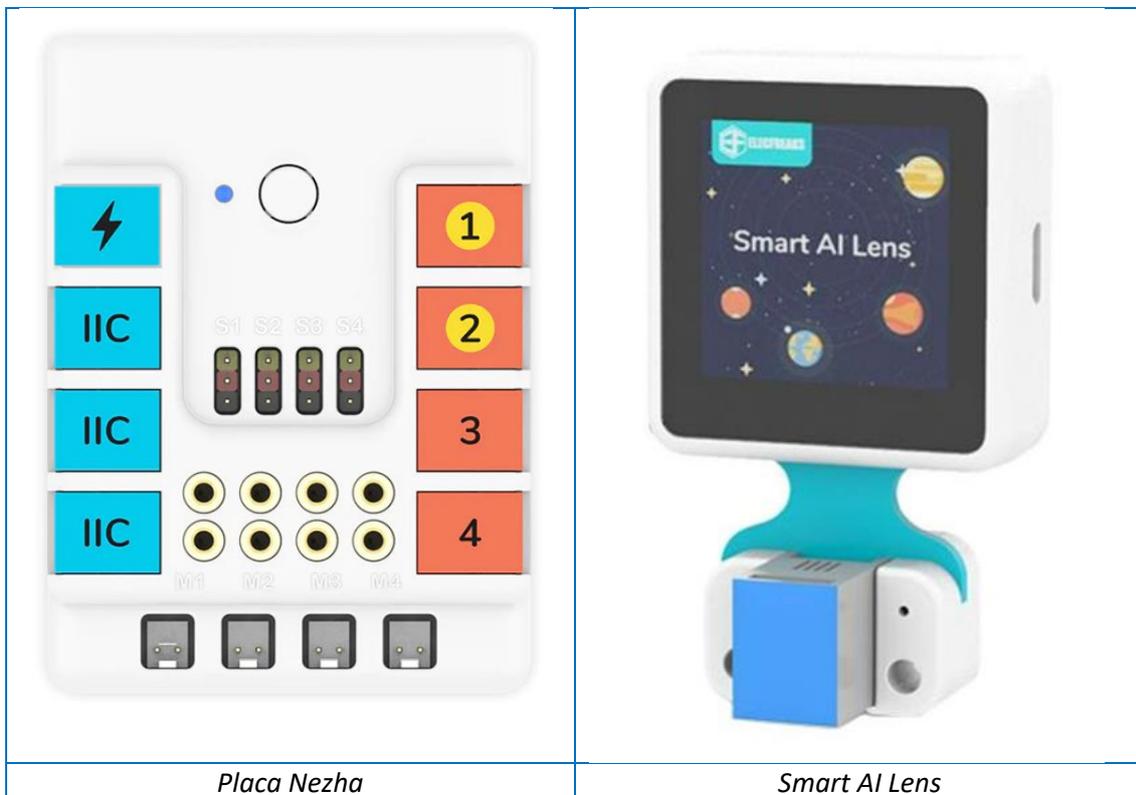


## Índice

1. ¿Qué es Smart AI Lens? .....	2
1.1. ¿Cómo instalar la última versión del firmware? .....	4
2. Programación .....	7
3. Placa Nezha .....	11
4. Ejemplos de programación por bloques para Smart AI Lens .....	12
4.1. Reconociendo las imágenes de las tarjetas.....	12
5. Proyectos combinados: Smart AI Lens con Nezha Inventor kit.....	16
5.1. Control automático de una barrera de parking .....	17

## 1. ¿Qué es Smart AI Lens?

Smart AI Lens es un producto de la casa ElecFreaks que nació como un sensor más, compatible con los elementos del proyecto Planet X que utilizan como cerebro la placa [Nezha](#):



Placa Nezha

Smart AI Lens

Incorpora una cámara por su parte trasera y una pantalla TFT de 1,3 pulgadas frontal que nos permite *analizar imágenes a tiempo real, reconocer caras, colores y formas, hacer un seguimiento de línea o de bolas y además, logra aprender.*

Su cámara se compone de unas lentes con inteligencia artificial con un ángulo de visión de 90°. El módulo funciona a 3,3V y relativo a la pantalla TFT es importante saber que el punto de origen de la Smart AI Lens se sitúa en la esquina superior izquierda (0,0), siendo el rango de las coordenadas X e Y de (0,224).

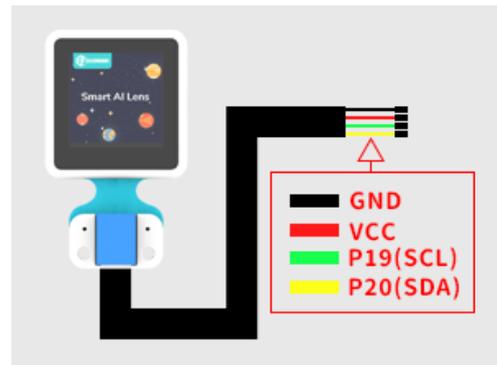
Este sensor de AI incorpora un chip con tecnología RIS5, que es una tecnología open hardware abierta puntera a la hora de diseño de chips con tecnología RIS de ARM. En este sentido, es novedoso y “con vistas” para quedarse.

El kit Smart AI Lens viene con dos cables, ambos de 200mm:

- Uno de ellos con conectores RJ11 a ambos extremos y que nos permite conectar la cámara al bloque Nezha (compatible con micro: bit)

- El otro cable es un conector RJ11 a dupont hembra y nos permite conectar la cámara a otras placas de expansión de la micro: bit.

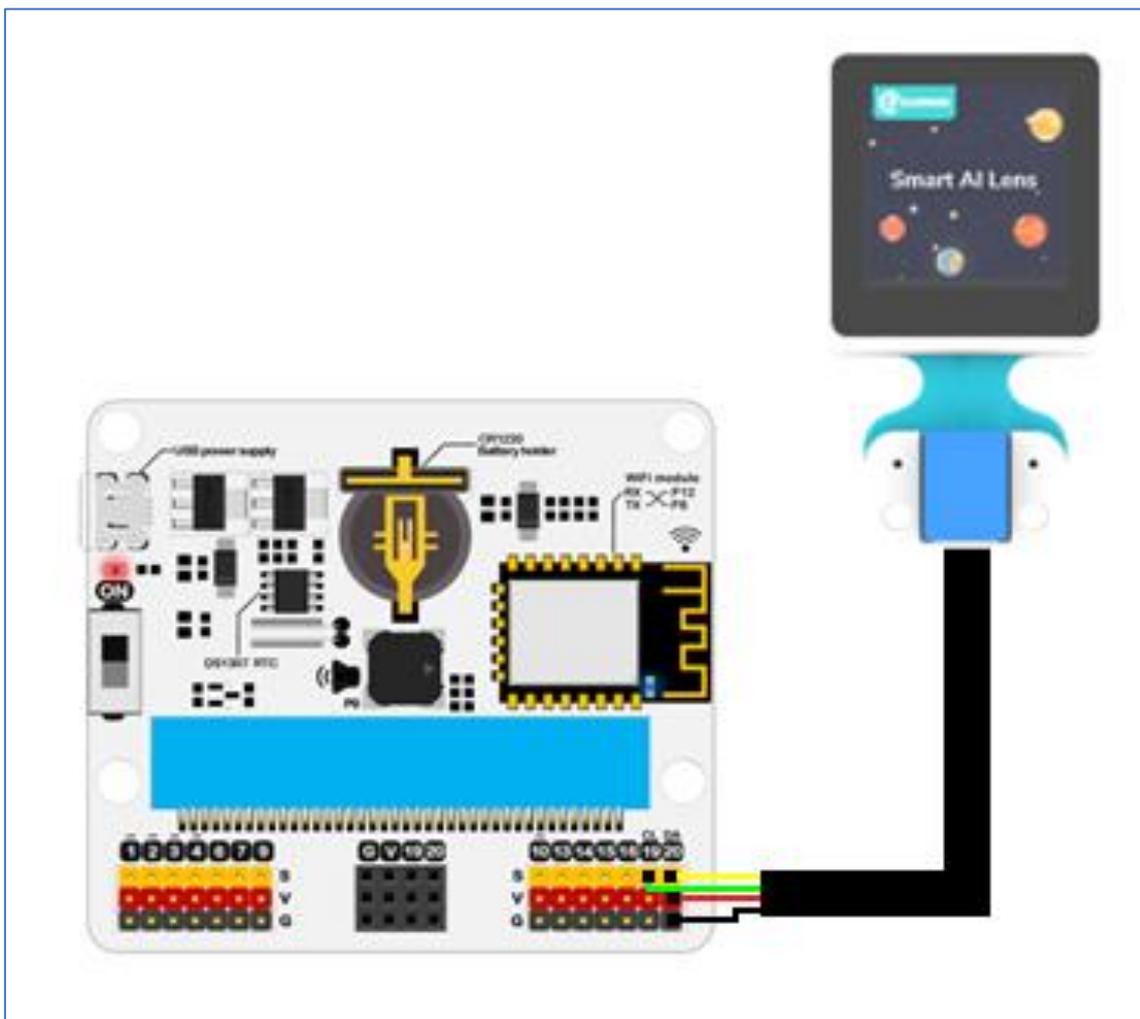
Este segundo cable nació de una necesidad: obviamente, la placa micro: bit no dispone de conectores RJ11 pero, en ella y sobre todo en sus placas de expansión podemos usar conectores dupont. De ahí la utilidad de un cable RJ11 a dupont. Necesitamos conectar el extremo RJ11 al dispositivo Smart AI Lens y los otros 4 conectores dupont se conectarán de la siguiente forma a una placa de expansión para la micro: bit: (Ver imagen)



Conexionado

- cable negro a GND
- rojo a VCC
- verde a P19 (SCL)
- amarillo a P20 (SDA)

Si el dispositivo no funciona desde el principio, posiblemente nos encontremos con un problema de alimentación y deberemos acoplarle una fuente de alimentación independiente.



