

# [Spanish ver.] HuskyLens Cámara Inteligencia Artificial



1204



1



0

Easy



AshG

Feb 19, 2021



## Guía de Usuario con Ejemplos de Utilización

Prof. José Manuel Ruiz Gutiérrez

Julio 2020

### Cámara Inteligencia Artificial

**HuskyLens** es una cámara basada en inteligencia artificial muy sencilla de utilizar. Tiene una pantalla integrada y un sistema de menús para configurarla con lo que se puede empezar a utilizar sin necesidad de programar. Tiene diferentes modos de funcionamiento:

- reconocimiento de caras
- reconocimiento de objetos
- seguimiento de líneas
- seguimiento de objetos
- reconocimiento de colores
- identificación de etiquetas basadas en códigos QR.

Utiliza un procesador de IA Kendryte K210 y dispone de conexiones I2C y UART para interactuar con Arduino, micro:bit o Raspberry Pi sin tener que entrar en la parte complicada de programar los algoritmos de inteligencia artificial.



**Processor**  
Kendryte K210, 400MHz, 64-Bit Dual Core RISC-V

**Image Sensor**  
Standard Version: OV2640, 2.0 Megapixel Camera  
Enhanced Version: OV5640, 5.0 Megapixel Camera

**Supply Voltage**  
4-Pin Connector: 3.3-5.0V, USB Connector: 5.0V

**Current Consumption<sup>(1)</sup>**  
310mA@3.3V, 220mA@5.0V

**Connection Interface**  
UART, 9600-115200bps; I2C, 100K-400K

**Display**  
2.0 inch IPS screen, 320\*240 resolution

**Dimensions**  
52mm \* 44.5mm

**Built-in Algorithms**  
Object Tracking, Face Recognition, Object Recognition, Line Tracking, Color Recognition, Tag Recognition

(1) 80% backlight brightness; Fill light off; Face recognition mode

### 1. Introducción

**HuskyLens** es un sensor de visión artificial AI fácil de usar con 6 funciones integradas: reconocimiento facial, seguimiento de objetos, reconocimiento de objetos, seguimiento de línea, detección de color y detección de etiquetas.

A través del puerto **UART/I2C**, HuskyLens puede conectarse a **Arduino**, **Raspberry Pi** o **micro:bit** para ayudarlo a hacer proyectos muy creativos sin jugar con algoritmos complejos.

### 2. Especificaciones

- Procesador: Kendryte K210
- Sensor Imagen:



AshG

Joined 1970-01-01

+ Follow



M-point

3650




Makelogs

50

#### Related Makelogs

Do plants need light?



Do plants need light? | DFRobot Science Lab EP07

#DFRobot #Science

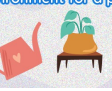
#DFScienceLab

AshG

Easy

0 0 4

What's the best environment for a plant?



What's the best environment for a plant? | DFRobot Science Lab EP09

#DFRobot #Science


#DFScienceLab

AshG

Easy

0 0 2

Project 1: Numbered Musical Notation of Color




#micro:Maqueen #AI

jie.tang

Easy

0 0 0

Project 2: Easy ETC (Electronic Toll Collection) System




#micro:Maqueen #AI

jie.tang

Easy

0 0 0

[Boson for micro:bit] Mind+ and micro: bit 01



#micro:bit #Mind+ #Science

alla

Easy

1 1 1

- SEN0305 HuskyLens: OV2640 (2.0MegaPixel Camera)
- SEN0336 HuskyLens PRO: OV5640 (5.0MegaPixel Camera)
- Tensión alimentación: 3.3~5.0V
- Consumo de corriente (TYP): 320mA@3.3V, 230mA@5.0V (modo de reconocimiento facial; 80% backlight brightness; llenar la luz apagada)
- Interfaz de conexión: UART; I2C
- Display: 2.0-inch IPS resolución de pantalla 320\*240
- Algoritmos integrados: reconocimiento de rostros, seguimiento de objetos, reconocimiento de objetos, seguimiento de líneas, reconocimiento de colores, reconocimiento de etiquetas
- Dimensiones: 52mm44.5mm / 2.051.75"

### 3. Vista de la tarjeta



#### 3.1 Conectores

- Conector USB: fuente de alimentación para Huskylens; conectarse a la computadora para actualizar el firmware
- Conector de 4 pines en Modo UART

Num	Etiqueta	Función	Pin	Descripción
1	T	TX		TX pin de HuskyLens
2	R	RX		RX pin de HuskyLens
3	-	GND		Polo negativo de alimentación(0V)
4	+	VCC		Polo positivo de alimentación (3.3~5.0V)

- Conector de 4 pines en Modo I2C

Num	Etiqueta	Pin	Función	Descripción
1	T		SDA	Reloj Línea Serie
2	R		SCL	Datos línea serie
3	-		GND	Negativo (0V)
4	+		VCC	Positivo (3.3~5.0V)

#### 3.2 Botones

Hay dos botones en HuskyLens, el botón de función y el botón de aprendizaje. Las operaciones básicas de estos dos botones se muestran a continuación:

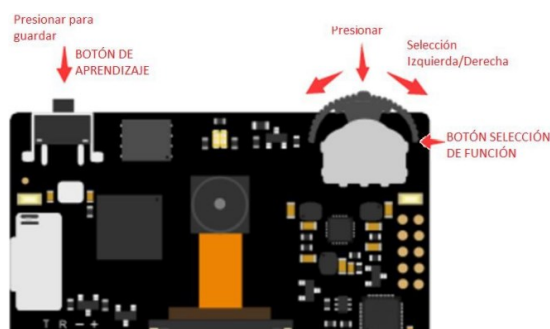
- Mueva el "botón de función" a izquierda o derecha para cambiar las diferentes funciones.

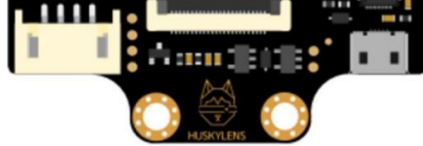


- Presione brevemente el "botón de aprendizaje" para aprender el objeto especificado; mantenga presionado el "botón de aprendizaje" para aprender continuamente el objeto especificado desde diferentes ángulos y distancias; Si HuskyLens ha aprendido el objeto antes, presione brevemente el botón "Aprender" para que se olvide.



- Mantenga presionado el "botón de función" para ingresar al menú de segundo nivel (configuración de parámetros) en la función actual. Marque a la izquierda, a la derecha o presione brevemente el "botón de función" para configurar los parámetros relacionados.





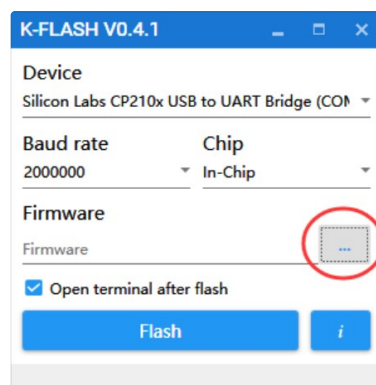
#### 4. Recargar el Firmware

Antes de usar este producto, se recomienda actualizar el firmware de HuskyLens a la última versión, especialmente para los patrocinadores de Kickstarter. Recomendamos cargar el firmware en Windows utilizando el software K-Flash, ya que cuenta con una GUI y es fácil de usar.

##### 4.1 En Windows

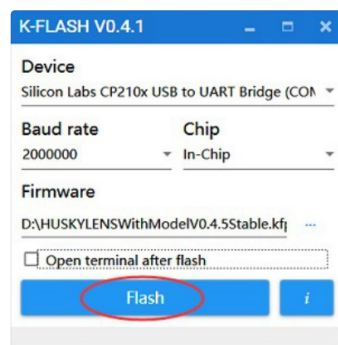
Se recomienda ejecutar el software K-Flash en Windows 10. Los pasos se muestran a continuación.

1. Descargue el software K-Flash. [Click aquí](#) para descargarlo Si K-Flash no puede ejecutarse, instale .NET Framework 4.7.1 primero. [Click aquí](#) para descargarlo
2. Descargue el controlador USB a UART e instálelo. [Click aquí](#) para descargarlo HuskyLens adopta el chip CP2102N para implementar la función USB a puerto serie.
3. Descargue el firmware más reciente. [Click aquí](#) para verificar los firmwares de todas las versiones. En este tutorial, adoptamos este firmware:  
**HUSKYLENSWithModelV0.4.6Stable.kfpgk.**
4. Abra el software K-Flash, luego haga clic en el botón (...) para cargar el nuevo firmware.

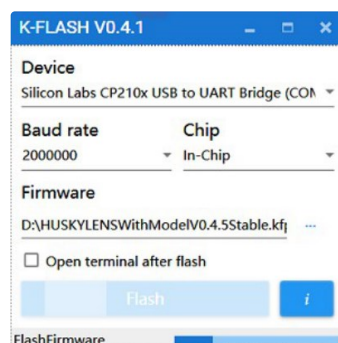


5. Configure el software K-Flash de acuerdo con los parámetros que se muestran a continuación:

- **Dispositivo:** seleccione el puerto COM en su computadora
- **Velocidad en baudios:** 2000000
- **Chip:** en chip
- **Desmarque** "Abrir terminal después de flash"



6. Haga clic en el botón Flash. Espere unos 5 minutos para completar la carga. Hay una actualización importante, por lo que puede llevar un poco de tiempo.



7. Aparece un cuadro y muestra "exitoso". La actualización se ha completado ahora. Disfrútala.

#### 4.2 En Linux o Mac

En esta sección, tomamos ubuntu 18.04.4 como ejemplo para mostrarle cómo actualizar el firmware de HuskyLens en Linux o Mac. Estos pasos se muestran a continuación::

1. Descargue el controlador USB a UART e instálelo. [Click aqui](#) para descargarlo.

HuskyLens adopta el chip CP2102N para implementar la función USB a puerto serie. En Ubuntu 18.04.4, el puerto serie USB de HuskyLens puede identificarse directamente cuando está enchufado, por lo que no es necesario instalar el controlador.

