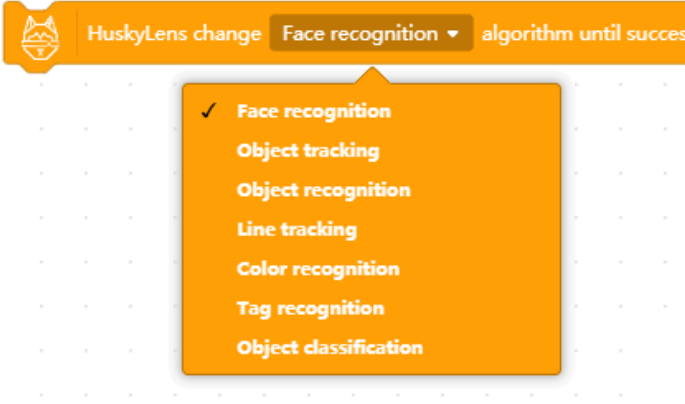



Para trabajar con nuestra cámara HuskyLens necesitamos conocer las funciones que se incorporan en la librería correspondiente dentro del entorno de programación Mind+.




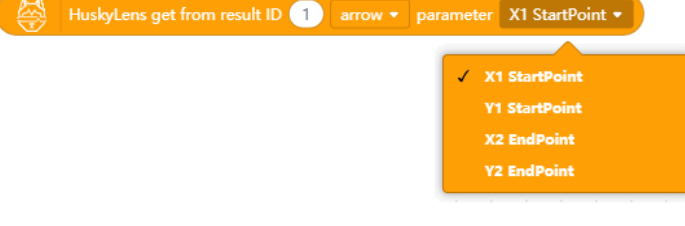

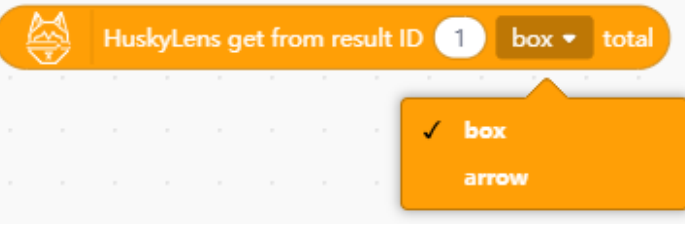
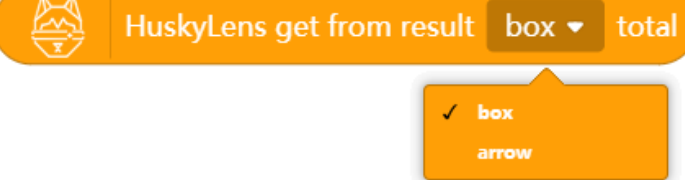
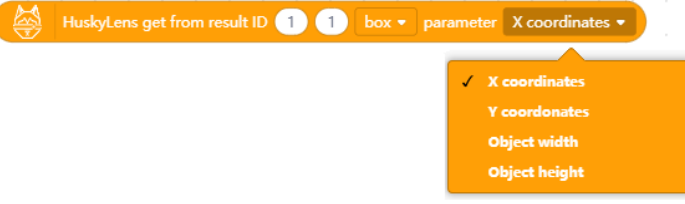
Una vez carga la librería aparecerá entre las que ya tenemos de Mind+ y la de Micro:bit que también debe estar.

