CEC Continuing Education Center

DesignNews

Exploring Smart AI Lens with the Micro:bit

DAY 3: Understanding Recognition Models

Sponsored by

DigiKey

informa markets

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Dr. Don Wilcher

Visit ‘Lecturer Profile’ in your console for more details.

LinkedIn Page:
https://www.linkedin.com/in/dr-don-wilcher-ed-d-mseit-ee-ceta-2735151/

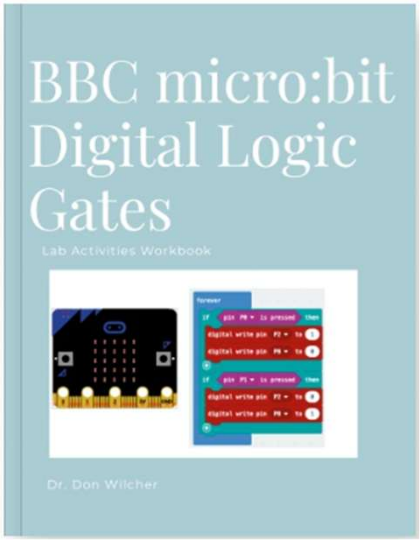
Patreon Page:
https://www.patreon.com/c/DrDon683

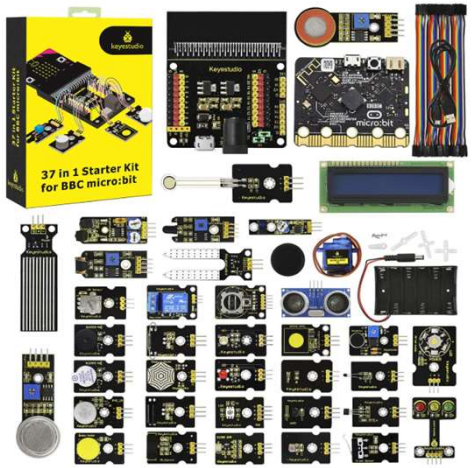
Course Kit and Materials

Micro:bit Version 2

Smart-AI Lens

BBC Micro:bit Digital Logic Gates Lab Activities Workbook





Keyestudio 37-In-1 Sensor Starter Kit

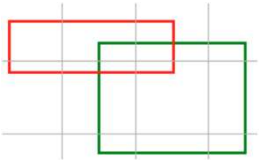
Research Perspective

"A single neural network predicts bounding boxes and class probabilities, directly from full images in one evaluation[1]."

Agenda:

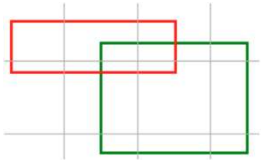
- What Is a Recognition Model?
- Types Of Recognition Models
- MicroPython: A Mini Tools Primer
- Lab: Characteristic Learn Device

What Is a Recognition Model



- A Recognition Model is an artificial intelligence or machine learning model designed to
  - a) identify
  - b) classify or
  - c) verify objects
  - d) patterns
  - e) or signals in data.
- Unlike detection models (which just find and locate things), recognition models focus on understanding what those things are.

What Is a Recognition Model. . .

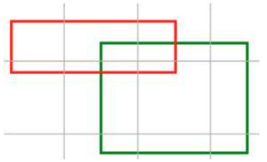


A recognition model maps input data (such as an image, sound, or sensor signal) to a known category, label, or identity.

Mathematically: $f(x) \rightarrow y$

Where:
x= input data (image, voice sample, fingerprint, etc.)
f= recognition model
y= recognized label (e.g., “cat,” “Alice’s face,” “digit 7”)

What Is a Recognition Model. . .



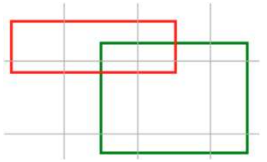
The probability of the image being recognized uses the following Softmax equation.

$$P(y = k|x) = \frac{e^{Zk}}{\sum_{j=1}^{K} e^{Zj}}$$

where:
- z_k =score (logit) for class k from the network
- K= number of classes

Softmax is a mathematical function used in neural networks to convert a vector of real numbers into a probability distribution.

What Is a Recognition Model. . .



A Python model can be used to illustrate the recognition model aided by the Softmax equation.

Classes = **x =[apple, banana, grape, orange]**

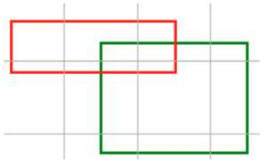
The Python model will determine that the image is a “grape” based on the Softmax equation’s probability score.

Question 1

For a recognition model, what types are used?

a) image, voice sample, fingerprints
b) voice sample, keyboard, mouse
c) tablet, image, voice sample
d) none of the above

What Is a Recognition Model. . .