2. Descargue el último firmware y el script kflash.py. [Click aqui](#) para comprobarlos En este tutorial, adoptamos este firmware:

**HUSKYLENSWithModelV0.4.6Stable.kfpgk.**

3. Puede clonar todo el repositorio de "HuskyLens / HUSKYLENSUploader" en su computadora mediante el comando git.

4. Instale pip3 primero si no lo tiene en su sistema operativo.

sudo apt install python3-pip

```
user@ubuntu:~$ sudo apt install python3-pip
[sudo] password for user:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  build-essential dh-python dpkg-dev fakeroot g++ g++-7 gcc gcc-7
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libasan4 libatomic1 libc-dev-bin libc6-dev libcilkrts5 libexpat1-dev
  libfakeroot libgcc-7-dev libitm1 liblsan0 libmpx2 libpython3-dev
  libpython3.6-dev libquadmath0 libstdc++-7-dev libtsan0 libubsan0
  linux-libc-dev make nanaspaces-dev python-pip-whl python3-dev
  python3-distutils python3-lib2to3 python3-setuptools python3-wheel
  python3.6-dev
Suggested packages:
  debconf debconf-doc g++-multilib g++-7-multilib gcc-7-doc libstdc++6-7-dbg
  gcc-multilib autoconf automake1.16 libtool flex bison gcc-doc gcc-7-multilib
  gcc-7-locales libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg
```

Instalar pip3 on MAC

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
brew install python3
```

5. Ejecute el siguiente script para instalar pyserial.

sudo pip3 install pyserial

```
user@ubuntu:~$ sudo pip3 install pyserial
The directory '/home/user/.cache/pip/http' or its parent directory is not owned by
the current user and the cache has been disabled. Please check the permission
s and owner of that directory. If executing pip with sudo, you may want sudo's -H
flag.
The directory '/home/user/.cache/pip/' or its parent directory is not owned by th
e current user and caching wheels has been disabled. check the permissions and o
wner of that directory. If executing pip with sudo, you may want sudo's -H flag.
Collecting pyserial
  Downloading https://files.pythonhosted.org/packages/0d/e4/2a744dd9e3be04a0c090
7414e2a01a7c88b3915cbe3c8cc06e209f59c30/pyserial-3.4-py2.py3-none-any.whl (193k
B)
    100% |#####| 194kB 1.1MB/s
Installing collected packages: pyserial
Successfully installed pyserial-3.4
```

6. Vaya a la carpeta HUSKYLENSUploader.

cd HUSKYLENSUploader

```
user@ubuntu:~/Downloads$ cd HUSKYLENSUploader-master
```

7. Ejecute el siguiente script para cargar el firmware.

sudo python3 kflash.py -b 2000000 HUSKYLENSWithModelV0.4.6Stable.kfpgk

```
user@ubuntu:~/Downloads/HUSKYLENSUploader-master$ sudo python3 kflash.py -b 2000
000 HUSKYLENSWithModelV0.4.6Stable.kfpgk
[sudo] password for user:
[INFO] COM Port Auto Detected, Selected /dev/ttyUSB0
[39][INFO] Default baudrate is 115200 , later it may be changed to the value you
set.
[39][INFO] Trying to Enter the ISP Mode...
[39]
[INFO] Greeting Message Detected, Start Downloading ISP
Downloading ISP: |#####| 2.9% Comple
Downloading ISP: |#####| 5.9% Compl
Downloading ISP: |#####| 8.8% Compl
Downloading ISP: |#####| 11.8% Comp
Downloading ISP: |#####| 14.7% Comp
Downloading ISP: |#####| 17.6% Comp
Downloading ISP: |#####| 20.6% Comp
Downloading ISP: |#####| 23.5% Comp
Downloading ISP: |#####| 26.5% Comp
Downloading ISP: |#####| 29.4% Comp
Downloading ISP: |#####| 32.4% Comp
Downloading ISP: |#####| 35.3% Comp
Downloading ISP: |#####| 38.2% Comp
```

8. Espere unos 5 minutos para completar la carga..

```
[INFO] Writing clearFlash.bin into 0x00d90000
Downloading: |#####| 100.0% Complete
[INFO] Writing clearFlash.bin into 0x00da0000
Downloading: |#####| 100.0% Complete
[INFO] Writing clearFlash.bin into 0x00db0000
Downloading: |#####| 100.0% Complete
[INFO] Writing detect.kmodel into 0x00e00000
Downloading: |#####| 100.0% Complet
e
[INFO] Writing key point.kmodel into 0x00e5f000
Downloading: |#####| 100.0% Complet
e
[INFO] Writing Feature.kmodel into 0x00e80000
Downloading: |#####| 100.0% Complet
e
[INFO] Writing mobilenetv1_1.0.kmodel into 0x001bb000
Downloading: |#####| 100.0% Complet
```