El kit incluye dos bolas (roja y azul), una serie de tarjetas con diferentes imágenes, colores, letras y números y por último el sensor inteligente que por su parte trasera, como ya se explicó, dispone de una pantalla TFT pequeña de 2 o 3 cm<sup>2</sup> y por el otro lado tiene una cámara. Por último, en un lateral dispone de un conector USB, que se usará para actualizar el firmware.

El número de la versión actual, tras alimentarlo, se mostrará en la esquina inferior derecha de la pantalla de inicio. Ojo, si no vemos este aviso, su versión es antigua y debemos actualizar su firmware.



### 1.1. ¿Cómo instalar la última versión del firmware?

Descargar el firmware: la última versión es esta

firmware v1.0.12 [haga clic para descargar](#)

1. Instalación del controlador del puerto serie de la cámara

Sistema Windows descargando [CH341SerSetup.exe](#)

sistema macOS descargando [CH34x Install V1.5.pkg](#)

2. Instalamos “kflash” para descargar el firmware.

Sistema Windows descargando [kflash\\_gui v1.6.5 2 windows.7z](#) y descomprimiéndolo.

Sistema macOS descargando [kflash\\_gui v1.6.5 2 macOS.dmg](#)

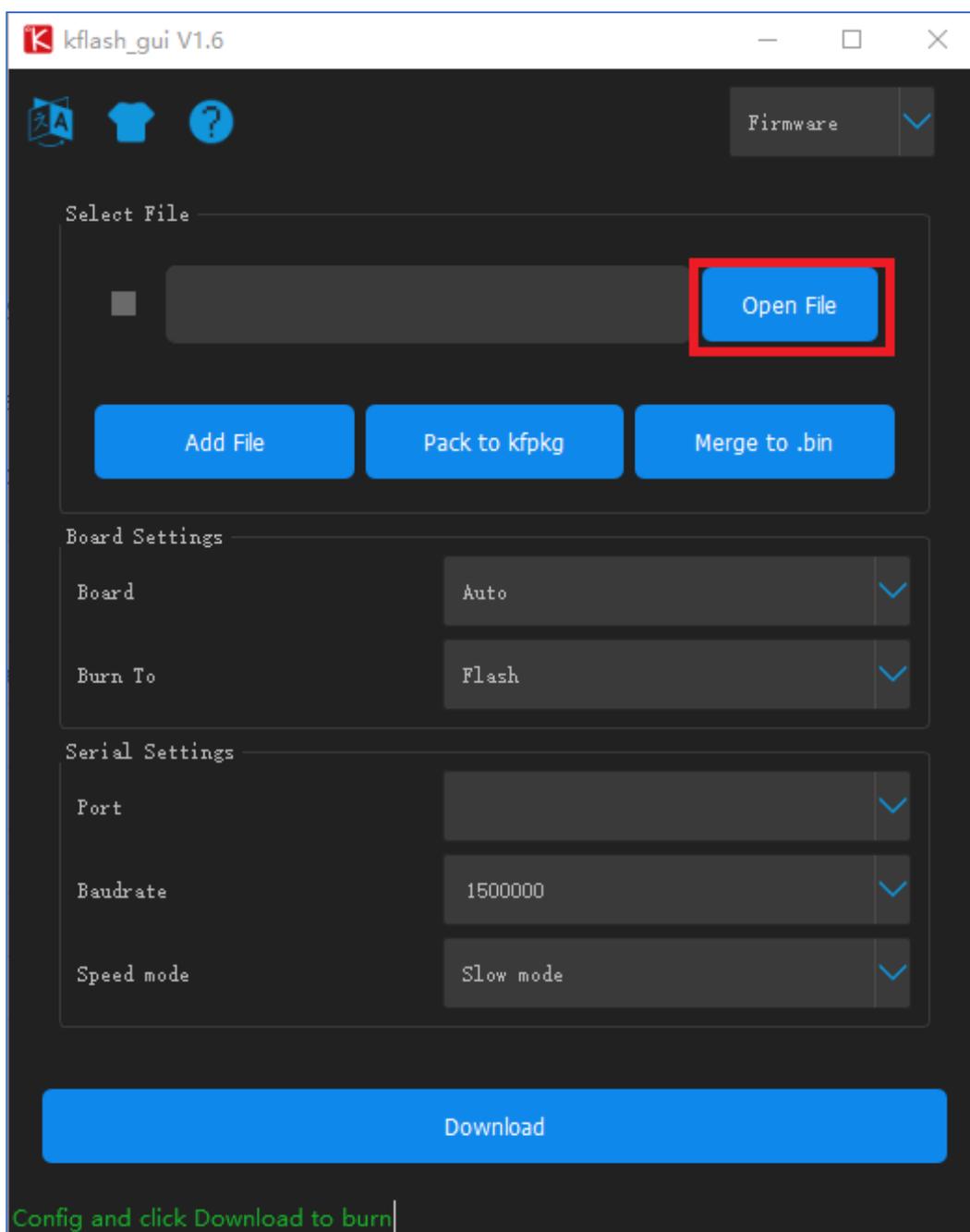
- ✚ Para el sistema Windows, abra la carpeta de archivos kflash\_gui y busque kflash\_gui.exe.

api-ms-win-crt-stdio-l1-1-0.dll	2020/7/29 18:41	应用程序扩展	27 KB
api-ms-win-crt-string-l1-1-0.dll	2020/7/29 18:41	应用程序扩展	27 KB
api-ms-win-crt-time-l1-1-0.dll	2020/7/29 18:41	应用程序扩展	23 KB
api-ms-win-crt-utility-l1-1-0.dll	2020/7/29 18:41	应用程序扩展	21 KB
base_library.zip	2020/7/29 18:41	WinRAR ZIP 压缩...	760 KB
d3dcompiler_47.dll	2020/7/29 18:41	应用程序扩展	4,077 KB
kflash_gui.conf	2021/1/5 17:49	CONF 文件	1 KB
<b>kflash_gui.exe</b>	2020/7/29 18:41	应用程序	1,811 KB
kflash_gui.exe.manifest	2020/7/29 18:41	MANIFEST 文件	2 KB
libcrypto-1_1-x64.dll	2020/7/29 18:41	应用程序扩展	2,426 KB
libEGL.dll	2020/7/29 18:41	应用程序扩展	24 KB
libGLSLv2.dll	2020/7/29 18:41	应用程序扩展	3,290 KB
libssl-1_1-x64.dll	2020/7/29 18:41	应用程序扩展	518 KB
MSVCP140.dll	2020/7/29 18:41	应用程序扩展	577 KB
MSVCP140_1.dll	2020/7/29 18:41	应用程序扩展	31 KB
opengl32sw.dll	2020/7/29 18:41	应用程序扩展	20,433 KB

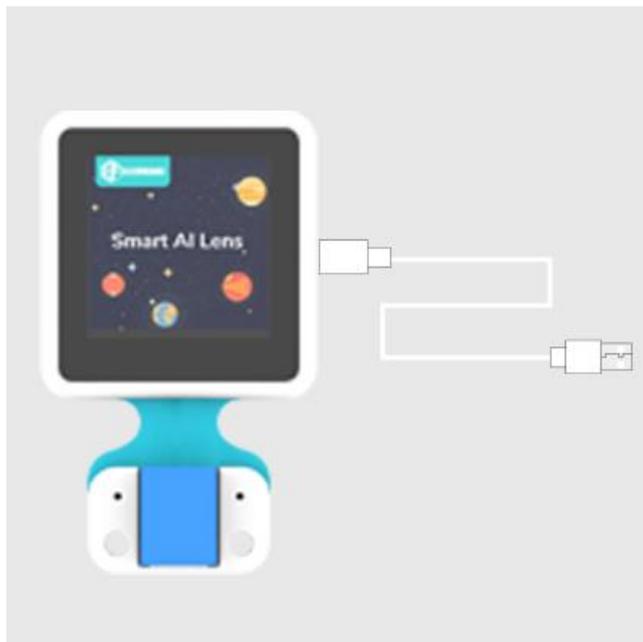
✚ Para el sistema macOS, haga clic en el icono de kflash\_gui después de la descarga.