Classes

Simulated scores

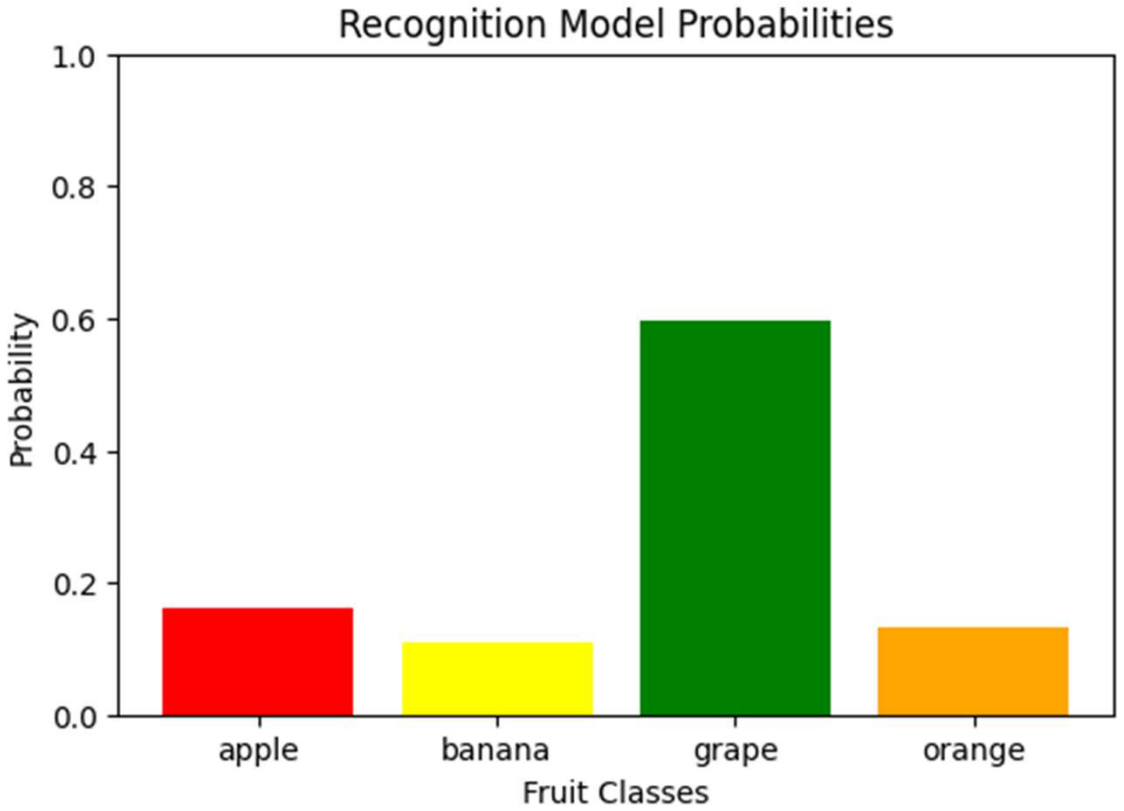
Softmax Eq.

```python
import numpy as np
import matplotlib.pyplot as plt

# --- Step 1: Define fruit classes ---
classes = ["apple", "banana", "grape", "orange"]

# The target image is a grape
true_label = "grape"

# --- Step 2: Create simulated logits (scores) for each fruit ---
# Higher score for grape to simulate recognition
z = np.array([1.2, 0.8, 2.5, 1.0])  # raw model scores (logits)

# --- Step 3: Apply Softmax to get probabilities ---
def softmax(z):
    exp_z = np.exp(z - np.max(z))  # stability trick
    return exp_z / np.sum(exp_z)

probs = softmax(z)
```

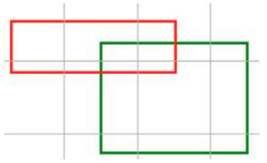
Python
Partial code

What Is a Recognition Model. . .

```
Class Probabilities (P(y=k|x)):
  apple: 0.162
  banana: 0.109
  grape: 0.596
  orange: 0.133

Predicted Class (ŷ): grape
```



What Is a Recognition Model. . .



Why are Recognition Models important?

Recognition models are important because they give machines the ability:

a) to *know what something is*, not just that it exists.

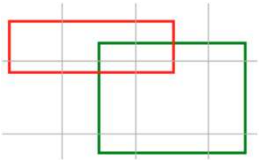
b) They provide

i. classification

ii. identity, and meaning,

c) These elements are critical for security, automation, personalization, and human-like interaction.

Types of Recognition Models

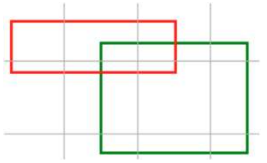


1. Image Recognition Models
   - **What they do:** Identify and classify objects, scenes, or patterns within images.
   - **Examples:**
     a) Recognizing whether an image contains a cat, dog, or car.
     b) Identifying a tumor in a medical scan.
   - **Mathematical core:** Softmax-based classifiers over extracted features:

$$\hat{y} = \arg\max_{k} * P(y = k \mid x)$$

   - **Algorithms/architectures:** Convolutional Neural Networks (CNNs), Vision Transformers (ViT).

Types of Recognition Models. . .