```
[INFO] Writing object detect.bin into 0x009d4000
Downloading: [ ] 100.0% Complet
e
[INFO] Rebooting...
```

9. La actualización se ha completado ahora. Disfrútala.

## 5. Configuración general

### 5.1 Operación Básica

Hay 10 configuraciones de parámetros diferentes en la configuración general. La operación básica se muestra de la siguiente manera:

1. Seleccione la configuración general: marque el botón de función hacia la derecha hasta que se muestren las palabras **"General Settings Configuración general"** en la parte superior de la pantalla, luego selecciónelo.

2. Ingrese al modo de configuración general: presione brevemente el botón de función para ingresarlo.



3. Marque el botón de función hacia la izquierda o hacia la derecha para seleccionar un parámetro diferente, luego presione brevemente el botón de función para configurar el parámetro. Marque el botón de función hacia la izquierda o hacia la derecha para ajustar el parámetro. Luego presione brevemente el botón de función nuevamente para confirmar el parámetro.

4. Guarde la configuración: después de ajustar los parámetros, marque el botón de función hacia la izquierda para seleccionar **"Save&Return Guardar y volver"**, luego presione brevemente el botón de función. Un mensaje **"Do you save data?"**

¿Guarda datos? aparecerá. La selección predeterminada es **"Yes Sí"**. En este momento, presione brevemente el botón de función para guardar y salir.



### 5.1 Introduccion de Parámetros

#### • Procol type *Tipo de Protocolo*

Huskylens admite tres velocidades de transmisión **UART** (9600, 115200, 1000000) y el protocolo **I2C**. Además, admite la detección automática de los protocolos, es decir, Huskylens cambiará automáticamente entre UART e I2C. Recomendamos utilizar el protocolo de detección automática, que es simple y conveniente. El valor predeterminado es autodetección.

#### • Brillo de pantalla *Screen Brightness*

La pantalla admite el brillo de 1 ~ 100. El valor predeterminado es 80.

#### • Menu Auto-hide *Ocultar Menú automáticamente*

Cuando no opere los Huskylens por un período de tiempo, el menú en la pantalla se ocultará automáticamente. Este tiempo de duración se puede ajustar de 1 a 100 segundos. El valor predeterminado es 10 segundos.

#### • LED Light *Luz LED*

Hay dos luces LED en la parte frontal de los Huskylens. Puede activarlo o desactivarlo. El valor predeterminado es OFF.

#### • LED Brightness *Brillo LED*

El brillo de estas dos luces LED varía de 1 a 100. El valor predeterminado es 50.

#### • RGB Light *Luz RGB*

También hay una luz RGB en el frente de los Huskylens. Puede activarlo o desactivarlo. El valor predeterminado es ON.

#### • RGB Brightness *Brillo RGB*

El rango de brillo de esta luz RGB es de 1 a 100. El valor predeterminado es 20.

#### • Factory Reset *Restaurar de fábrica*

Huskylens se puede restablecer a la configuración de fábrica a través de esta función.

#### • Version

La versión actual del firmware incorporado.

#### • Language Huskylens admite chino e inglés.

## 6. Definiciones que necesita saber

### 6.1 Instrucciones de color

En cada función, las definiciones de color del marco y el símbolo "+" en el centro de la pantalla son todas iguales, lo que le ayuda a conocer el estado actual de HuskyLens.



Color	Estado
De naranja a amarillo, luego de amarillo a naranja	Aún no he aprendido el objeto pero estoy listo para aprender
Amarillo	Aprendiendo el nuevo objeto
Azul	He aprendido el objeto y lo he reconocido.

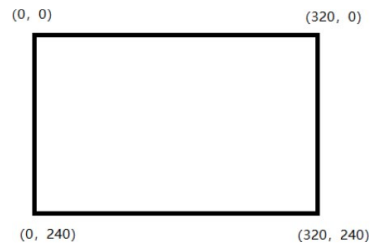
El indicador **LED RGB** actualmente solo se usa para indicar el estado de la función de reconocimiento facial. Sus colores se definen de la siguiente manera.

Color	Estado
Azul	Aún no he aprendido la cara, pero detecté la cara
Amarillo	Aprendiendo la nueva cara
Verde	He aprendido la cara y la he reconocido.

## 6.2 Sistema de coordenadas

Cuando HuskyLens detecta un objeto aprendido, el objetivo se seleccionará automáticamente mediante un marco de color en la pantalla. Las coordenadas de la posición del marco de color x e y se asignan de acuerdo con el siguiente sistema de coordenadas. Después de obtener las coordenadas del puerto UART / I2C, puede conocer la posición del objeto.

Formato:(x,y)



## 7. Introducción de funciones

### 7.1 Reconocimiento facial

Esta función puede detectar cualquier contorno de cara, reconocer y rastrear la cara aprendida.

#### 7.1.1. Aprende una cara

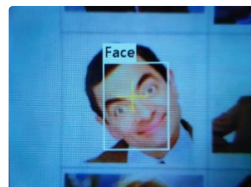
La configuración predeterminada es aprender y reconocer una sola cara.

##### Operación y configuración

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

##### Aprendizaje y Detección

**1. Detección de cara:** Apunte las lentes Husky a cualquier cara. Cuando se detecta una cara, se seleccionará automáticamente mediante un marco blanco con las palabras **"Face Cara"** en la pantalla.



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



Sin embargo, HuskyLens solo aprendió un plano (unidimensional) de la cara después de la operación anterior, mientras que la cara humana es tridimensional. Si se ha cambiado el ángulo de la cara, es posible que HuskyLens no lo reconozca. Por lo tanto, debe dejar que HuskyLens aprenda una cara desde sus diferentes ángulos. La operación se muestra de la siguiente manera: (Antes de que HuskyLens aprenda nuevas cosas, por favor, primero debe olvidar las cosas anteriores).

Mantenga presionado el **"botón de aprendizaje"**, apunte "+" de HuskyLens en diferentes ángulos de la cara. Durante este proceso, se mostrará un marco amarillo con las palabras **"Face: ID1"** en la cara, lo que indica que HuskyLens está aprendiendo la cara. Después de que HuskyLens aprendió la cara desde todos los ángulos, suelte el **"botón de aprendizaje"**. Luego, cuando HuskyLens detectó la cara aprendida, se mostrará un marco azul con las palabras **"Face: ID1"**, lo que indica que el proceso de aprendizaje de la cara se ha completado con éxito. Ahora HuskyLens puede reconocer la cara desde diferentes ángulos.



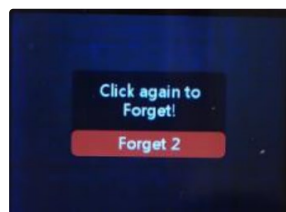


**Consejos:** si no hay un símbolo "+" en el centro de la pantalla antes de aprender, significa que HuskyLens ya ha aprendido la cara en la función actual, ahora HuskyLens lo está detectando. Si desea dejar que HuskyLens aprenda una nueva cara, primero debe hacer que olvide la cara aprendida.

**3. Reconocimiento facial:** La información de la cara aprendida se guardará automáticamente. Cuando HuskyLens detecta la cara aprendida de varias caras, esta cara se seleccionará con un marco azul y se identificará como **face: ID1**.

**4. Olvida la cara aprendida:** Si desea reconocer otra cara o volver a ingresar a la interfaz de información de la cara, debe eliminar la información de la cara actual. Cuando HuskyLens esté en el modo de reconocimiento facial, presione brevemente el **"botón de aprendizaje"**, la pantalla mostrará **"click again to forget 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, luego se mostrará el símbolo amarillo "+". Ahora puedes dejar que HuskyLens aprenda una nueva cara.

La operación de olvidar es totalmente la misma en las demás funciones. Por lo tanto, esta operación no se repetirá en capítulos posteriores.

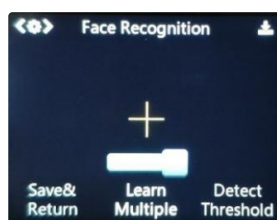


### 7.1.2. Aprende múltiples caras

La configuración predeterminada es aprender una sola cara. Para aprender varias caras, debemos habilitar el **"Learn Multiple Aprendizaje múltiple"** del reconocimiento de caras.

#### Operación y configuración

1. 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. Mantenga presionado el botón de función para ingresar la configuración de parámetros de la función de reconocimiento facial.
3. Marque el botón de función hasta que aparezca **"Learn Multiple 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 a 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 **"Save & Return Guardar y volver"**. Y la pantalla muestra **"Do you want to save the parameters? ¿Desea guardar los parámetros?"** Seleccione **"Yes Si"** de forma predeterminada, ahora presione brevemente el botón de función para guardar los parámetros y regresar automáticamente.

#### Aprendizaje y Detección

1. **Aprendizaje de caras múltiples:** Apunte el símbolo "+" a la cara, presione prolongadamente el **"Learn Buton botón de aprendizaje"** para conocer la cara de la primera persona. Luego suelte el **"botón de aprendizaje"**, se mostrará un marco azul con las palabras **"Face: ID1"** si HuskyLens detecta la misma cara, mientras tanto, se

mostrará el mensaje "Haga clic de nuevo para continuar. Haga clic en otro botón para finalizar". Presione brevemente el **"botón de aprendizaje"** antes de que finalice la cuenta regresiva si desea conocer la cara de otra persona. De lo contrario, presione brevemente el **"botón de función"** antes de que finalice la cuenta regresiva, o no presione ningún botón para dejar que finalice la cuenta regresiva.



En este capítulo, aprenderemos la siguiente cara continuamente. Por lo tanto, debemos presionar brevemente el **"botón de aprendizaje"** antes de que finalice la cuenta regresiva. Entonces podemos dejar que HuskyLens aprenda la cara de la segunda persona. Al igual que los pasos para reconocer la primera cara, apunte el símbolo "+" a la segunda cara, presione prolongadamente el **"botón de aprendizaje"** para conocer la cara de la segunda persona. Luego suelte el **"botón de aprendizaje"**, se mostrará un marco azul con las palabras **"Cara: ID2"** si HuskyLens detecta la misma cara.

**Consejos:** si no hay un símbolo "+" en el centro de la pantalla antes de aprender, significa que HuskyLens ya ha aprendido, ahora HuskyLens está detectando la cara. Si quieres dejar que HuskyLens

aprenda una nueva cara, primero debes dejar que HuskyLens olvide la cara aprendida.

Pase a la 7.1.1. Aprende una cara para ver cómo olvidar la cara aprendida.

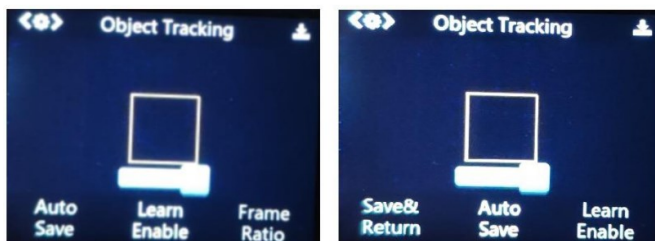
**2 Reconocimiento de rostros múltiples:** La información de la cara aprendida se guardará automáticamente. Cuando HuskyLens detecta la cara aprendida de varias caras, la cara aprendida se seleccionará con un marco y se identificará con la cara del mensaje: IDx. Por ejemplo, cuando HuskyLens detecta la cara aprendida de la primera persona, se seleccionará con un marco azul e identificará la cara: ID1; cuando HuskyLens detecta la cara aprendida de la segunda persona, se seleccionará con un marco amarillo e identificará la cara: ID2; y así.



## 7.2 Seguimiento de objetos

### Operación y configuración

1. 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. Mantenga presionado el botón de función para ingresar la configuración de parámetros de la función de seguimiento de objetos.
3. Marque el botón de función a la derecha para seleccionar "Habilitar aprendizaje", luego presione brevemente el botón de función y márkelo a la derecha para activar "Learn Enable Habilitar aprendizaje", es decir, el icono cuadrado en la barra de progreso está activado a la derecha. Luego presione brevemente el botón de función para confirmar este parámetro.
4. El método para activar el cambio de guardar modelos automáticamente es el mismo que antes. De acuerdo con los pasos anteriores para activar "Guardar automáticamente".



5. También puede ajustar el tamaño del cuadro configurando "Relación de cuadro" y "Tamaño de cuadro" para que coincida con la forma del objeto.
6. Marque el botón de función hacia la izquierda para seleccionar "Save&Return Guardar y volver", y presione brevemente el botón de función para guardar los parámetros y regresar automáticamente.

### Aprendizaje y Detección

1. **Aprendizaje de objetos:** Apunte HuskyLens al objeto objetivo, ajustando la distancia y hasta que el objeto esté incluido en el marco amarillo del centro de la pantalla. 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 marco amarillo con las palabras "Aprendizaje: ID1". Luego suelte el "botón de aprendizaje" para completar el aprendizaje.



2. **Seguimiento de objetos:** Mueva el HuskyLens o el objetivo, el marco seguirá al objetivo automáticamente. Al rastrear el objeto, se mostrarán las palabras amarillas "Aprendizaje: ID1", lo que indica que HuskyLens está rastreando el objeto mientras aprende. Esta configuración mejora la capacidad de seguimiento de objetos. También puede presionar y mantener presionado el "botón de función" para ingresar a la configuración de parámetros del menú secundario, seleccionar "Aprender en" y desactivar este parámetro.



#### Tips:

- Solo se puede rastrear un objeto a la vez. Puede ser cualquier objeto con un contorno claro, incluso varios gestos.
- Si no hay un marco amarillo en el centro de la pantalla, significa que HuskyLens ya ha aprendido un objeto. Consulte el método de eliminación de caras en el reconocimiento de caras para eliminar el objeto aprendido.

## 7.3 Reconocimiento de objetos

HuskyLens puede reconocer 20 objetos incorporados. Son avión, bicicleta, pájaro, bote, botella, autobús, automóvil, gato, silla, vaca, comedor, perro, caballo, moto, persona, maceta, oveja, sofá,