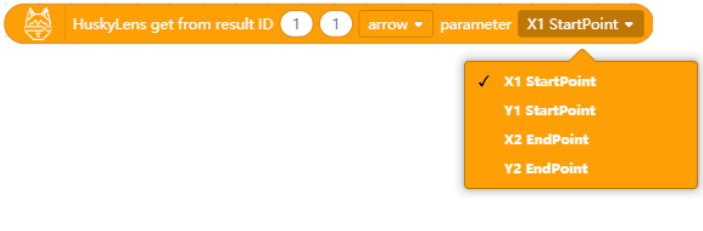
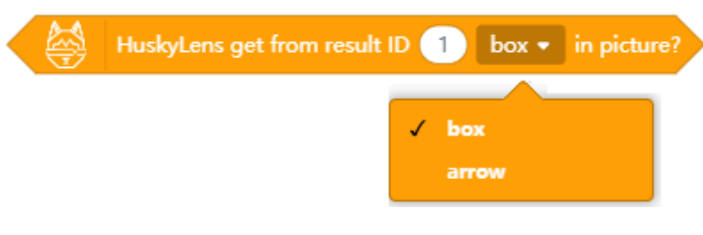
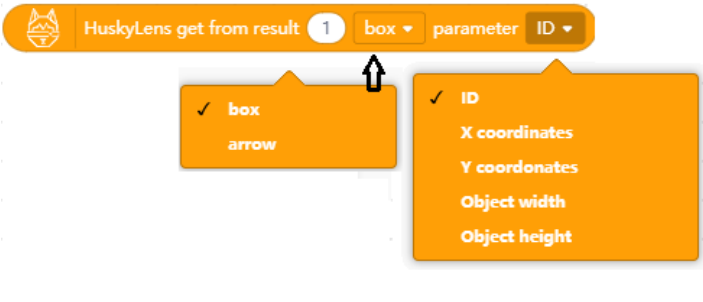
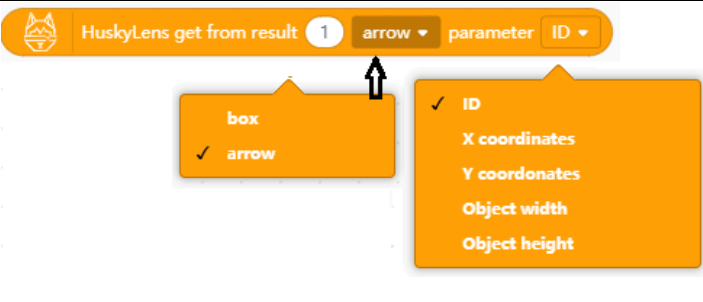
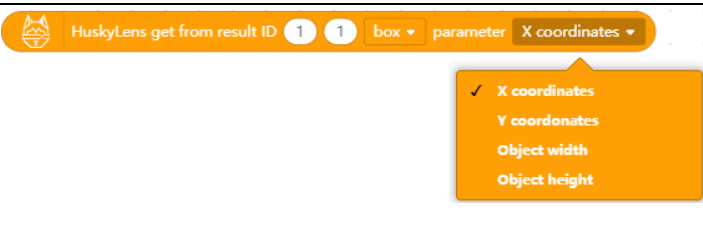
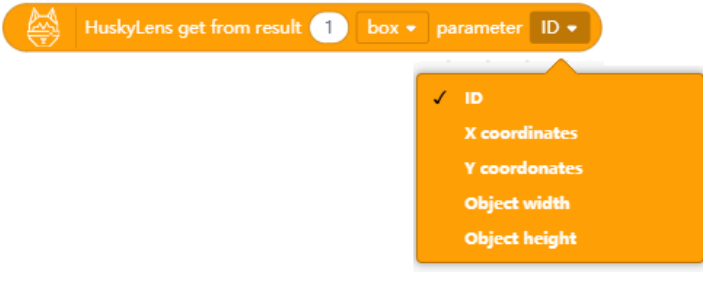


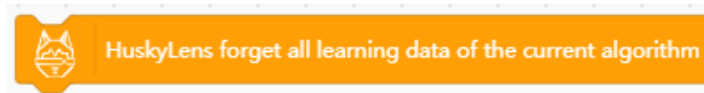
La forma de trabajo en este caso es en modo Offline, es decir programamos nuestro robot descargando la aplicación completa y funcionando después de manera autónoma.

Bloques de construcción	Explicación
	<p>La inicialización solo debe realizarse una vez, entre el inicio del programa principal y la ejecución del bucle, se puede seleccionar I2C o puerto serie, y no es necesario cambiar la dirección I2C. <b>Tenga en cuenta que Husky Lens necesita ajustar el "Protocolo de salida" en la configuración para que sea coherente con el programa; de lo contrario, los datos no se pueden leer.</b></p>
	<p>Algoritmo de cambio, puede cambiar a otros algoritmos en cualquier momento, solo puede haber un algoritmo al mismo tiempo, tenga en cuenta que el algoritmo de cambio lleva algo de tiempo.</p>
	<p>La placa de control principal solicita a HuskyLens que almacene los datos en el "resultado" (almacenado en la variable de memoria de la placa de control principal, y solicite actualizar los datos almacenados en la memoria una vez), y luego los datos se pueden obtener del "resultado". Después de llamar a este módulo, el "resultado" "Para obtener los últimos datos.</p>
	<p>Obtenga si el IDx se ha aprendido del "resultado" obtenido de la solicitud</p>
	<p>Obtenga si el IDx está en la pantalla del "resultado" obtenido por la solicitud. El cuadro se refiere al algoritmo cuyo objetivo es el cuadro en la pantalla. La flecha corresponde al algoritmo cuyo objetivo es la flecha en la pantalla. La flecha está actualmente seleccionada para el algoritmo de seguimiento de línea.</p>