2. Object/Face Recognition Models

•What they do: Detect and identify specific instances of objects or people.

•Examples:

a) Face ID unlocking a smartphone (recognizes *you*).

b) Recognizing a product by its brand logo.

•Two parts:

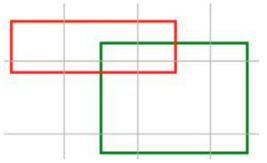
Feature extraction (embeddings from CNNs/ViT).

- Similarity measure (cosine similarity, Euclidean distance).

•Mathematics:

$$\text{similarity } (f, (x_1) f(x_2)) > \textit{threshold} \quad \Rightarrow \quad \textit{same identity}$$

Types of Recognition Models. . .



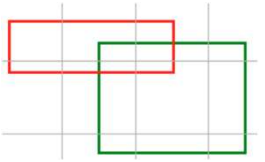
3. Speech Recognition Models

- What they do: Convert spoken language into text.
- Examples:
  - a) Voice assistants (Alexa, Siri, Google Assistant).
  - b) Automated transcription.
- Pipeline:
  - a) Audio waveform → feature extraction (MFCCs, spectrograms).
  - b) Sequence model (RNN, Transformer).
  - c) Probability distribution over vocabulary words.
- Mathematics (sequence recognition):

$$\hat{y} = \arg\max_{y} * P(y \mid x_{1:T})$$

where $x_{1:T}$ is the audio sequence.

Types of Recognition Models. . .



4. Text Recognition Models (OCR)

•What they do: Recognize printed or handwritten characters from images.(Associated with Modified National Institute of Standards and Technology (MNIST).

•Examples:

a) License plate recognition.

b) Reading handwritten digits (MNIST).

•Pipeline:

a) Image → segmentation → character recognition.

b) Often CNN + RNN + CTC loss for sequence alignment.

Question 2

Which equation is true for a Softmax-based classifier?

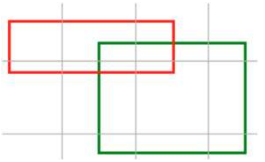
a) $\hat{y} = \arg\max_{k} * P(y = k \mid x)$

b) $\hat{y} = \arg\max_{k} * P(\mathrm{z} = k \mid x)$

c) $y = \arg\max_{k} * P(y = k \mid x)$

d) none of the above

Types of Recognition Models. . .



Other Types of Recognition Models include:

- Pattern/Character Recognition – Reads structured data like handwriting or license plates.

  Example: OCR (Optical Character Recognition).

- Speech Recognition – Converts spoken words into text.

  Example: Siri or Alexa understanding a command.

- Biometric Recognition – Identifies humans by unique features (iris, fingerprints, gait).

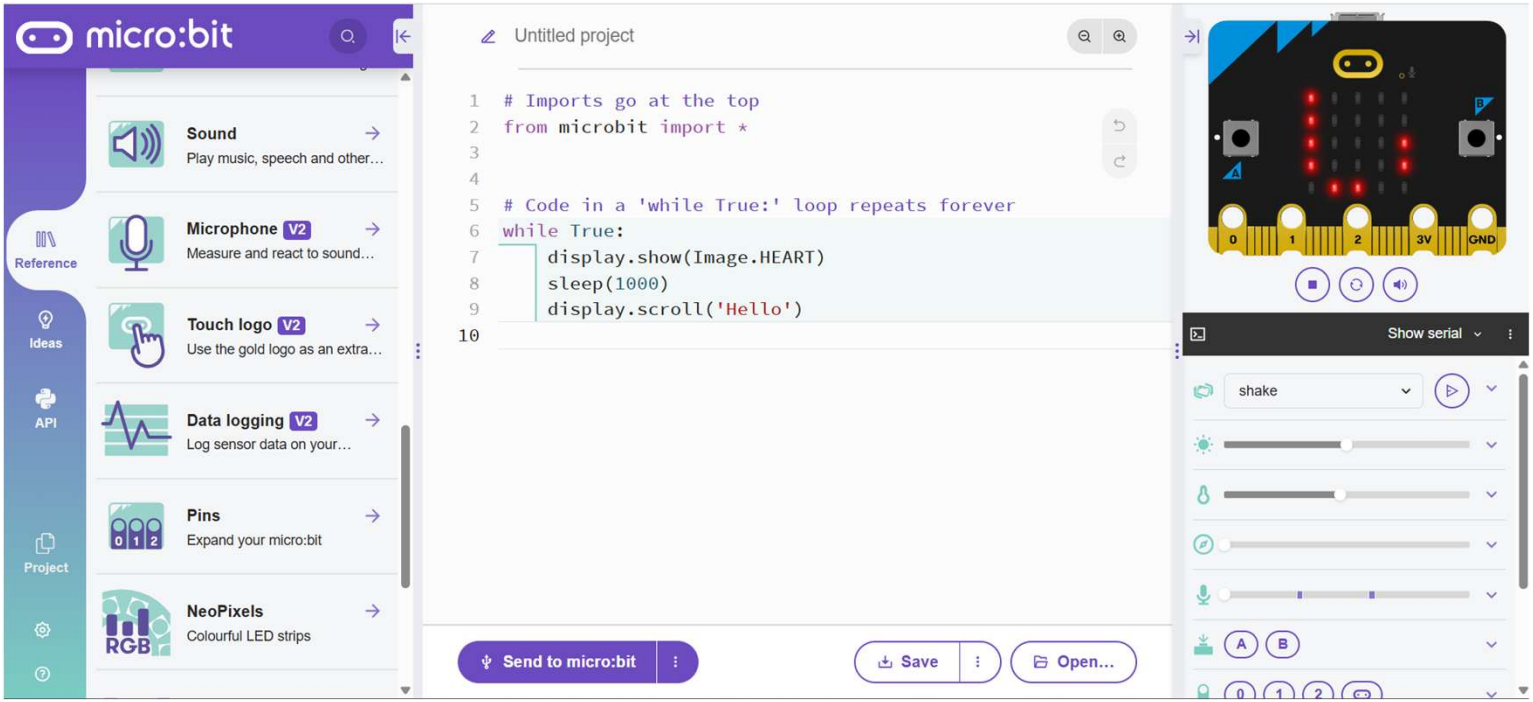
  Example: Border control systems verifying passports.