tren, TV

### 7.3.1. Reconocer un solo objeto

La configuración predeterminada es reconocer un solo objeto.

#### Operación y configuración

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

#### Aprendizaje y Detección

**1. Detección de objetos:** Al detectar objetos, HuskyLens lo reconocerá automáticamente y el objeto se mostrará con el marco blanco con su nombre en la pantalla.

En la actualidad, solo se pueden reconocer 20 objetos integrados, y los objetos restantes no se pueden reconocer temporalmente.



**2. Marcador de objeto:** Apunte el símbolo "+" al objeto, luego presione brevemente el "botón de aprendizaje". Al presionar, el color del marco cambia de blanco a azul, y el nombre del objeto y su número de identificación aparecerán en la pantalla.

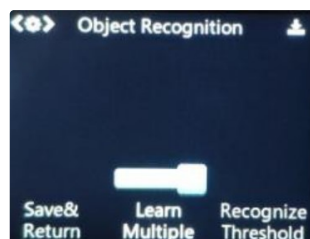
**Consejos:** Si no hay un símbolo amarillo "+" en el centro de la pantalla, significa que HuskyLens ya ha aprendido un objeto. Consulte el método de eliminación de caras en el reconocimiento de caras para eliminar el objeto aprendido.

**3.Reconocimiento de objetos:** Al encontrar los objetos aprendidos, serán seleccionados por el marco azul, y se mostrarán el nombre y el número de identificación. Al encontrar nuevos, el marco de selección es blanco. Esto se puede usar como un filtro simple para descubrir lo que necesita de un grupo de objetos.



### 7.3.2. Reconocer múltiples objetos

La configuración predeterminada es reconocer un solo objeto, por lo que primero debemos habilitar el parámetro " **Learn Multiple** Aprender múltiples" de la función de reconocimiento de objetos. Consulte el capítulo de aprendizaje y reconocimiento de caras múltiples para configurar, este capítulo no se repetirá para presentarlo.



El número de identificación está relacionado con el orden de marcado de los objetos. Por ejemplo, si un perro se marca por primera vez y un gato se marca por segunda vez, cuando se reconoce al perro, las palabras " **dog: ID1** " se mostrarán en la pantalla; y cuando se reconoce al gato, las palabras " **cat ID2** " se mostrarán en la pantalla.



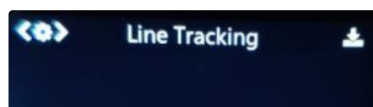
### 7.4 Seguimiento de línea

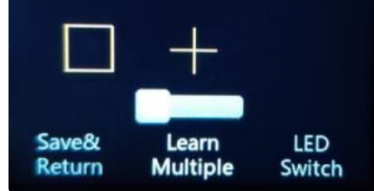
#### Operación y configuración

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 "Learn Multiple", luego presione brevemente el botón de función y gírelo hacia la izquierda para desactivar el interruptor " **Learn Multiple** ", es decir, el icono cuadrado en la barra de progreso se gira a la izquierda. Luego presione brevemente el botón de función para completar este parámetro.



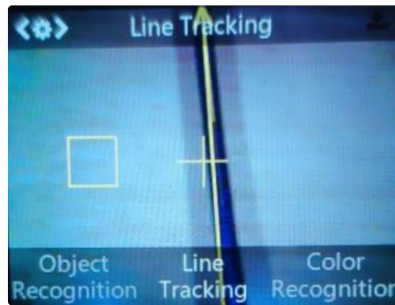


4. También puede encender los LED configurando "Interruptor LED". Esto es muy útil en el ambiente oscuro.

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

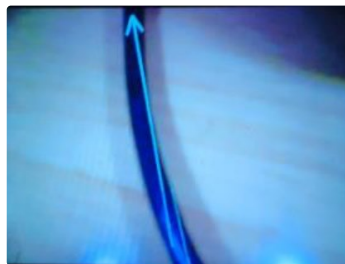
### Aprendizaje y Detección

**1. Aprendizaje línea:** Apunte el símbolo "+" a la línea, luego apunte el marco amarillo al área de fondo. Se recomienda que dentro del campo de visualización de HuskyLens, solo permanezcan líneas para aprender y no haya líneas cruzadas. Luego, presione brevemente el "botón de aprendizaje" para completar el proceso de aprendizaje. Aparecerá una flecha azul de dirección de ruta en la pantalla.



**Consejos:** Si no hay un marco amarillo y el símbolo "+" en el centro de la pantalla, significa que HuskyLens ya ha aprendido la línea. Consulte el método de eliminación de caras en el reconocimiento de caras para eliminar la línea aprendida.

**2. Predicción de línea:** Cuando HuskyLens detecta la línea que se ha aprendido, aparecerá automáticamente una flecha azul en la pantalla. La dirección de la flecha indica la dirección predicha de la línea..



### Consejos:

- Al aprender la línea, necesitamos ajustar la posición de HuskyLens para que sea paralela a la línea.
- HuskyLens puede aprender varias líneas de acuerdo con el color de las líneas, pero estas líneas deben ser líneas monocromas con un color obvio que difiera del fondo.
- En la mayoría de los casos, el color de la línea de seguimiento es solo uno. Por lo tanto, para garantizar la estabilidad, recomendamos seguir la línea única.
- El color de las líneas tiene mucho que ver con la luz ambiental. Cuando patrulla la línea, trate de mantener la luz ambiental lo más estable posible.

## 7.5 Reconocimiento de Color

### 7.5.1. Aprende un solo color

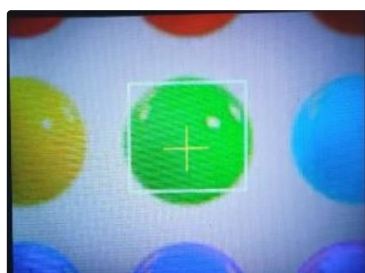
La configuración predeterminada es aprender un solo color.

#### Operación y configuración

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

### Aprendizaje y Detección

**1. Detección de color:** Apunte el icono "+" en el centro de la pantalla 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.

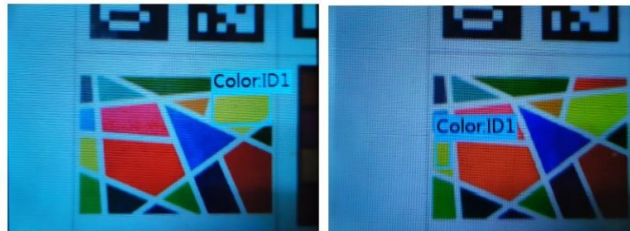


**2. Aprendizaje de color:** Apunte el símbolo "+" al bloque de color y mantenga presionado el "botón de aprendizaje". Aparecerá un marco amarillo en la pantalla, indicando que HuskyLens está aprendiendo el color. En este momento, ajuste la distancia y el ángulo entre HuskyLens y el bloque de color, para que el tamaño del marco amarillo sea igual al del bloque de color. Luego, suelte el "botón de aprendizaje" para completar el aprendizaje.



**Consejos:** Si no hay un símbolo "+" en el centro de la pantalla antes de aprender, significa que HuskyLens ya ha aprendido, ahora HuskyLens está detectando el color. Si desea dejar que HuskyLens aprenda un nuevo color, primero debe dejar que HuskyLens olvide el color aprendido.

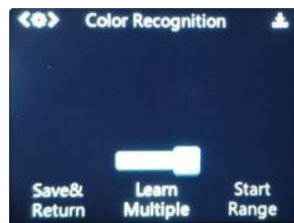
**3. Color Recognition:** Cuando encuentre bloques de color iguales o similares, se mostrará automáticamente un marco azul con una ID en la pantalla, y el tamaño del marco azul es igual al tamaño de los bloques de color. Cuando aparecen varios bloques de colores iguales o similares al mismo tiempo, no se pueden reconocer los otros bloques de color, es decir, solo se puede reconocer un bloque de color cada vez.



**Consejos:** 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 sin cambios.

#### 7.5.1. Aprende múltiples colores

La configuración predeterminada es reconocer un solo color, por lo que debemos habilitar el parámetro "Aprender múltiples" de la función de reconocimiento de color. Consulte el capítulo de aprendizaje y reconocimiento de caras múltiples para configurar, este capítulo no lo repetirá.



El número de identificación está relacionado con el orden del color aprendido. Por ejemplo, si se marca un bloque amarillo por primera vez y se marca un bloque verde por segunda vez, cuando se reconoce el bloque amarillo, las palabras "Color: ID1" se mostrarán en la pantalla y cuando el bloque verde es reconocido, las palabras "Color: ID2" se mostrarán en la pantalla.



### 7.6 Reconocimiento de etiquetas

#### 7.6.1. Aprende una sola etiqueta

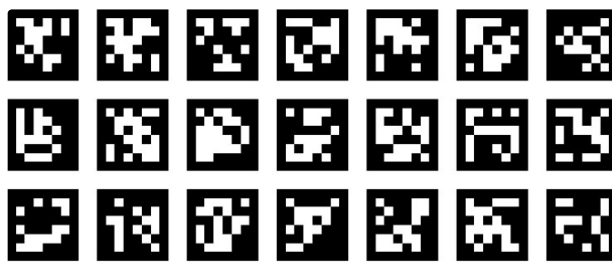
La configuración predeterminada es aprender una sola etiqueta.

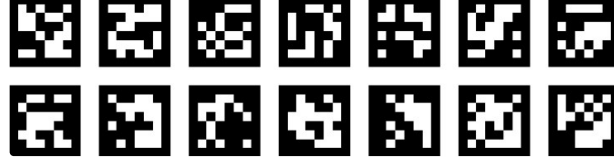
##### Operación y configuración

Marque el botón de función hacia la derecha o hacia la izquierda hasta que las palabras "Reconocimiento de etiqueta" se muestren en la parte superior de la pantalla.

##### Aprendizaje y Detección

Puede usar las siguientes imágenes de etiquetas / códigos QR para probar esta función.





**1. Detección de etiqueta:** Cuando HuskyLens detecta la etiqueta, la etiqueta será seleccionada automáticamente por el marco blanco en la pantalla.



**2. Aprendiendo Tags:** Apunte el símbolo "+" a la etiqueta y presione el "botón de aprendizaje". Se mostrará un marco amarillo con las palabras "Etiqueta: ID1" en la pantalla, lo que indica que HuskyLens está aprendiendo la etiqueta ahora. Luego, suelte el "botón de aprendizaje" para completar el proceso de aprendizaje.



**Consejos:** Si no hay un símbolo "+" en el centro de la pantalla antes de aprender, significa que HuskyLens ya ha aprendido, ahora HuskyLens está detectando la etiqueta. Si desea dejar que HuskyLens aprenda una nueva etiqueta, primero debe dejar que HuskyLens olvide la etiqueta aprendida.

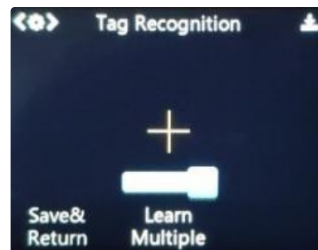
**4. Reconocimiento de etiquetas** Al encontrar la etiqueta aprendida, un marco azul con una ID se mostrará automáticamente en la pantalla.



#### 7.6.2. Aprende múltiples etiquetas

La configuración predeterminada es reconocer una sola etiqueta, por lo que debemos habilitar el parámetro "Aprender múltiples" de la función de reconocimiento de etiquetas.

Consulte el capítulo de aprendizaje y reconocimiento de caras múltiples para configurar, este capítulo no lo repetirá.



Del mismo modo, la ID está relacionada con el orden de las etiquetas aprendidas.



Education

Enter Keywords

HOME

COMMUNITY

FORUM

WIKI

BLOG

EDUCATION



LOGIN/SIGN UP



#### 7.7 Clasificación de Objetos (Beta)

Ir a este link para la demostración [LINK](#) Actualización de HuskyLens: objeto clasificado

En la nueva versión 0.4.9Class, se lanza una nueva función "Clasificación de objetos". Las capacidades de aprendizaje automático esperadas finalmente están llegando. ¿Qué podríamos hacer con esta nueva función? Vamos a ver.

¿Qué es la clasificación de objetos?

La función de clasificación de objetos de HuskyLens puede aprender de múltiples fotos de diferentes objetos mediante algoritmos de aprendizaje automático integrados. Después de completar el aprendizaje de clasificación de objetos, cuando HuskyLens detecta el objeto aprendido, puede reconocer y mostrar el número de ID del objeto. Bueno, cuanto más aprende, más preciso puede ser el reconocimiento.

#### ¿Qué podríamos hacer con la clasificación de objetos?

Ahora todos sabemos que HuskyLens admite 6 funciones básicas: reconocimiento facial, seguimiento de objetos, reconocimiento de objetos, seguimiento de líneas, reconocimiento de colores y reconocimiento de etiquetas.

Bueno, cuando llega un proyecto popular de reconocimiento de máscaras, que es reconocer si las personas usan máscaras. ¿Qué podría hacer HuskyLens?

El reconocimiento facial no puede distinguir las máscaras, el seguimiento de objetos no puede aprender máscaras múltiples, el reconocimiento de objetos no puede reconocerlas, sin mencionar el reconocimiento de color y el reconocimiento de etiquetas. Pero ahora, con la nueva función: clasificación de objetos, HuskyLens puede reconocer y distinguir la cara con una máscara. Pero como hacerlo?

Primero, aprende una cara sin máscara (clase1).

Luego, aprende la cara con una máscara simple (clase2).

Y también puede aprender la cara con una máscara KN95 (clase 3).

Después, HuskyLens puede reconocerlos. Encontrará que cuando HuskyLens muestra ID1 cuando reconoce el objeto sin máscara, muestra ID2 cuando reconoce el objeto con una máscara simple y muestra ID3 cuando reconoce el objeto con una máscara KN95. Además, lo mejor es que la clasificación de objetos no distingue rostros de diferentes personas, todos pueden hacerlo.

¡Practicémoslo!

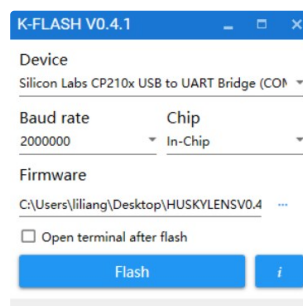
### 7.8. Ejemplo 1: Reconocimiento de máscarilla

#### Paso 1

##### Actualiza el firmware

Actualice el firmware a **HUSKYLENSWithModelV0.4.9Class.kfpkg** con K-Flash.

Consulte la [Section 4](#) de la wiki para obtener más detalles.

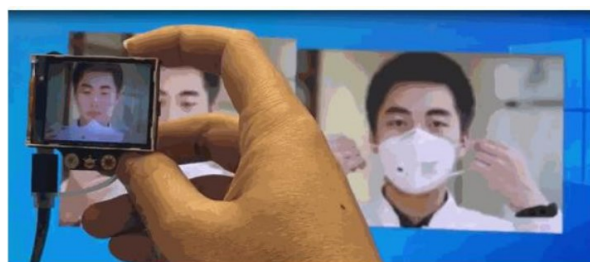


#### Paso 2

##### Entrena para aprender

Cambie a la Clasificación de objetos, presione el botón de aprendizaje para aprender imágenes sin máscara. Con una pulsación larga, HuskyLens puede aprender imágenes desde diferentes ángulos. Nuevamente, cuanto más aprende, más preciso puede ser el reconocimiento. Se recomiendan fotos de más de 30. Suelte el botón, luego se completa el proceso de aprendizaje.

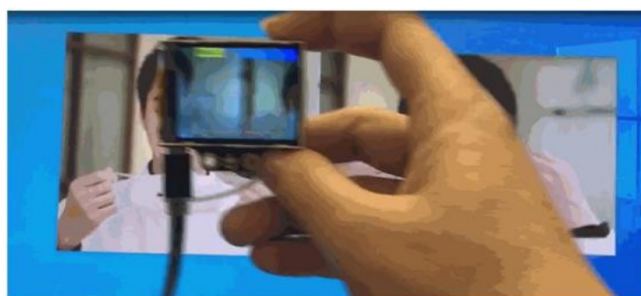
Luego, en operaciones similares, deje que HuskyLens aprenda imágenes con una máscara, también aprenda más de 30 imágenes y luego suelte el botón de aprendizaje.



#### Paso 3

##### Prueba de reconocimiento

Una vez realizado el aprendizaje, apunte HuskyLens a la imagen sin máscara y a la imagen con máscara, si la pantalla muestra ID1 e ID2 respectivamente, lo que significa que HuskyLens ya ha definido estos 2 tipos de objetos.



#### Paso 4

Codificación



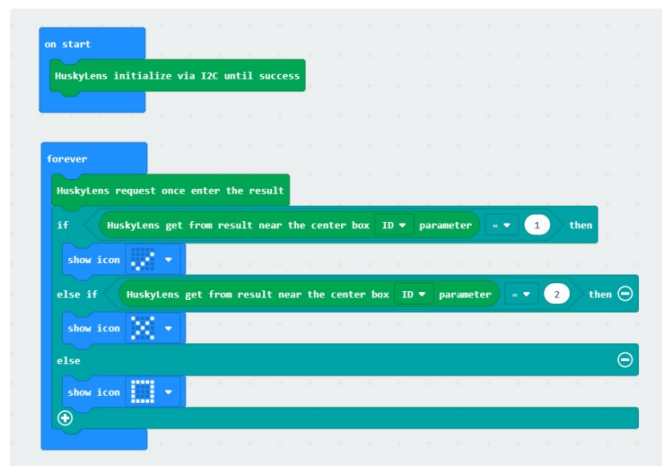
Una vez que HuskyLens puede reconocer objetos, podemos crear un proyecto con la placa base, como Arduino o micro: bit.

Aquí usamos micro: bit para realizar una alarma de máscara. Si HuskyLens detecta una máscara en una cara, el '✓' estará activado. De lo contrario, la 'x' estará activada.

La clasificación de objetos no tiene que cambiar algoritmos, puede leer la cantidad de cuadros aprendidos y las ID de los cuadros.

Consulte la [Setción 10](#) en la wiki para obtener más detalles sobre el cableado, la extensión de carga, etc.

El código de muestra basado en makecode para micro: bit se muestra a continuación.



El código de muestra para Arduino se muestra a continuación. La comunicación es I2C.

Consulte la [Setción 8](#) en la wiki para obtener más detalles sobre el cableado, la instalación de la biblioteca arduino, etc.

## CÓDIGO

### CODE