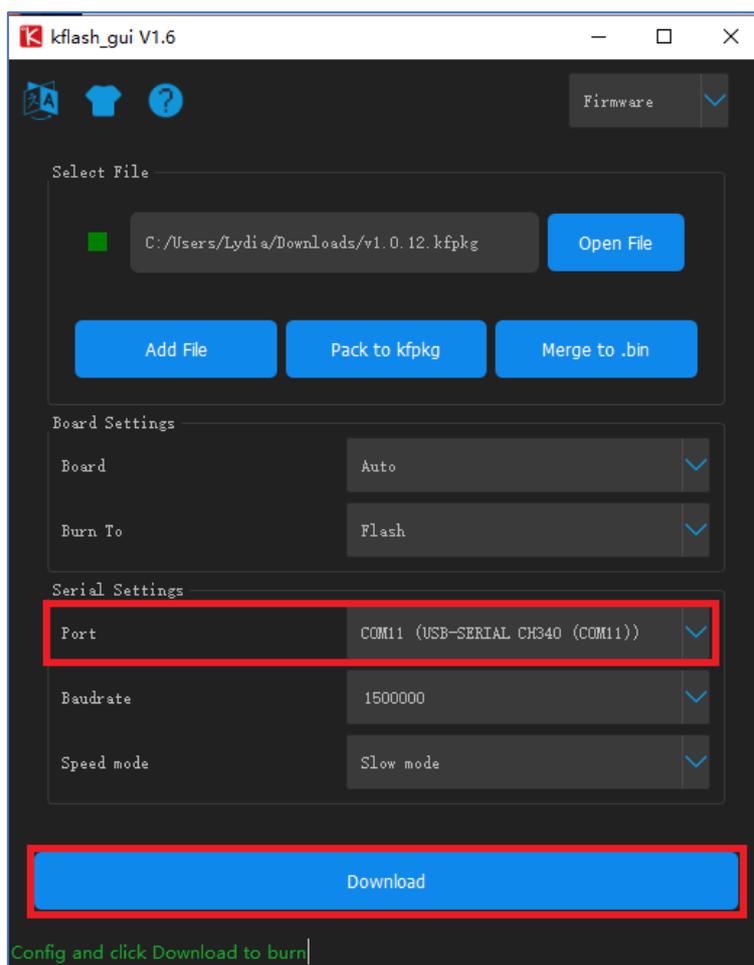
Haga doble clic en kflash\_gui.exe para abrir el archivo y elija abrir el firmware más reciente.



3. Conecte la lente AI con el cable USB.



4. Elija el puerto para descargarlo.



## 2. Programación

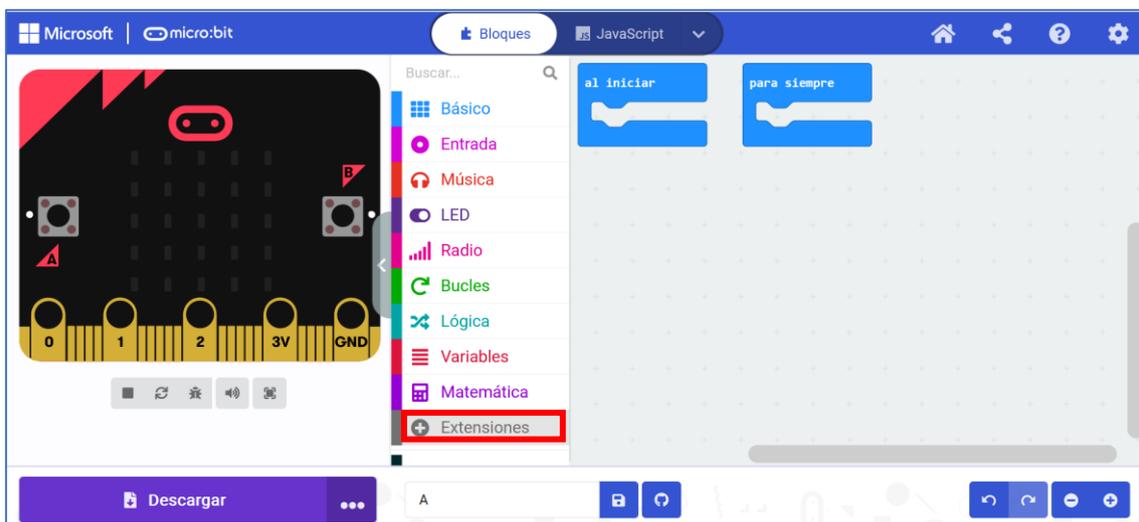
Utilizando la micro: bit, este módulo se puede programar en Python con su editor de micro: bit accesible en el siguiente link ([Python](#)) o por bloques, desde MakeCode para micro: bit ([bloques](#)).

En este tutorial nos centraremos en la programación por bloques. Para saber más, podemos usar el siguiente link para comenzar con ejemplos de programación con Python:

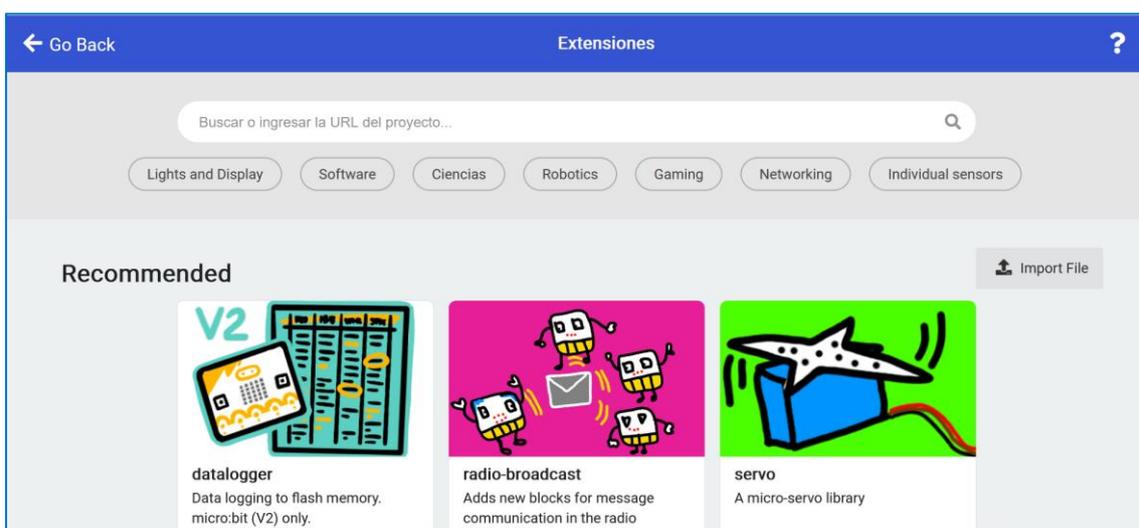
<https://www.electfreaks.com/learn-en/microbitplanetX/ai/Plant-X-EF05045-python-en.html>

Relativo a la programación por bloques en el entorno MakeCode micro: bit, necesitaremos instalar una o dos extensiones, dependiendo del uso de uno o más dispositivos. Me explico, puedo usar solo el módulo Smart AI Lens o combinarlo con motores o servomotores conectados a la placa Nezhá. Todas estas extensiones pueden incluirse fácilmente desde el entorno MakeCode para micro: bit: <https://makecode.microbit.org/>