MicroPython: A Mini Tools Primer

https://python.microbit.org/v/3

Micro:bit.org Editor

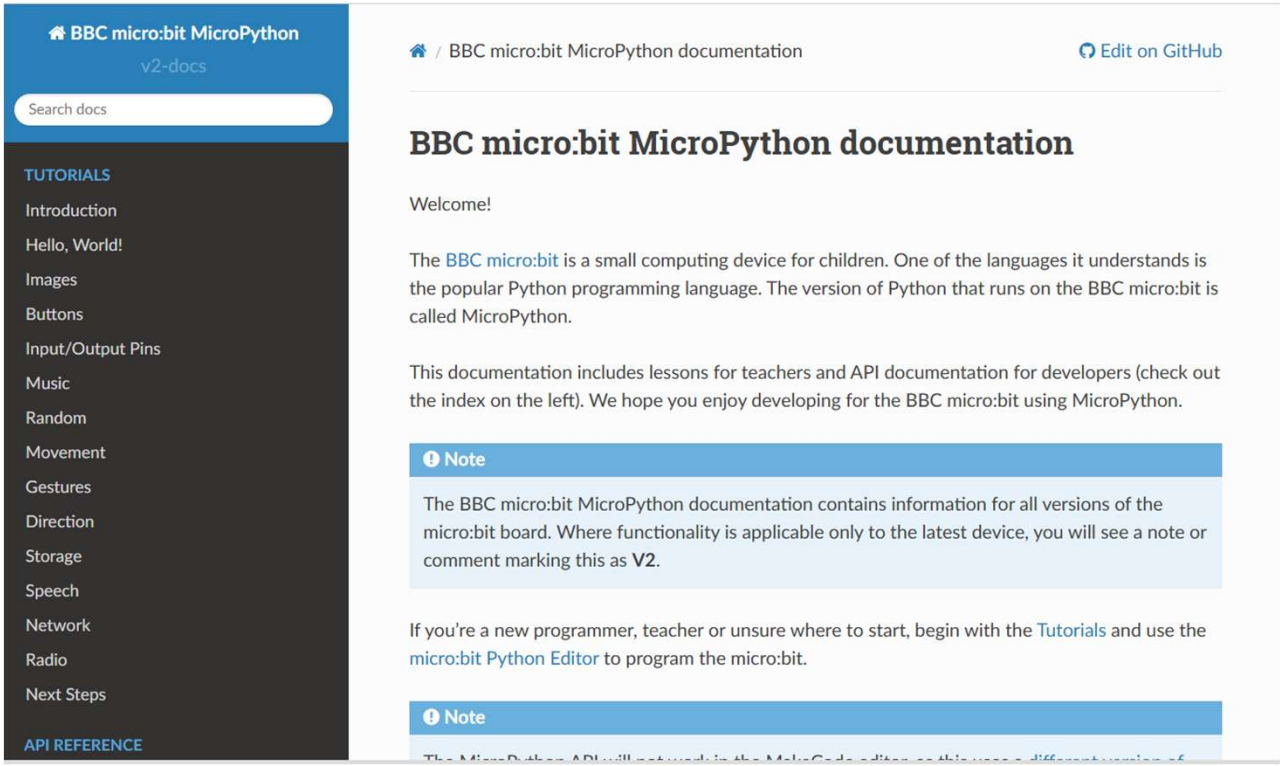
BBC Micro:bit Development Tools (Nordic Semiconductor nRF SoC)