```
1 #include "HUSKYLENS.h" #include "SoftwareSerial.h"
2
3 HUSKYLENS huskylens;
4 //HUSKYLENS green line >> SDA; blue line >> SCL void printResult(HUSKYLENSResult result);
5
6 void setup() { Serial.begin(115200); Wire.begin();
7 while (!huskylens.begin(Wire))
8 {
9   Serial.println(F("Begin failed!"));
10  Serial.println(F("1.Please recheck the \"Protocol Type\" in HUSKYLENS (General Settings>>P
11  Serial.println(F("2.Please recheck the connection.)); delay(100);
12 }
13 }
14 }
```

## Paso5

### Prueba

Cuando se reconoce una máscara, se mostrará el '✓', que indica transitable. De lo contrario, sin una máscara, se mostrará la 'x', lo que significa intransitable.



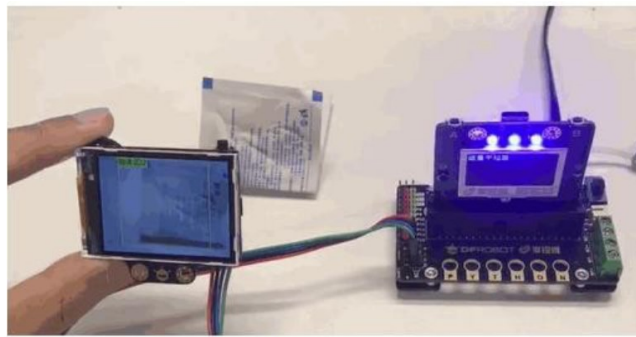
## 7.9. Ejemplo 2: Piedra Papel Tijera

El aprendizaje de gestos de máquina puede realizar el reconocimiento de gestos. Se puede ampliar, como reconocimiento digital de gestos, lenguaje de señas, reconocimiento de señas, etc.



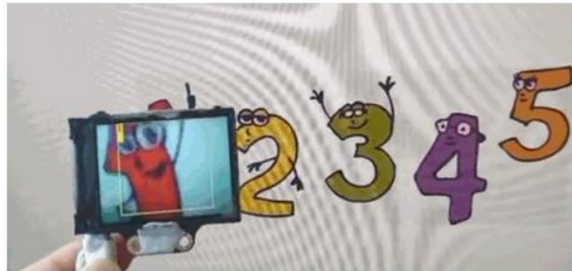
## 7.10. Ejemplo 3: Clasificación de basura

Función de clasificación de basura aparentemente muy avanzada, a través del aprendizaje automático es muy fácil de lograr. Y luego conecte el control de la placa de control principal, unos pocos timones pueden simular la función de clasificación de basura real. Dado que puede clasificar la basura, también se pueden realizar fácilmente funciones como la clasificación de monedas, la clasificación de frutas, la limpieza automática de la cantimplora y otras.



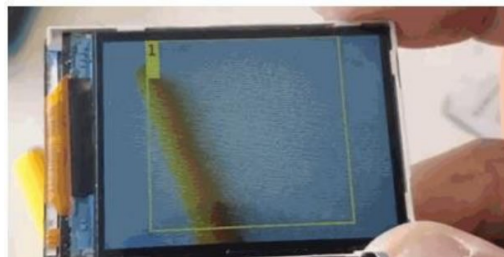
#### 7.11. Ejemplo 4: Reconocimiento de números

Ya sea que se trate de un número de dibujos animados, un número escrito a mano o un número mayúsculo a un número árabe, está bien jugar con HuskyLens.



#### 7.12. Ejemplo 5: Reconocimiento de posición

Aunque la clasificación de objetos no puede devolver los valores de coordenadas de los objetos, podemos aprender colocando objetos en diferentes ubicaciones, logrando indirectamente el reconocimiento de posición. Al utilizar esta función, encontrará que también se puede usar para hacer líneas de patrulla, líneas a la izquierda o derecha, incluso intersecciones, las uniones en T son reconocibles, así como la identificación de intersecciones, semáforos analógicos, etc. implementado directamente. Así es, la función sin conductor se asienta fácilmente.



#### Resumen

Todos los ejemplos anteriores son solo para comenzar nuevas exploraciones. Aplicando la función de clasificación de objetos, podemos implementar muchas de las funciones que solíamos hacer confiando en una cámara de computadora en esta pequeña cámara AI. Con el aprendizaje automático, la idea concebible se vuelve muy grande, aunque parece tener solo una función, pero puede superar las 6 funciones básicas.

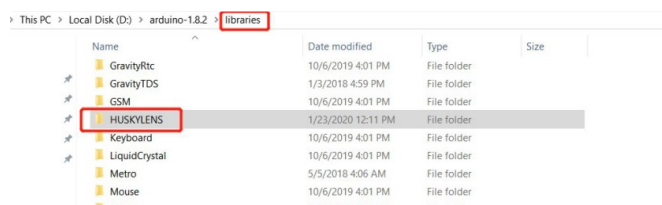
HuskyLens puede aprender, entonces, siempre y cuando quieras que se reconozca, solo échale un vistazo. ¡Ven a abrir tu mente o idea, únete a nosotros para entrenar tus HuskyLens, desbloquea más jugadas nuevas, deja que se vuelva más inteligente y poderoso!

### 8. Arduino Tutorial

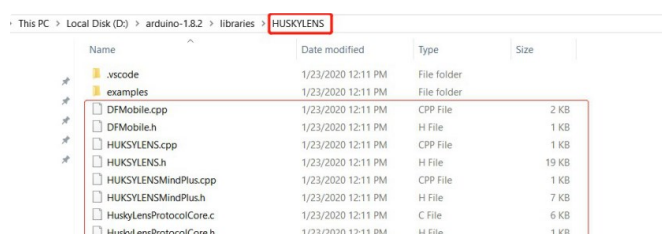
Descargue e instale primero la librería [HUSKYLENS Library](#).

#### 8.1 Instalar la librería

1. Descomprima el archivo, luego copie la carpeta a la carpeta "librerías" del IDE de Arduino. Luego verifique si el nombre de la carpeta es "HUSKYLENS". Si no, cámbielo como "HUSKYLENS". Tenga en cuenta que el nombre del archivo de la biblioteca debe ser HUSKYLENS.



2. Todos los archivos .h y .cpp deben estar en el directorio raíz de la carpeta "HUSKYLENS".



## 8.2 Introducción a la API

Haga clic en el enlace a continuación para ver los documentos de introducción de la API..

<https://github.com/HuskyLens/HUSKYLENSArduino/blob/master/HUSKYLENS%20Arduino%20API.md>

## 8.3 Proyecto 1: Leer datos de posición

En este proyecto, HuskyLens se conectará a la placa base Arduino. Y Arduino Uno leerá los datos de posición del objeto de HuskyLens. Luego, el monitor del puerto serie imprimirá los datos. Para que pueda leer la posición del objeto en tiempo real.

### Requerimientos

#### · Hardware

o DFRduino UNO R3 (o similar) x 1

o HUSKYLENS x 1

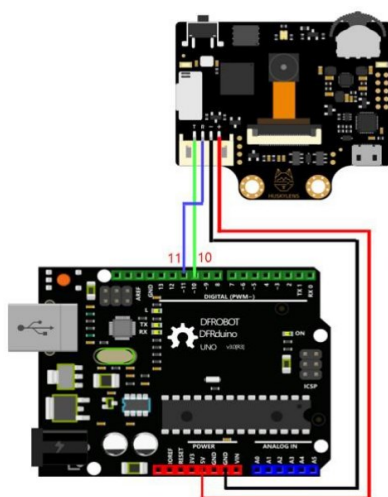
o M-M/F-M/F-F Conectores

#### · Software

o Arduino IDE(se recomienda la versión 1.8.x)

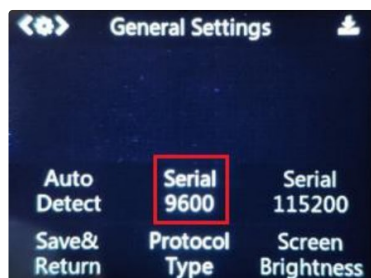
o Descargue e instale el [HUSKYLENS Library](#) (Acerca de cómo instalar la librería?)

### Modo UART (SoftwareSerial) Diagrama de conexión



### Configuración del protocolo HuskyLens

Debe configurar el tipo de protocolo de HuskyLens. El protocolo debe ser 'Serie 9600'. Por supuesto, puede adoptar el protocolo de detección automática, que es fácil de usar y conveniente.



### Código de muestra

#### CODE