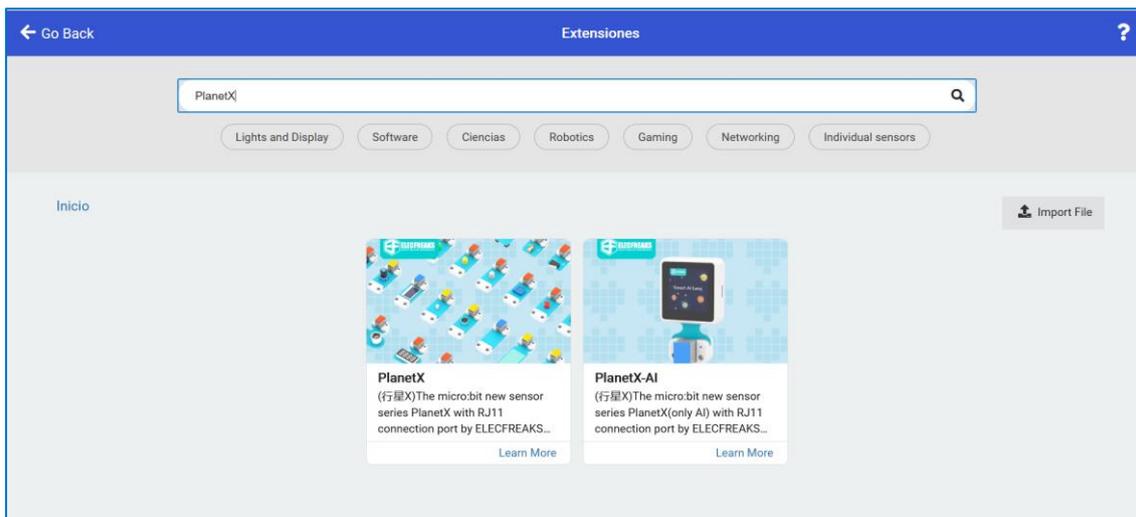
Primero debemos hacer click en “Extensiones”:



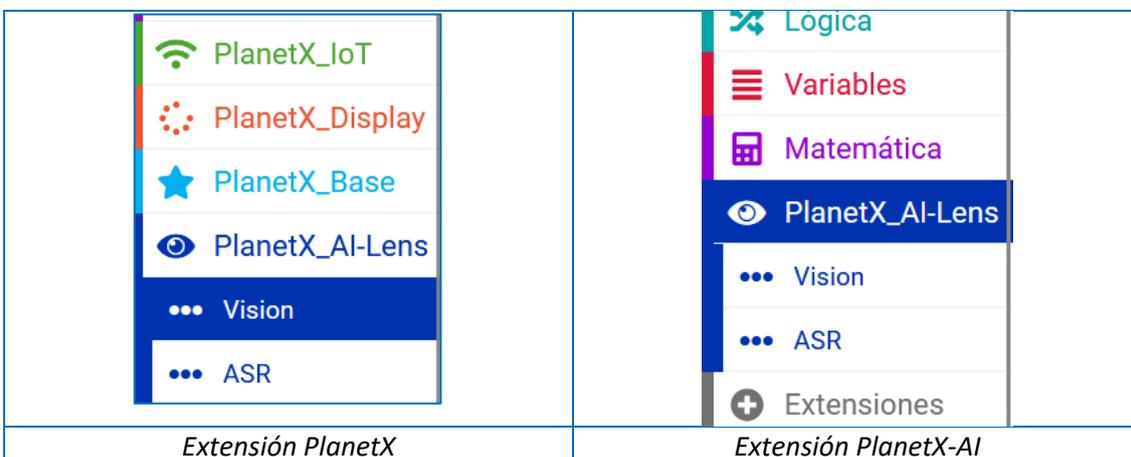
Nos llevará a esta pantalla:



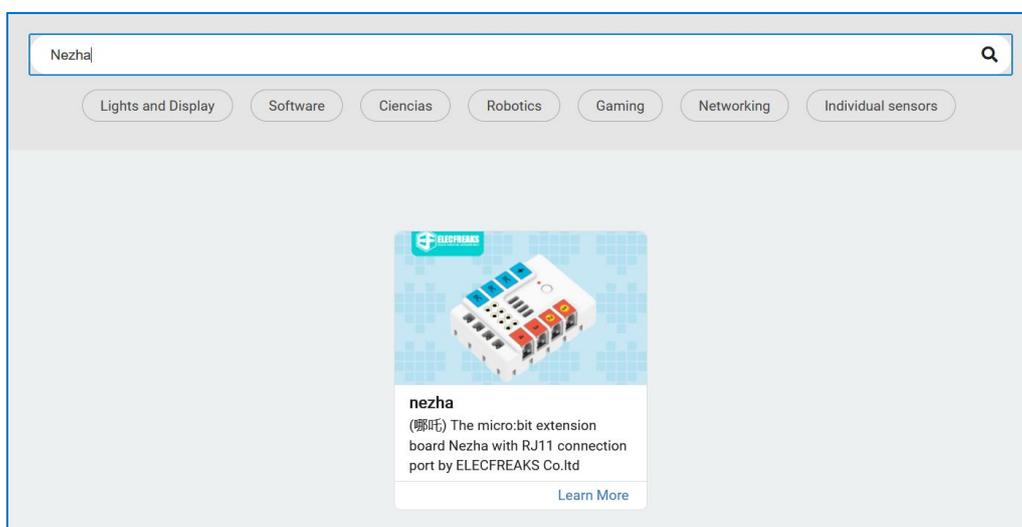
En el buscador de la misma escribimos “PlanetX”: Veremos que podemos añadir dos extensiones sin más que hacer click encima de ellas.



Si hacemos click en la extensión PlanetX, ya nos incluye la extensión de la cámara AI, pero si sólo vamos a usar la cámara, podemos cargar solo la extensión “PlanetX-AI” de la imagen anterior:



A veces, necesitaremos instalar la extensión de la placa Nezhah y del sensor AI. La extensión de la placa se busca de la misma forma que la de la cámara:

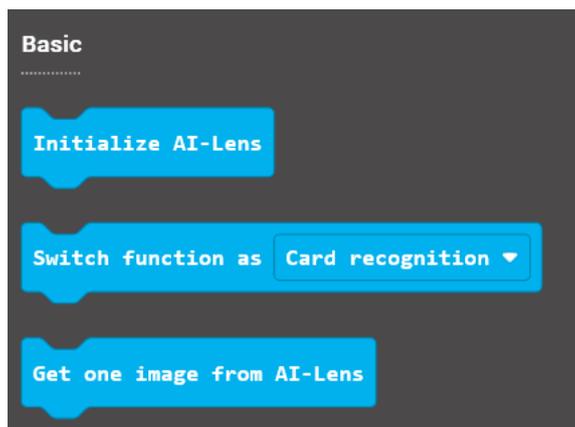


Vemos que la extensión de la cámara (PlanetX\_AI-Lens) carga dos grupos de bloques que se llaman "Visión" y "ASR".

- El de visión se compone de unos bloques básicos y de los bloques específicos para cada una de las 6 funciones que puede realizar y que pueden verse en la siguiente imagen

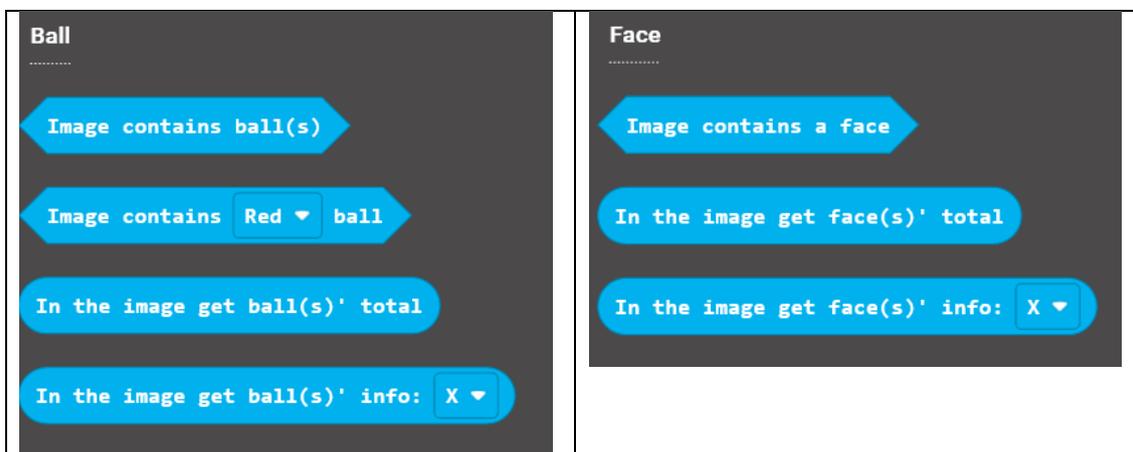


Funciones que puede realizar



Bloque básico

Comandos de las 6 funciones:



<p><b>Color</b></p> <p>Image contains color card(s): Black ▾</p> <p>In the image get color card(s)' total</p> <p>In the image get color card(s)' info: X ▾</p>	<p><b>Card</b></p> <p>Image contains number card(s): 0 ▾</p> <p>Image contains letter card(s): A ▾</p> <p>Image contains traffic card(s): Forward ▾</p> <p>Image contains other card(s): Mouse ▾</p> <p>In the image get Card(s)' total</p> <p>In the image get Card(s)' info: X ▾</p>
<p><b>Tracking</b></p> <p>In the image get line(s)' info: Angle ▾</p> <p>Image contains line's direction towards Left ▾</p>	<p><b>Learn</b></p> <p>Learn an object with: ID1 ▾</p> <p>Clear learned objects</p> <p>Image contains learned objects: ID1 ▾</p> <p>In the image get learn object ID1 ▾ Confidence</p>