MicroPython: A Mini Tools Primer

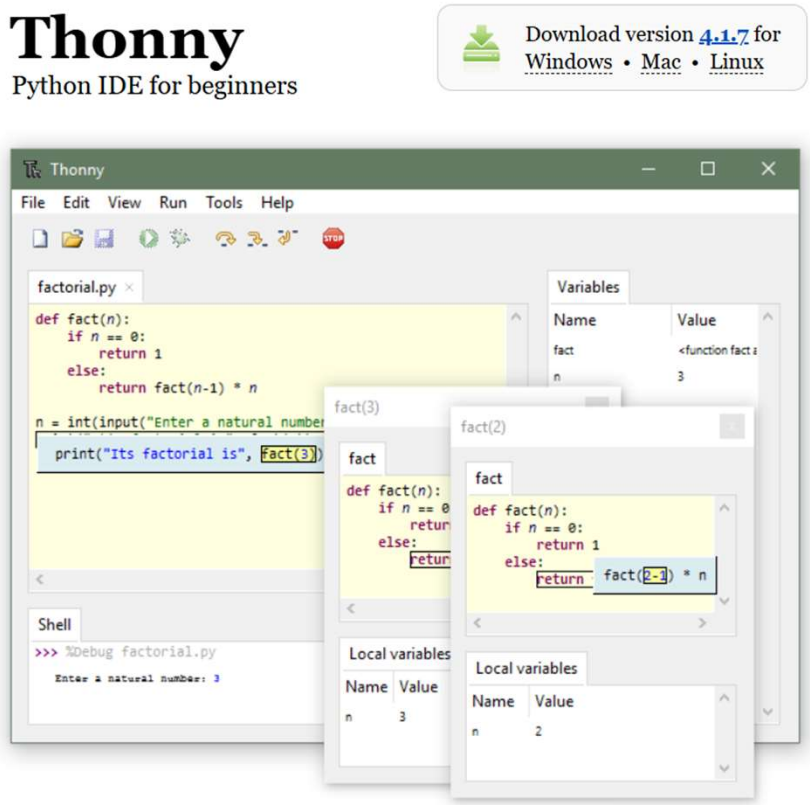
https://microbit-micropython.readthedocs.io/en/v2-docs/index.html

BBC Micro:bit Development Tools (Nordic Semiconductor nRF SoC)



MicroPython: A Mini Tools Primer

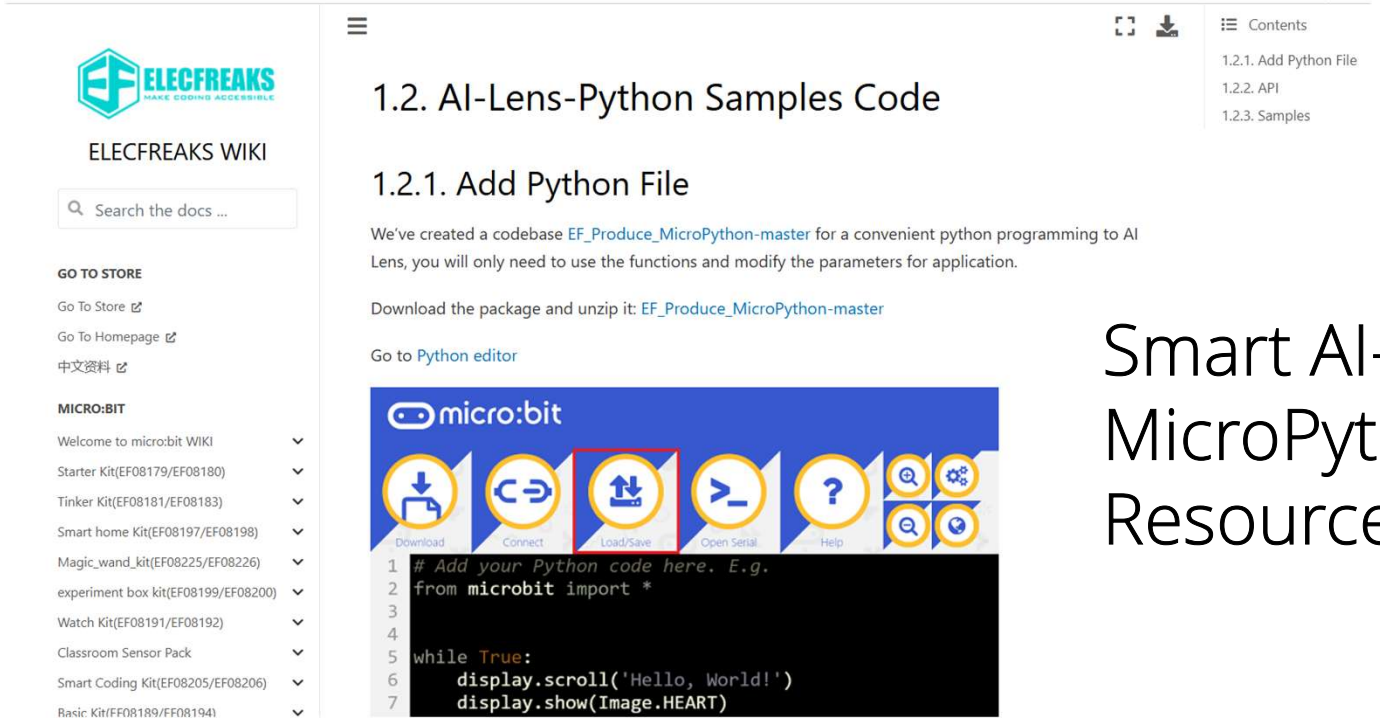
https://thonny.org/



BBC Micro:bit Development Tools (Nordic Semiconductor nRF SoC)