```
1 #include "HUSKYLENS.h" #include "SoftwareSerial.h"
2
3 HUSKYLENS huskylens;
4 SoftwareSerial mySerial(10, 11); // RX, TX
5 //HUSKYLENS green line >> Pin 10; blue line >> Pin 11 void printResult(HUSKYLENSResult res
6
7 void setup() { Serial.begin(115200); mySerial.begin(9600);
8 while (!huskylens.begin(mySerial))
9 {
10 Serial.println(F("Begin failed!"));
11 Serial.println(F("1.Please recheck the \"Protocol Type\" in HUSKYLENS (General Settings)>>P
12 Serial.println(F("2.Please recheck the connection.)); delay(100);
13 }
14 }
```

### Operaciones y resultados esperados

1. Cargue los códigos anteriores en su placa Arduino.
2. Deje que su HuskyLens aprenda primero algo nuevo. Puede consultar los capítulos anteriores de este tutorial.
3. Abra el monitor en serie de Arduino IDE, luego obtendrá los datos de posición del objeto.

Si HuskyLens está en el modo de reconocimiento facial, seguimiento de objetos, reconocimiento de objetos, reconocimiento de color, reconocimiento de etiquetas, obtendrá los resultados de la siguiente manera:

```
Block:xCenter=162,yCenter=135,width=138,height=146
Block:xCenter=163,yCenter=134,width=138,height=146
Block:xCenter=163,yCenter=134,width=138,height=146
Block:xCenter=162,yCenter=137,width=108,height=145
Block:xCenter=162,yCenter=135,width=108,height=146
Block:xCenter=162,yCenter=135,width=108,height=146
Block:xCenter=163,yCenter=135,width=138,height=146
Block:xCenter=163,yCenter=136,width=109,height=146
Block:xCenter=163,yCenter=137,width=138,height=146
Block:xCenter=162,yCenter=136,width=138,height=146
Block:xCenter=162,yCenter=136,width=109,height=145
Block:xCenter=162,yCenter=137,width=109,height=145
Block:xCenter=163,yCenter=135,width=109,height=146
Block:xCenter=163,yCenter=137,width=109,height=145
```

width and height of the recognition frame

center coordinates of the recognition frame

Si HuskyLens está en el modo de seguimiento de línea, obtendrá los resultados de la siguiente manera:

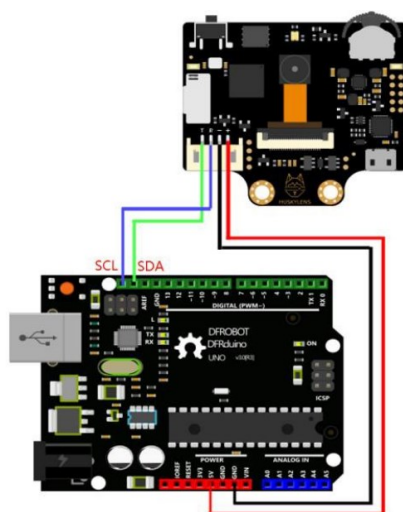
```
COM4
Arrow:xOrigin=304,yOrigin=158,xTarget=224,yTarget=82
Arrow:xOrigin=312,yOrigin=164,xTarget=224,yTarget=82
Arrow:xOrigin=312,yOrigin=164,xTarget=216,yTarget=86
Arrow:xOrigin=304,yOrigin=164,xTarget=216,yTarget=86
Arrow:xOrigin=304,yOrigin=166,xTarget=216,yTarget=86
Arrow:xOrigin=304,yOrigin=164,xTarget=208,yTarget=86
Arrow:xOrigin=304,yOrigin=164,xTarget=216,yTarget=84
Arrow:xOrigin=312,yOrigin=164,xTarget=216,yTarget=84
Arrow:xOrigin=312,yOrigin=164,xTarget=216,yTarget=84
Arrow:xOrigin=304,yOrigin=164,xTarget=216,yTarget=82
Arrow:xOrigin=312,yOrigin=164,xTarget=216,yTarget=84
Arrow:xOrigin=312,yOrigin=162,xTarget=216,yTarget=84
Arrow:xOrigin=312,yOrigin=164,xTarget=216,yTarget=82
Arrow:xOrigin=304,yOrigin=162,xTarget=216,yTarget=82
Arrow:xOrigin=312,yOrigin=162,xTarget=216,yTarget=80
Arrow:xOrigin=312,yOrigin=164,xTarget=216,yTarget=82
```

starting coordinates of the path arrow

end coordinates of the path arrow

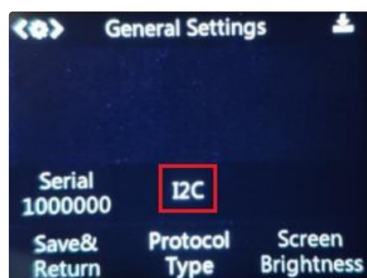
### Modo I2C

#### Diagrama de conexión



#### Configuración del protocolo HuskyLens

Debe configurar el tipo de protocolo de HuskyLens. El protocolo debe ser 'I2C'. Por supuesto, puede adoptar el protocolo de detección automática, que es fácil de usar y conveniente.



#### Código de muestra

#### CODE

```
#include "HUSKYLENS.h" #include "SoftwareSerial.h"
HUSKYLENS huskylens;
//HUSKYLENS green line >> SDA; blue line >> SCL void printResult(HUSKYLENSResult result);
void setup() { Serial.begin(115200); Wire.begin();
while (!huskylens.begin(Wire))
{
Serial.println(F("Begin failed!"));
Serial.println(F("1.Please recheck the \"Protocol Type\" in HUSKYLENS (General Settings>>"));
Serial.println(F("2.Please recheck the connection.")); delay(100);
}
}
```

#### Operaciones y resultados esperados

1. Sube los códigos anteriores a tu placa Arduino.
2. Deje que su HuskyLens aprenda primero algo nuevo. Puede consultar los capítulos anteriores de este tutorial.

3. Abra el monitor en serie de Arduino IDE, luego obtendrá los datos de posición del objeto, igual que los resultados en modo UART. Consulte el capítulo anterior, que no se repetirá aquí..

## 9. Raspberry PI Tutorial

En este capítulo, utilizaremos la Raspberry Pi para leer los datos de HuskyLens. El protocolo de comunicación es I2C..

### 9.1 Inicializar Raspberry Pi

En su Raspberry Pi, debe habilitar I2C en la configuración antes de usarlo. Por lo tanto, abra una terminal en su Raspberry Pi y ejecute los siguientes comandos.

1. Ejecute `sudo raspi-config`
2. Use la flecha hacia abajo para seleccionar 5 opciones de interfaz
3. Flecha hacia abajo a P5 I2C.
4. Seleccione sí cuando le pida que habilite I2C
5. También seleccione sí si le pregunta sobre cargar automáticamente el módulo del núcleo.
6. Use la flecha derecha para seleccionar el botón Finalizar.
7. Seleccione sí cuando solicite reiniciar.
8. Después de reiniciar, ejecute `sudo apt-get install -y i2c-tools`
9. Ejecute `sudo apt-get install python-smbus`
10. Ejecute `sudo pip3 install pyserial`

### 9.2 I2C Guía de cableado

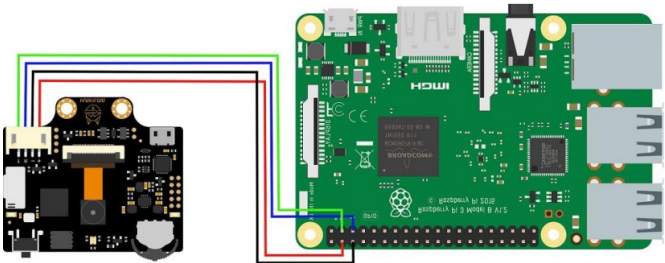
El protocolo principal para la comunicación entre HuskyLens y Raspberry Pi es I2C. Esto requiere que use el conector de 4 pines para cablear tierra, alimentación, SDA y SCL. Para leer más sobre cómo funciona I2C, consulte el siguiente enlace:

<https://en.wikipedia.org/wiki/I%C2%B2C>

Pin Outline (Raspberry Pi)  
Pin Outline (HUSKYLENS)

Etiqueta	Pin	Función	Descripción
T	SDA		línea de reloj serial
R	SCL		línea de datos en serie
-	GND		polo negativo de la fuente de alimentación (0V)
+	VCC		polo positivo de la fuente de alimentación (3.3 ~ 5.0V)

#### Diagrama de conexión



**Consejos:** HuskyLens consume mucha corriente, hasta 3.3V 320mA, 5V 230mA o más. Recomendamos conectar HuskyLens a los pines de la fuente de alimentación de 5V en su Raspberry Pi, que puede suministrar suficiente energía a HuskyLens.

### 9.1 Guía de codificación

1. Descargar librería [HuskyLens Python Library](#).
2. Coloque el `huskylensPythonLibrary.py` en su carpeta de proyectos.
3. En su archivo de Python (por ejemplo, `test.py`), importe la biblioteca de uso

`de huskylensPythonLibrary import HuskyLensLibrary`

`Init the HuskyLens`

`my_Var= HuskyLensLibrary("I2C","",address=0x32)`

`Now begin calling functions !`

#### CODE

```
1 # Check if HuskyLens can recieve commands
2 print(my_Var.command_request_knock())
3 # Get all the current blocks on screen
4 blocks=my_Var.command_request_blocks()
5 # Print the data
6 print(blocks)
```

Arduino Copy



## 9.4 Introducción a la API

### CODE