- El bloque ASR (Audio Science Review) trabaja con audio

ASR sensor hear Hi, Shaun ▾

ASR sensor enter learning-model

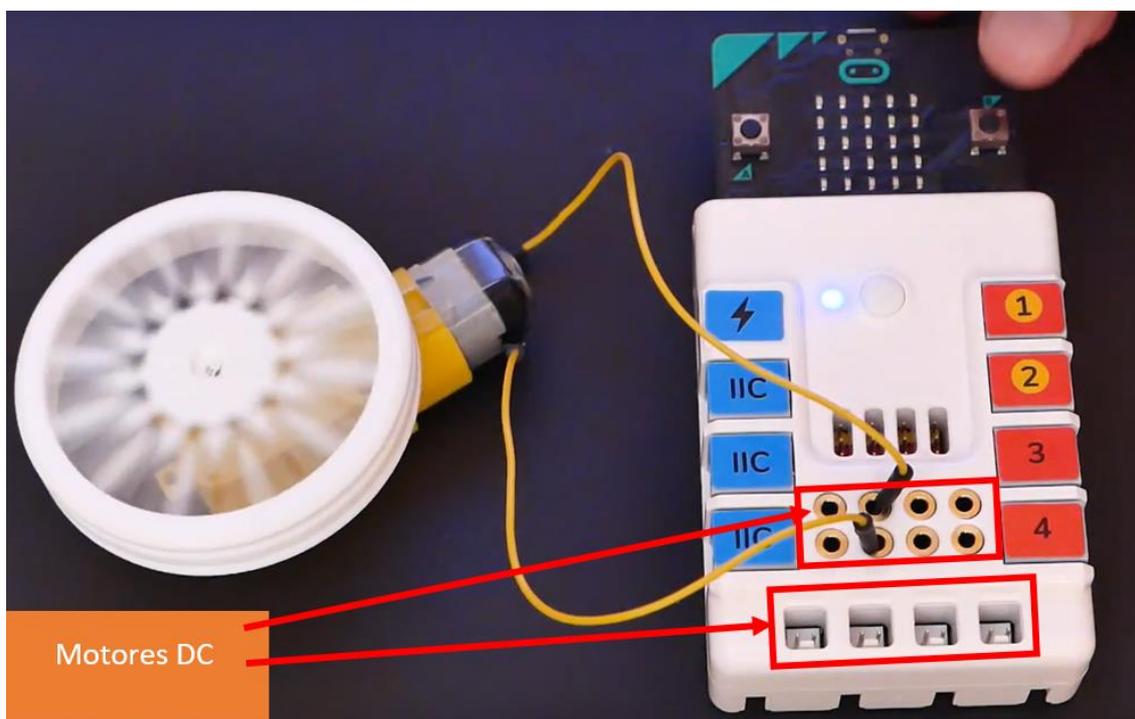
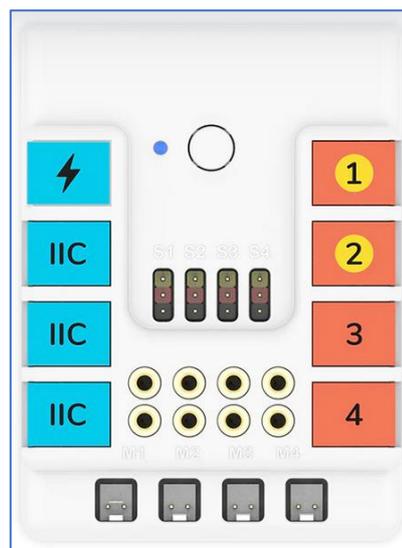
ASR sensor clear learned entrys

### 3. Placa Nezha

La placa Nezha dispone de un hub en el que conectar todos los elementos electrónicos, necesita una placa Micro: bit para que funcione como microprocesador o cerebro del sistema.

El hub ofrece **cuatro puertos de conexión de entrada/salida y tres puertos I2C para sensores y actuadores** del sistema ELECFREAKS. El tipo de conector que se utiliza en estos puertos es un RJ11 de 4 hilos a través cables macho-macho de diferentes tamaños.

Además, permite la conexión de **hasta cuatro servos y cuatro motores de continua**. Para los motores el hub ofrece dos tipos de conexión por lo que tenemos 8 puertos para un total de 4 motores:



Se puede alimentar de dos formas: A través de USB 5V o por su batería interna recargable Li-Ion 900mAh.

Lógicamente, a parte de los sensores y actuadores que incluye el kit de Nezha tenemos que añadir los que nos ofrece la propia placa Micro: bit, como son: su sensor de temperatura, acelerómetro, magnetómetro, sensor de luminosidad, matriz de LEDs y pulsadores. Con la Micro: bit V2 añadimos el sensor de sonido, pulsador táctil capacitivo y piezoeléctrico (buzzer).

Los contenidos educativos se concentran en este link:

[https://www.electfreaks.com/learn-en/microbitKit/Nezha\\_Inventor\\_s\\_kit\\_for\\_microbit/](https://www.electfreaks.com/learn-en/microbitKit/Nezha_Inventor_s_kit_for_microbit/)

Wiki que incluye una breve descripción del kit Nezha y 80 ejemplos prácticos que incluyen ejemplos de programación por bloques. No todos los ejemplos prácticos van asociados a un modelo de construcción, ya que para alguno de ellos solo se utiliza la placa Micro: bit junto con el hub. Sin embargo, la mayoría son modelos de construcción y se incluyen las instrucciones de montaje paso a paso.

Podemos ver otros [proyectos con Nezha en el canal de Youtube de ELECFREAKS](#) o en su [canal de Discord](#).

#### 4. Ejemplos de programación por bloques para Smart AI Lens

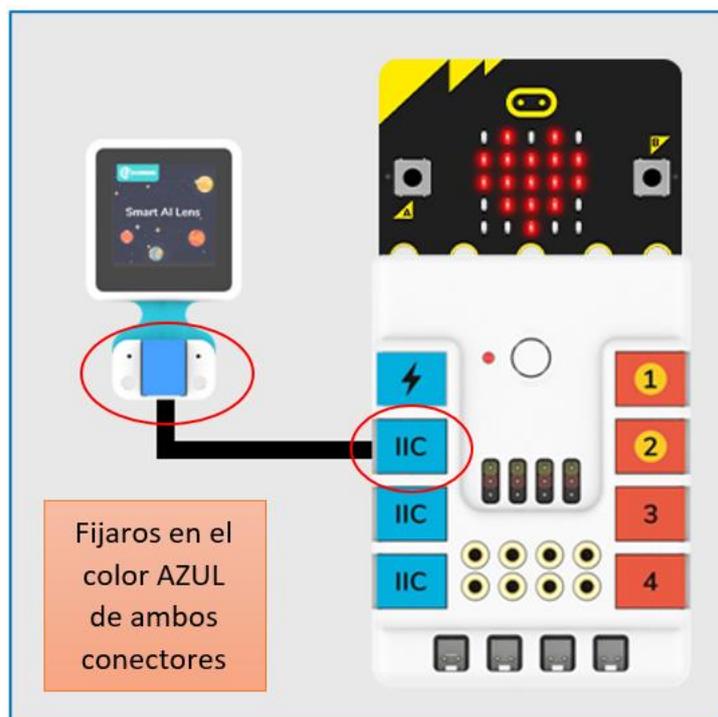
En la wiki de Elecfreaks podemos encontrar los [contenidos educativos](#) que han documentado como ejemplos de utilización. En ellos vemos ejemplos no sólo con programación por bloques (usando MakeCode) sino también, usando código Python.

Nosotros nos centraremos en MakeCode describiendo en cada ejemplo, de forma ordenada, los siguientes ítems: **Objetivo, materiales y conexionado, programación y comprobación.**

##### 4.1. Reconociendo las imágenes de las tarjetas

- ✚ Objetivo: La cámara incorpora unas tarjetas con imágenes. Vamos a ver cómo podemos ver si las reconoce o no.
- ✚ Materiales y conexionado: Utilizaremos la cámara AI, la placa Nezha y una micro: bit

Conectamos la cámara al bloque Nezha por medio del cable RJ11 a uno de los 3 módulo I2C del bloque introducimos la micro: bit dentro del módulo Nezha. (Fijaros que el color azul del módulo conector de la cámara se corresponde con el color del conector I2C del bloque Nezha).



Y por último, conectamos el cable micro USB de la micro: bit con el PC.