MicroPython: A Mini Tools Primer

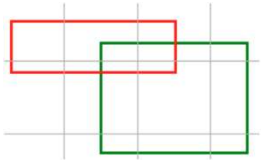
BBC Micro:bit Development Tools (Nordic Semiconductor nRF SoC)



Smart AI-Lens MicroPython Resources

https://www.elecfreaks.com/learn-en/microbitplanetX/ai/Plant-X-EF05035-python-en.html

MicroPython: A Mini Tools Primer



BBC Micro:bit Development Tools (Nordic Semiconductor nRF SoC)

1.2.2. API

API	Description
AILENS()	Init AI Lens
switch_function(func)	Choose AI Lens functions, funcchoose function: (Learncharacteristics learn; Cardcard recognition; Faceface recognition; Trackingtracking recognition; Colorcolor recognition; Ballball recognition)
get_card_content()	Recognize the contents on the card
get_card_data()	Get the info of the cards from the AI Lens to a list(X-axis; Y-axis; Width; Height; Confidence coefficient; Total numbers of cards; ID of current cards)
get_face()	Judge if there is a human face recognized in the AI Lens
get_face_data()	Get the info of the face(s) from the AI Lens to a list(X-axis; Y-axis; Width; Height; Confidence coefficient; Total numbers of face(s); ID of current face(s))
get_ball_color()	Recognize the color of the balls in the Ai Lens
get_ball_data()	Get the info of the ball(s) from the AI Lens to a list(X-axis; Y-axis; Width; Height; Confidence coefficient; Total numbers of ball(s); ID of current ball(s)).
get_track_data()	Get the info of the segment(s) from the AI Lens to a list (Deviation angel, deviation distance, segment length)
get_color_type()	Recognize the color of the cards in the AI Lens
	Get the info of the color(s) from the AI Lens to a list(X-axis; Y-axis; Width;

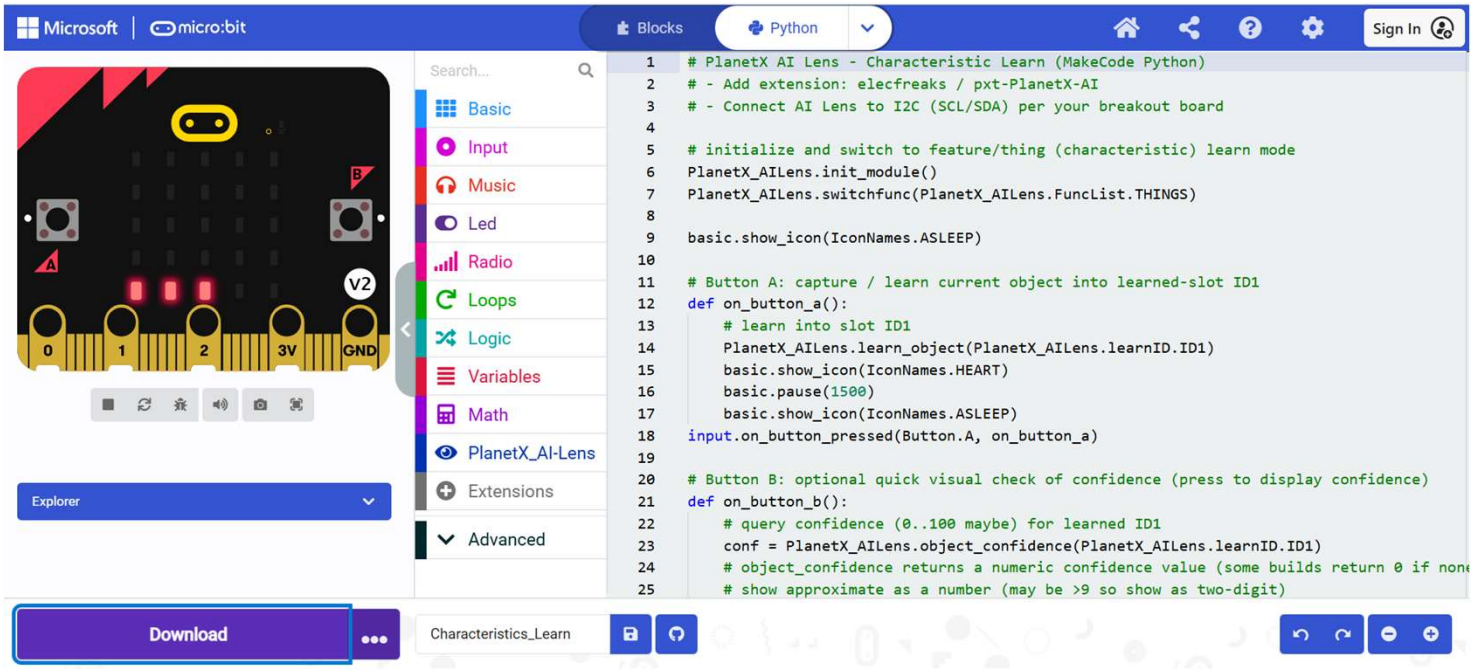
Smart AI-Lens MicroPython API: Partial List