```
1 command_request()
2 => Return all data
3
4 command_request_blocks()
5 => Return all blocks on the screen
6
7 command_request_arrows()
8 => Return all arrows on the screen (only in line tracking mode)
9
10
11 command_request_learned()
12 => Return all learned objects on screen
13
14 command_request_blocks_learned()
15 => Return all learned blocks on screen
```

## 9.5 Proyecto 1: Leer datos de posición

En este proyecto, utilizamos la Raspberry Pi para leer los datos de posición del objeto de HuskyLens. El protocolo de comunicación es I2C.

### Requirimientos

#### · Hardware

- o Raspberry Pi 3 Model B+ (o similar) x 1
- o HUSKYLENS x 1
- o M-M/F-M/F-F Conectores para cableado

#### · Software

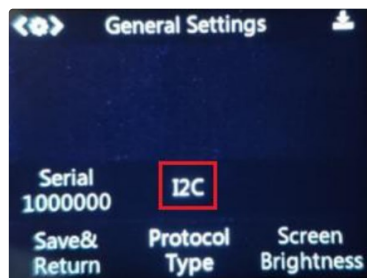
- o HUSKYLENS Python Library

### Diagrama de conexión

Consulte el apartado "9.2 Guía de cableado de I2C".

### Configuración del protocolo HuskyLens

Debe configurar el tipo de protocolo de HuskyLens. El protocolo debe ser I2C.



### Código de muestra

El siguiente código está en test.py. [Hacer click aquí](#) para verlo y descargarlo.

### CODE

```
1 # Import the library
2 from huskylensPythonLibrary import HuskyLensLibrary # Initialize the HuskyLens
3 test = HuskyLensLibrary("I2C","",address=0x32)
4 print("First request a knock: {}".format(test.command_request_knock()))
5
6 # Change to face recognition algorithm test.command_request_algorithm("ALGORITHM_FACE_RECO
7
8 # Display a simple menu where you can call every function in a loop!
9 ex=1
10 print("""
11 Menu options:
12 1)command_request()
13 2)command_request_blocks()
14 3)command_request_arrows()
```

### Operaciones y resultados esperados

1. Run the following code in the terminal on your Raspberry Pi.

```
python3 test.py
```

1. Deje que sus HuskyLens aprendan primero algo nuevo, p. tu cara. Puede consultar los capítulos anteriores de este tutorial.
2. Apunte la HuskyLens a su cara. Luego ingrese el número de comando en su terminal. Obtendrá los resultados de la siguiente manera:

```
pi@raspberrypi:~/Desktop/test/huskylens $ python3 test.py
First request a knock: Knock Recieved

Menu options:
1) command_request()
2) command_request_blocks()
3) command_request_arrows()
4) command_request_learned()
5) command_request_blocks_learned()
6) command_request_arrows_learned()
7) command_request_by_id() ***format 7 ID_VAL***
8) command_request_blocks_by_id() ***format 8 ID_VAL***
9) command_request_arrows_by_id() ***format 9 ID_VAL***
10) Exit

Enter cmd number:1
[[151, 116, 64, 87, 1]]
Enter cmd number:2
[[141, 118, 65, 86, 1]]
Enter cmd number:3
[]
Enter cmd number:4
[[132, 123, 65, 87, 1]]
Enter cmd number:5
[[125, 125, 65, 86, 1]]
```