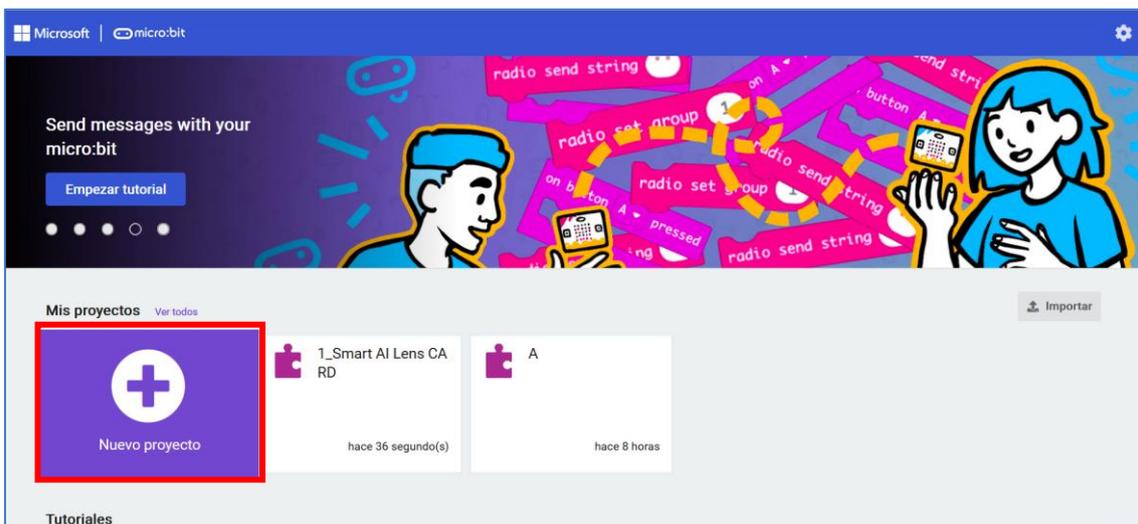


Smart AI Lens dispone de una cámara que va a captar la imagen, una pantalla que nos va a enviar la imagen de la cámara y sobre la pantalla se verán los datos que el sensor reconoce internamente.

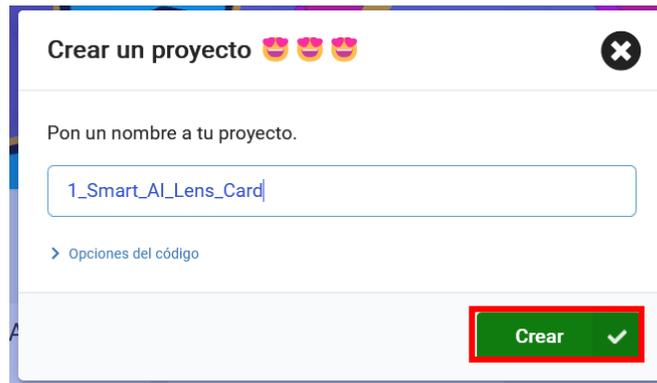
Encendemos el bloque o placa Nezha, para que alimente también por la batería, por si hiciera falta.

 Programación

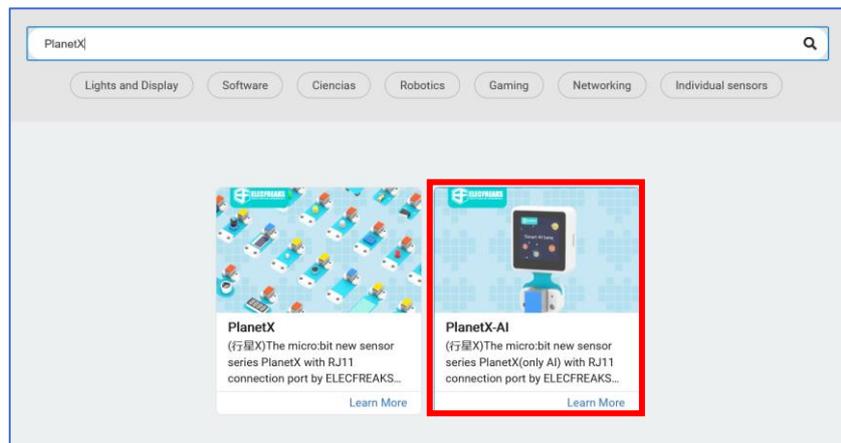
Desde MakeCode micro: bit creamos un nuevo proyecto:



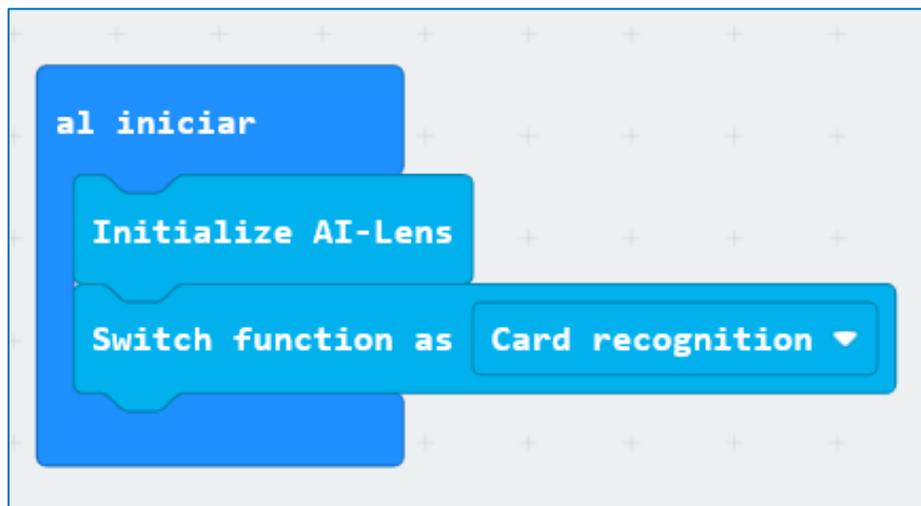
Nos pedirá un nombre. En mi caso le llamaré “1\_Smart\_AL\_Lens\_Card” y hacemos click en el botón verde “Crear”



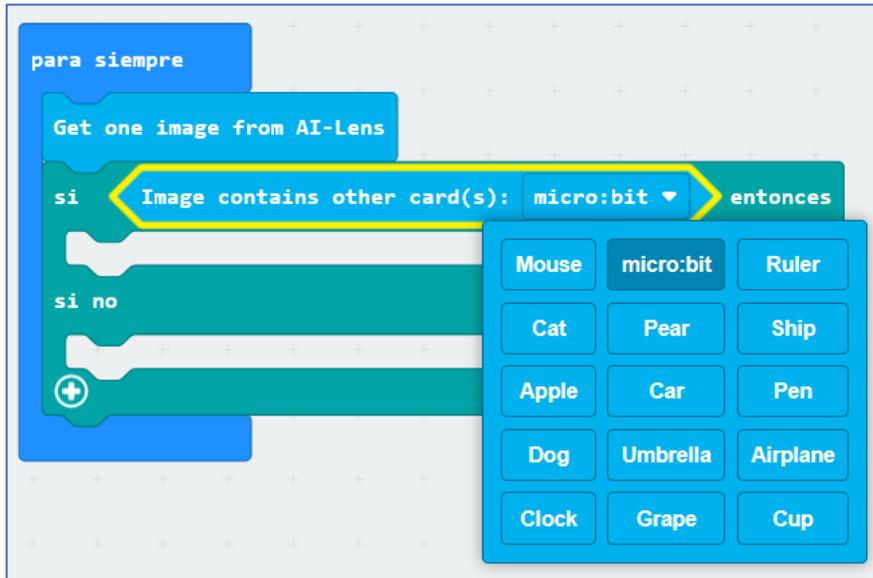
Cargaremos la extensión de la cámara, buscándola por “PlanetX”:



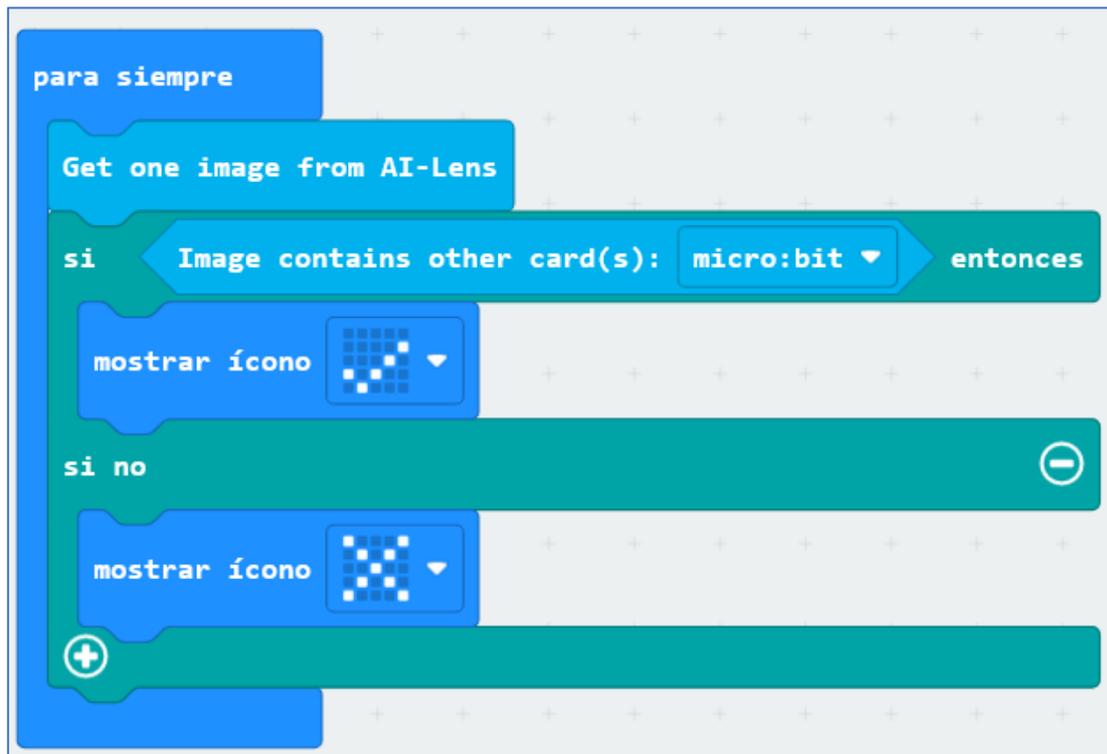
Ya podemos comenzar nuestro programa. Como primer paso, inicializamos el sensor de AI y, entre sus opciones o “funciones Switch”, escogemos la función de reconocimiento de tarjetas.



Entre las tarjetas que tiene el kit elegimos una para trabajar. Yo voy a usar la que nos muestra una imagen de una micro: bit. Obviamente, en el bloque “para siempre” se debe conseguir que testee si está viendo o no esa tarjeta. Es decir, debe **obtener** una imagen de la cámara y **comprobar** si en esa imagen ve el objeto “micro: bit”:

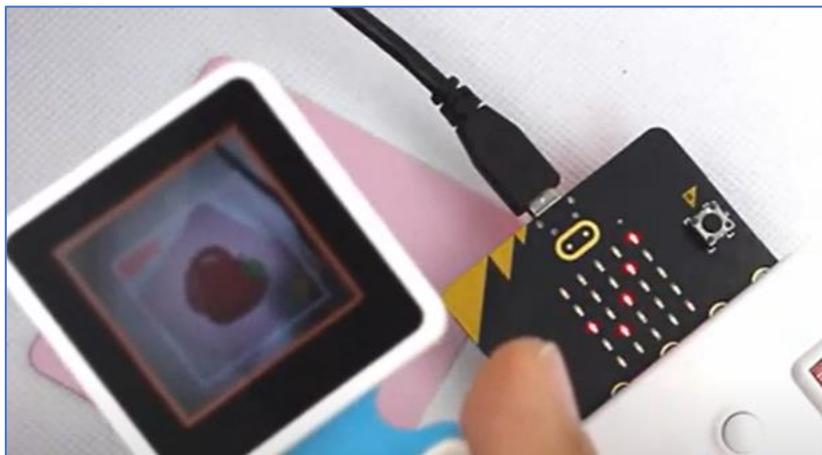


En caso de que la vea, nos lo hará saber y esta vez quiero que lo haga de forma visual usando la matriz de LEDs de nuestra micro: bit. Un posible script para conseguirlo sería el siguiente:

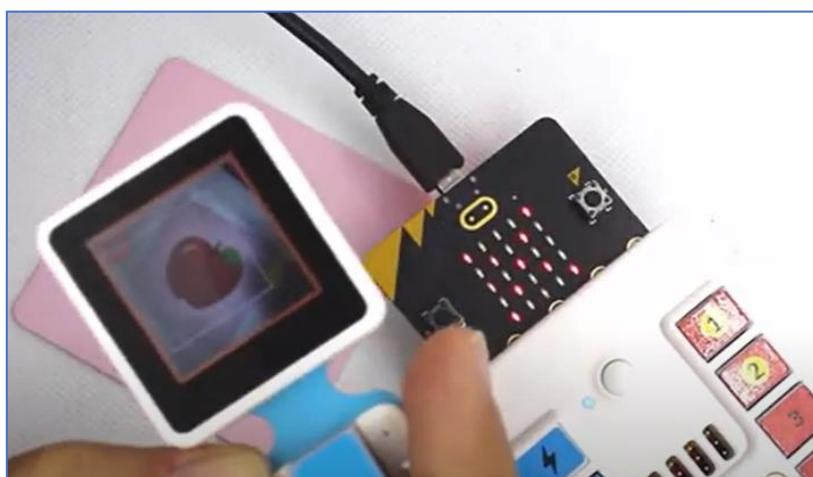


Por último, cargamos el programa y comprobamos si la reconoce o no. Veremos que funciona perfectamente.

Podemos obtener una imagen desde la cámara:



Puede reconocer hasta 4 tarjetas juntas



#### 5. Proyectos combinados: Smart AI Lens con Nezha Inventor kit

El potencial de esta cámara aparece cuando la unimos con los módulos, sensores y piezas del kit "Nezha Inventor Kit". Esta cámara se convierte en un módulo más para incluir en un diseño de un proyecto, y para esta finalidad fue creada. Veamos un ejemplo.