Bloques de construcción	Explicación
	Otros algoritmos Seleccionar cuadros.
	<p>Obtenga los parámetros IDx del "resultado" obtenido de la solicitud, si la ID no está en la pantalla o no hay aprendizaje, devolverá -1</p>
	<p>Obtenga los parámetros IDx del "resultado" obtenido de la solicitud. Si esta ID no está en la pantalla o no se aprende, devolverá -1. Este módulo corresponde al resultado de salida del algoritmo de "patrulla de línea"</p>
	<p>Obtenga cuántos objetivos se han aprendido con el algoritmo actual a partir de los "resultados" obtenidos de la solicitud. Tenga en cuenta que HuskyLens puede establecer si aprender múltiples objetivos después de presionar durante mucho tiempo la tecla de selección para abrir la configuración avanzada.</p>
	<p>Obtenga el número de todos los "cuadros" o "flechas" en el campo de visión actual bajo el algoritmo actual de los "resultados" obtenidos por la solicitud, incluidos los objetivos que no se han aprendido o se han aprendido.</p>
	<p>Obtenga el número de objetivos con la misma ID bajo el algoritmo actual del "resultado" obtenido de la solicitud. Por ejemplo, si dos fotos de caras idénticas están en el campo de visión, el número es 2.</p>
	<p>Obtenga los parámetros del cuadro enésimo del IDx bajo el algoritmo actual del "resultado" obtenido por la solicitud. El mismo ID puede tener múltiples objetivos (por ejemplo, dos fotos de la misma persona aparecen en el campo de visión al mismo</p>

Bloques de construcción	Explicación
	<p>tiempo), preste atención al mismo ID El orden objetivo es aleatorio.</p>
	<p>Este bloque tiene el mismo efecto que el anterior, la diferencia es que este bloque lee los datos de la flecha.</p>
	<p>Obtenga si hay un cuadro o una flecha en la interfaz actual del "resultado" obtenido por la solicitud, incluido el aprendizaje (el ID es mayor que 0) y el no aprendizaje, si hay uno o más, devuelve 1.</p>
	<p>Obtenga la información del cuadro cerca del centro de la interfaz actual del "resultado" obtenido por la solicitud. El ID del cuadro no aprendizaje es 0, y si no, devuelve -1.</p>
	<p>Este bloque tiene el mismo efecto que el anterior, la diferencia es que este bloque lee los datos de la flecha.</p>
	<p>Obtenga la información del cuadro Nth (orden aleatorio) en la interfaz actual del "resultado" obtenido de la solicitud, incluido el aprendizaje (id mayor que 0) y no aprendizaje.</p>
	<p>Este bloque tiene el mismo efecto que el anterior, la diferencia es que este bloque lee los datos de la flecha.</p>

**Mind + V1.6.4 nuevos bloques de construcción:****Bloques de construcción****Explicación**

Esta activación de bloque es equivalente a presionar el **botón de aprendizaje** una vez, y el **aprendizaje de Erha** puede ser controlado por el programa. Puede ejecutar este módulo varias veces si necesita aprender varias veces.

Esta activación del bloque de construcción es equivalente a activar la función de **olvido**, y todo el contenido aprendido por el algoritmo actual se borra.

Este bloque se puede configurar con cualquier ID correspondiente al nombre que se muestra, y los nombres de diferentes algoritmos se pueden configurar por separado.

Este bloque puede mostrar caracteres en cualquier lugar de la pantalla. Puede mostrar letras, números y símbolos en inglés, pero no en chino.

Este bloque puede activarse para borrar todos los caracteres que se muestran en el bloque de **visualización de superposición de pantalla**.

Este ladrillo puede activar una foto o captura de pantalla para guardarla en la tarjeta TF en HuskyLens. Tenga en cuenta que debe insertar una tarjeta TF en HuskyLens. Las **capturas de pantalla** contienen los cuadros de texto y otro contenido que se muestra en la pantalla. Las **imágenes tomadas** solo contienen las fotos en la cámara, y no hay información de texto del cuadro.

**Agradecimientos**

Deseo manifestar mi agradecimiento a **DFRobot** por la facilidad en el acceso a la documentación técnica y el uso de las imágenes.

**Prof. José Manuel Ruiz Gutiérrez**

Agosto 2020