https://www.elecfreaks.com/learn-en/microbitplanetX/ai/Plant-X-EF05035-python-en.html

MicroPython: A Mini Tools Primer

Microsoft MakeCode's version of MicroPython

BBC Micro:bit Development Tools (Nordic Semiconductor nRF SoC)

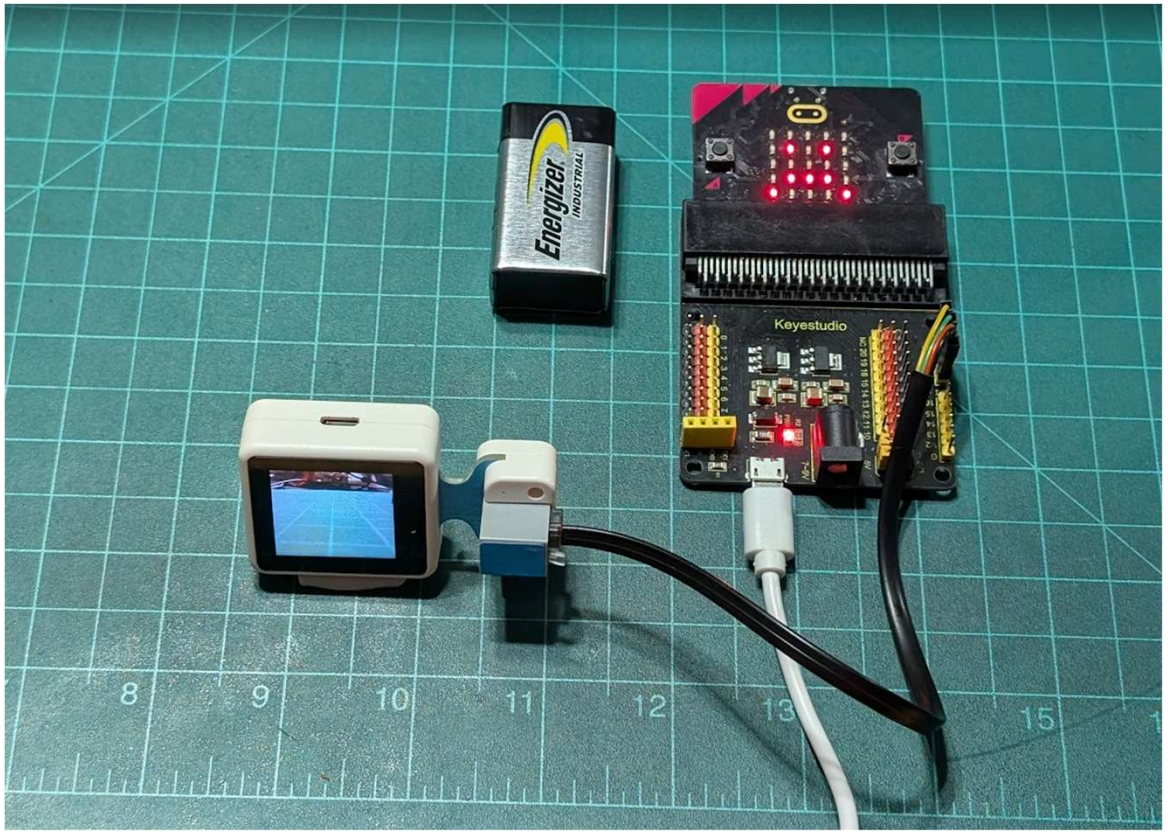


Question 3

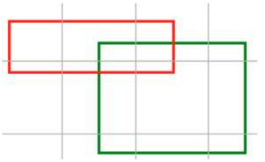
In reviewing slide 25, which API "Get the info of the cards from the AILens to a list?

a) switch_function(func)
b) get_card_content()
c) get_card_data()
d) none of the above

Lab: Characteristics Learn Device



Lab: Characteristics Learn Device. . .