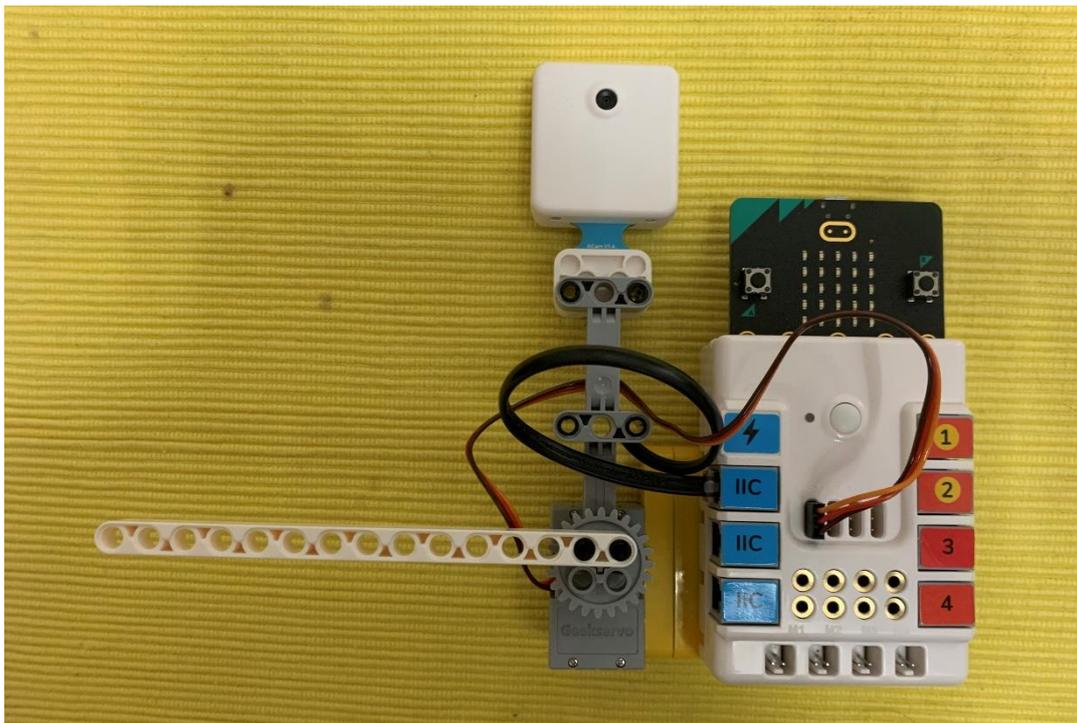


### 5.1. Control automático de una barrera de parking

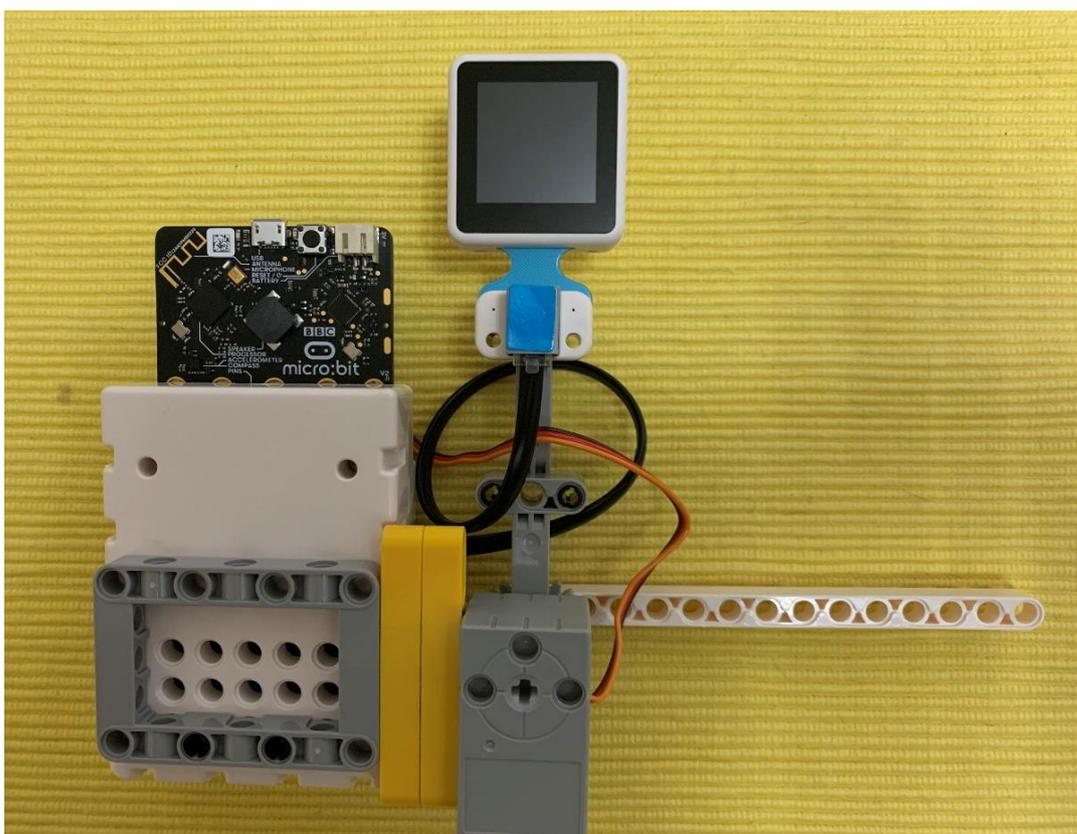
Vamos a modificar un proyecto que consiste en el control de una puerta (apertura y cerrado automático) mediante la detección por un módulo de ultrasonidos, cambiando el sensor de ultrasonidos por la detección que no ofrece la cámara Smart AI Lens.

- ✚ Objetivo: La cámara incorpora unas tarjetas con imágenes. Vamos a controlar la apertura o cierre de la puerta de un parking a través de una imagen (usaremos la tarjeta de una micro: bit).
- ✚ Materiales y conexionado: Utilizaremos la cámara AI, el kit inventor con el módulo Nezha y una micro: bit

En el siguiente [link](#) se muestra cómo construir una barra de parking. Nosotros sólo lo usaremos para montar la barrera, con el servomotor. En lugar del módulo de ultrasonidos, instalaremos nuestra cámara Smart AI Lens y nuestro proyecto se vería como se muestra en las siguientes imágenes: La cámara se conecta a un puerto I2C y el servomotor de 360 grados al primer conector de servomotores S1.



*Parte frontal: puerta parking*



*Parte trasera: puerta parking*

Ahora toca calibrar el servomotor. Me gustaría que la barrera en horizontal correspondiera con un ángulo de  $180^\circ$ . Para calibrarlo desmonto la barra del engranaje y subo el siguiente programa a la micro: bit, forzando de esta forma a que el servo se sitúe en esa posición:



Ya puedo introducir la barra. Ahora, la barra en horizontal equivaldrá a un ángulo de 180 grados.

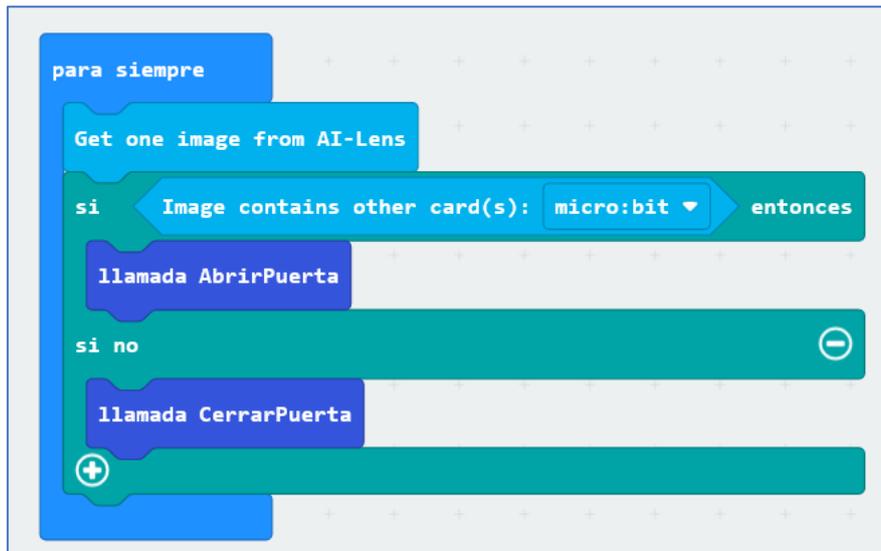
#### 🧩 Programación

Un posible script que controle la apertura y cierre de esta barra de parking a través de la detección de la tarjeta micro: bit V2 sería el siguiente:

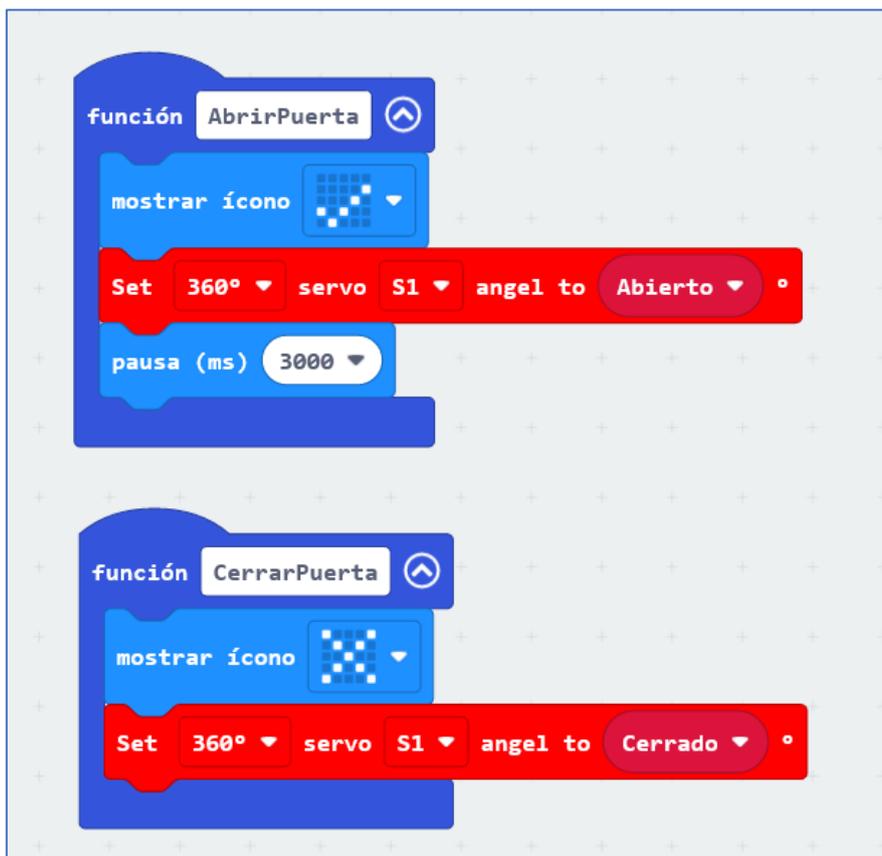
Para el bloque “iniciar”: Creo dos variables que serán ángulos y que llamo “Abierto” y “Cerrado”. El ángulo de apertura será 240° y el de cerrado de la barra de 180°. Ordenamos al servomotor que inicialmente se muestre la barra cerrada e inicializamos la cámara AI Lens con la función de reconocimiento de las tarjetas que vienen en la cámara



Para el bloque “por siempre”: Esperamos a que la cámara obtenga una imagen y, si esa imagen la reconoce (la tarjeta de la micro: bit), llamará a la función “AbrirPuerta”. En el caso de que no reconozca la imagen, llamará a la función “CerrarPuerta”.



Por último, cada subprograma “AbrirPuerta” y “CerrarPuerta” se definen ordenando al servomotor que gire al ángulo de las variables “Abierto” y “Cerrado”:



Sólo falta cargarlo en la micro: bit, comprobar que funciona y, obviamente, intentar mejorar el proyecto.

<https://jandalo.com/puerta-automatica/>