```
Enter cmd number:8:1  
[[124, 130, 65, 86, 1]]  
Enter cmd number:1
```

Los resultados proporcionan las coordenadas x,y, el ancho, la altura del marco en la pantalla y la ID del objeto. El formato se muestra a continuación:

[Centro X del bloque, Centro Y del bloque, Ancho del bloque, Altura del bloque, Índice de los elementos aprendidos]

## 10. micro:bit Tutorial

En este capítulo, utilizaremos la placa **micro:bit** para leer datos de **HuskyLens**. El protocolo de comunicación es **I2C**. Adoptamos **Mind+** y **MakeCode**, luego lo combinamos con varios casos de proyectos para demostrar el uso.

Ahora adoptamos Mind + para la demostración primero.

### 10.1 Mind+ Introducción

Mind + es una herramienta de programación basada en Scratch 3.0, que le permite crear un programa arrastrando y ajustando bloques de codificación. Con toneladas de tutoriales, proyectos de muestra y una gran comunidad, ¡es una de las mejores herramientas para aprender a programar desde cero!

Mind + admite una amplia gama de hardware, incluido Arduino, [micro:bit](#) o incluso una serie de microcontroladores educativos basados en ESP32.

Por favor vea el sitio oficial de Mind + <http://mindplus.cc> para descargar la última versión de Mind +.

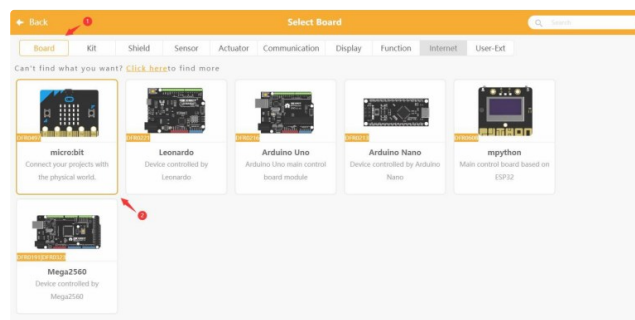
### 10.2 Carga de la extensión HuskyLens

1. En la esquina superior derecha de la ventana de Mind +, marque el interruptor al modo fuera de línea.



1.1. Haga clic en el botón Extensiones en la esquina inferior izquierda para ver la ventana de extensiones.

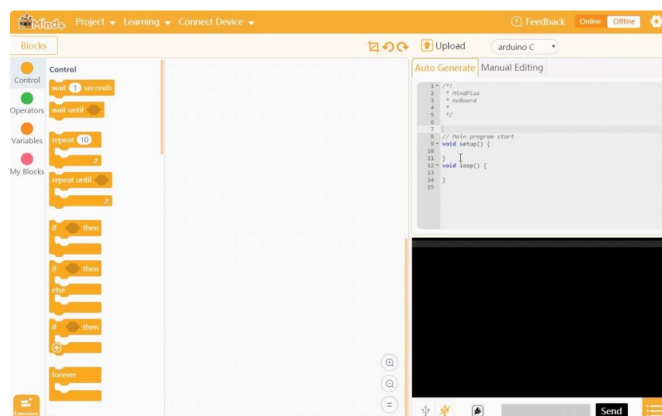
1.2. Seleccione una placa de control principal. Aquí seleccionamos el micro: bit.



2. Después de seleccionar la placa de control principal, las etiquetas de extensión correspondientes se pueden seleccionar. Luego haga clic en Etiqueta del sensor, busque HuskyLens. Haz clic para cargar.You can enter the sensor's name in the search bar in upper right corner when there are many sensors listed in the menu.



3. Después de seleccionar el panel de control principal y las extensiones, haga clic en el botón Atrás en la esquina superior izquierda para volver a la ventana de programación.



### 10.3 Mind + Proyecto 1: reconocimiento facial

Este capítulo muestra cómo conectar HuskyLens a la placa micro: bit, luego la placa micro: bit lee los resultados del reconocimiento facial de HuskyLens. Si HuskyLens lo reconoce (la cara aprendida), la pantalla de matriz de puntos del micro: bit muestra una cara sonriente, de lo contrario, muestra una cara triste.

El protocolo de comunicación entre HuskyLens y micro: bit es **I2C**.

### Requirimientos

· Hardware

o [micro:bit board](#) x 1

o [micro:bit expansion board](<https://www.dfrobot.com/search-micro:bit-expansion.html>) x 1

o [HUSKYLENS](#) x 1

o M-M/F-M/F-F Jumper wires

· Software

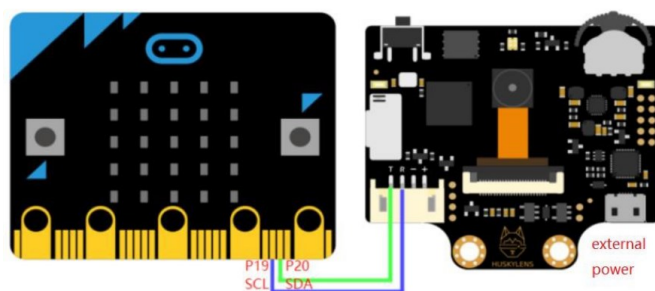
o [Mind+](#)

o HUSKYLENS Extension: Mind+ built-in

### Diagrama de conexión

La siguiente imagen es solo de referencia cuando se realiza el cableado. Los pines R y T de HuskyLens (sus funciones son SCL y SDA aquí) están conectados a los pines SCL (P19) y SDA (P20) del micro: bit respectivamente. El protocolo de comunicación entre HuskyLens y micro: bit es I2C.

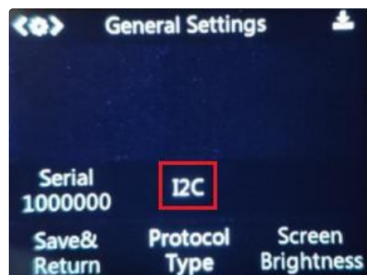
Se recomienda una placa de expansión micro: bit para simplificar el cableado.



Consejos: HuskyLens consume mucha corriente, hasta 3.3V 320mA o más. La placa micro: bit no es suficiente para suministrar energía. Por lo tanto, se requiere una fuente de alimentación externa. Puede conectar la fuente de alimentación externa al conector de alimentación externa de la placa de expansión micro: bit o al conector USB de HuskyLens.

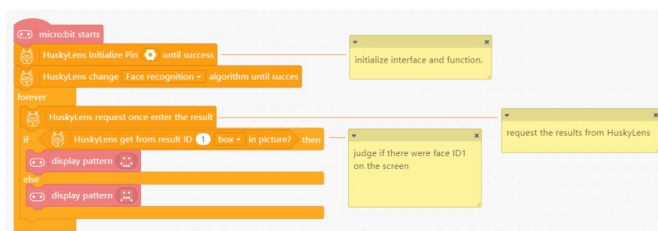
### Configuración del protocolo HuskyLens

Debe configurar el tipo de protocolo de HuskyLens. El protocolo debe ser I2C. Por supuesto, puede adoptar el protocolo de detección automática, que es fácil de usar y conveniente.



### Código de muestra

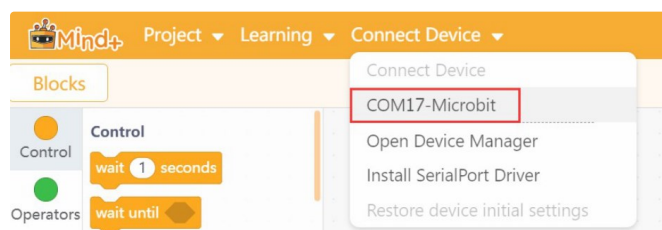
Arrastre y ajuste los bloques de codificación para programar, y realice proyectos simples de reconocimiento facial. El código de muestra se muestra a continuación.



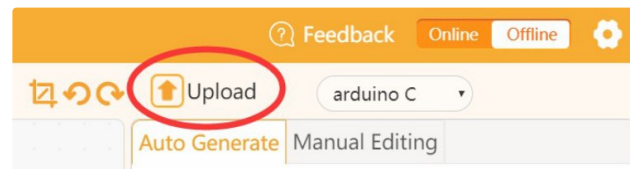
### Operaciones y resultados esperados

1. Haga clic en el botón Conectar dispositivo, luego seleccione el puerto COM correspondiente al micro:bit.

Si no se encuentra ningún puerto COM o usa micro:bit por primera vez, haga clic en el botón Instalar controlador de puerto serie para instalar el controlador con un solo clic.



2. Haga clic en el botón Cargar para cargar el código de muestra en su micro: placa de bits.



3. Deje que su HuskyLens aprenda primero su rostro. Puede consultar el capítulo 7.1 de este tutorial.

4. Cuando HuskyLens reconozca su cara, la pantalla de matriz de puntos en el tablero micro: bit mostrará una cara sonriente. Si no fuera tu cara, o no apareciera ninguna cara, mostraría una cara triste.

## 10.4 Introducción a la codificación

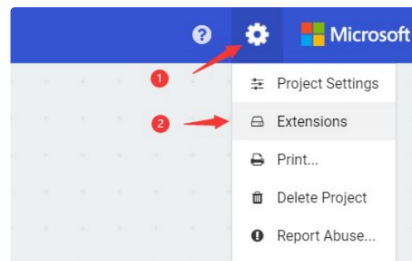
Adoptamos makecode para la demostración a continuación.

### 10.4.1 Introducción a MakeCode

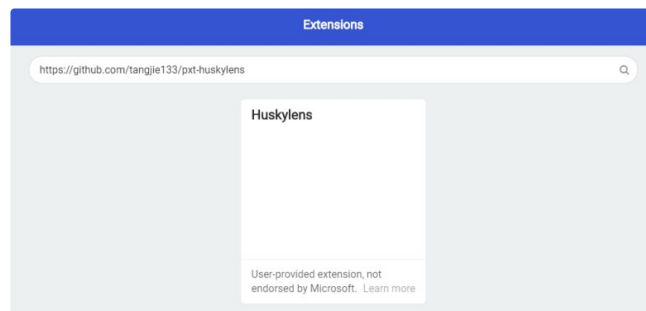
Microsoft MakeCode es una plataforma gratuita de código abierto para crear experiencias de aprendizaje informáticas atractivas que apoyan un camino de progresión hacia la programación del mundo real. [Click here](#) para ver el MakeCode para micro: bit.

#### 10.4.2. Cargar la Extensión HuskyLens

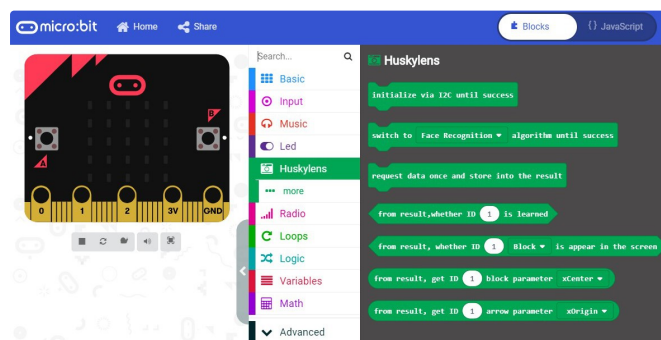
1. Crea un nuevo proyecto en [MakeCode](#) versión web, y luego haga clic en el botón "Más ..." (icono de engranaje) en la parte superior derecha y seleccione "Extensiones" en su menú desplegable para abrir la página de extensión.



2. Ponga <https://github.com/tangjie133/pxt-huskylen> en la barra de búsqueda, luego haga clic en el botón de búsqueda (el botón de lupa a la derecha de la barra de búsqueda), verá las extensiones de HuskyLens. Luego haga clic para cargar la extensión HuskyLens en MakeCode.



3. En la página de programación, puedes ver el módulo Huskylense.



### 10.4.3. Proyecto 1: Reconocimiento Facial

Este capítulo muestra cómo conectar HuskyLens a la placa micro: bit, luego la placa micro: bit lee los resultados del reconocimiento facial de HuskyLens. Si HuskyLens lo reconoce (la cara aprendida), la pantalla de matriz de puntos del micro: bit muestra una cara sonriente, de lo contrario, muestra una cara triste.

El protocolo de comunicación entre HuskyLens y micro: bit es I2C.

## Requirements

### Hardware

- o [micro:bit board](#) x 1
- o [micro:bit expansion board](#) x 1
- o [HUSKYLENS](#) x 1

o M-M/F-M/F-F Jumper wires

#### · Software

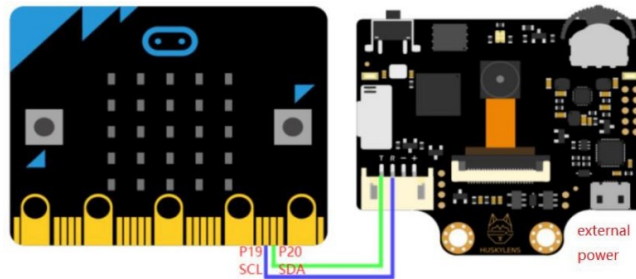
o [Microsoft MakeCode for micro:bit](#)

o [HUSKYLENS MakeCode Extension](#)

### Diagrama de conexiones

La siguiente imagen es solo de referencia cuando se realiza el cableado. Los pines R y T de HuskyLens (sus funciones son SCL y SDA aquí) están conectados a los pines SCL (P19) y SDA (P20) del micro: bit respectivamente. El protocolo de comunicación entre HuskyLens y micro: bit es I2C.

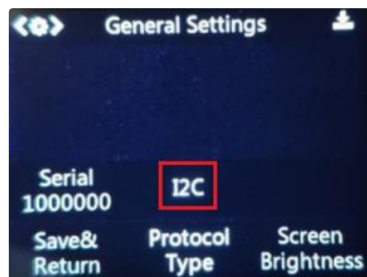
Se recomienda una placa de expansión micro: bit para simplificar el cableado.



**Consejos:** HuskyLens consume mucha corriente, hasta 3.3V 320mA o más. La placa micro: bit no es suficiente para suministrar energía. Por lo tanto, se requiere una fuente de alimentación externa. Puede conectar la fuente de alimentación externa al conector de alimentación externa de la placa de expansión micro: bit o al conector USB HuskyLens.

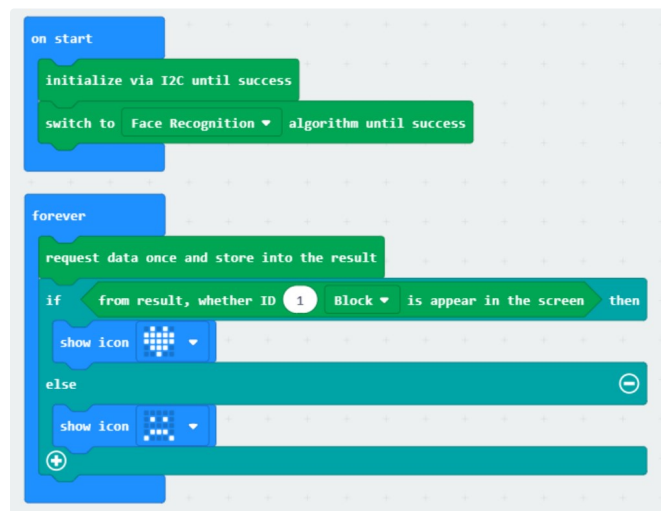
### Configuración del protocolo HuskyLens

Debe configurar el tipo de protocolo de HuskyLens. El protocolo debe ser I2C. Por supuesto, puede adoptar el protocolo de detección automática, que es fácil de usar y conveniente.



#### 10.4.4. Introduccion del Código Bloques

##### Código



### Operaciones y resultados esperados

1. Cargue los códigos anteriores en la placa micro: bit.
2. Consulte el capítulo anterior que explica la función de reconocimiento facial (capítulo 7.1), deje que su HuskyLens aprenda una cara, como su cara.
3. Cuando HuskyLens reconozca su cara, la pantalla de matriz de puntos en el tablero micro: bit mostrará una cara sonriente. Si no fuera tu cara, o no apareciera ninguna cara, mostraría una cara triste.

## 11. Preguntas más frecuentes

**Pregunta.** Cuando subo el firmware mediante el k-flash, pero aparece el mensaje de error "no se pudo encontrar el registro del directorio central". ¿Cómo puedo arreglar esto?

**Respuesta.** El firmware que descargó no es correcto, es decir, el archivo está incompleto e incorrecto. (por ejemplo, el firmware correcto es de hasta 9 MB, pero el firmware que descarga es de solo 66 KB). Puede deberse a la red inestable u otros asuntos. Por favor, descárguelo nuevamente de Github.



## 12. Documentación

- [Librería Arduino\(github\)](#)
- [Librería Raspberry Pi Python \(github\)](#)
- [Librería micro:bit Makecode\(github\)](#)
- [Protocolo Documento](#)
- [Imágenes de Marcas](#)
- [Muestras de Colorers](#)
- [3D model file\(.stp\)](#)
- [Descargar .NET Framework 4.7.1 para K-Flash](#)
- [Descarga WIKI Pagina\(PDF\)](#)
- [Foro](#)
- [Comunidad](#)

**Nota:** Este trabajo es una traducción y adaptación del documento que se muestra en la [Wiki de DFRobot](#)

**Agradecimiento especial a DFRobot**, que me ha facilitado todo cuanto he necesitado para poder evaluar este producto y desarrollar este

manual. [www.dfrobot.com](http://www.dfrobot.com)

**Special thanks to: DFRobot**, who has provided me with everything I have needed to evaluate this product and develop this manual. [www.dfrobot.com](http://www.dfrobot.com)

Profesor: **José Manuel Ruiz Gutiérrez**

[j.m.r.gutiérrez@gmail.com](mailto:j.m.r.gutiérrez@gmail.com)

Julio 2020

---

Author: Profesor: **José Manuel Ruiz Gutiérrez.**

Posted by **DFRobot Team**.

License



Tags

#DFRobot

#AI



1



0



Education

Enter Keywords



LOGIN/SIGN UP

HOME

COMMUNITY

FORUM

WIKI

BLOG

EDUCATION

Post a comment and please be nice.

Post



### INFORMATION

[About Us](#)  
[Warranty](#)  
[Privacy Policy](#)  
[Shipping](#)  
[Payment](#)  
[FAQ](#)

### CUSTOMER SERVICE

[DFRobot Distributors](#)  
[Contact us](#)  
[Site Map](#)

### MY ACCOUNT

[Affiliates](#)  
[Specials](#)  
[Coupon](#)