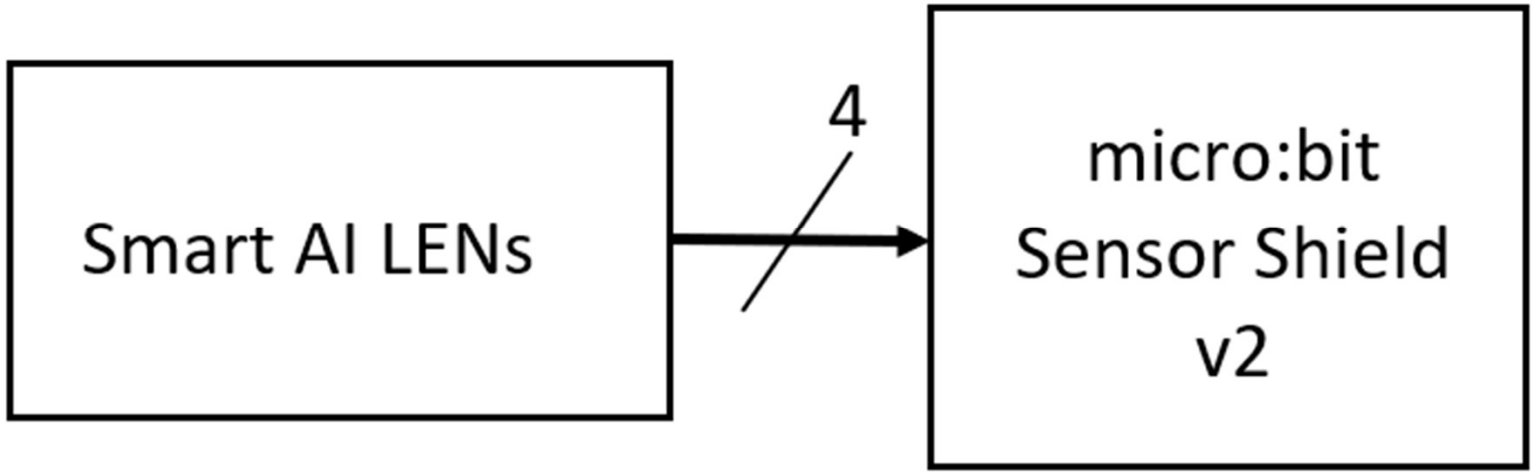


Participant Learning Objectives:

- Participants will learn how to use the Microsoft MakeCode Python to build the Characteristics Learn Device.
- Participants will learn to implement a learning event on recognizing an object using the Smart AI-Lens to Micro:bit Sensor Shield.
- Participants will learn to view the Confidence Level on the Characteristic Learn Device.

Lab: Characteristics Learn Device. . .

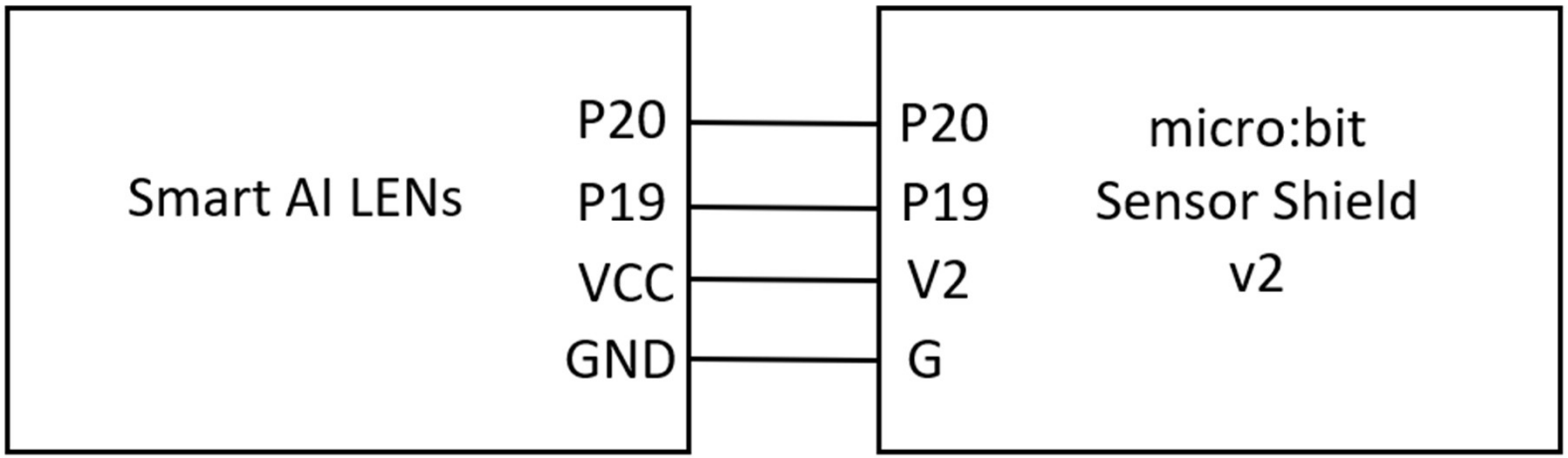
Concept Block Diagram for a Characteristics Learn Demonstrator



Lab: Characteristics Learn Device

Characteristics Learn Device Demonstrator Electrical Wiring Diagram

Smart AI-Lens To Micro:bit: Electrical Wiring Diagram

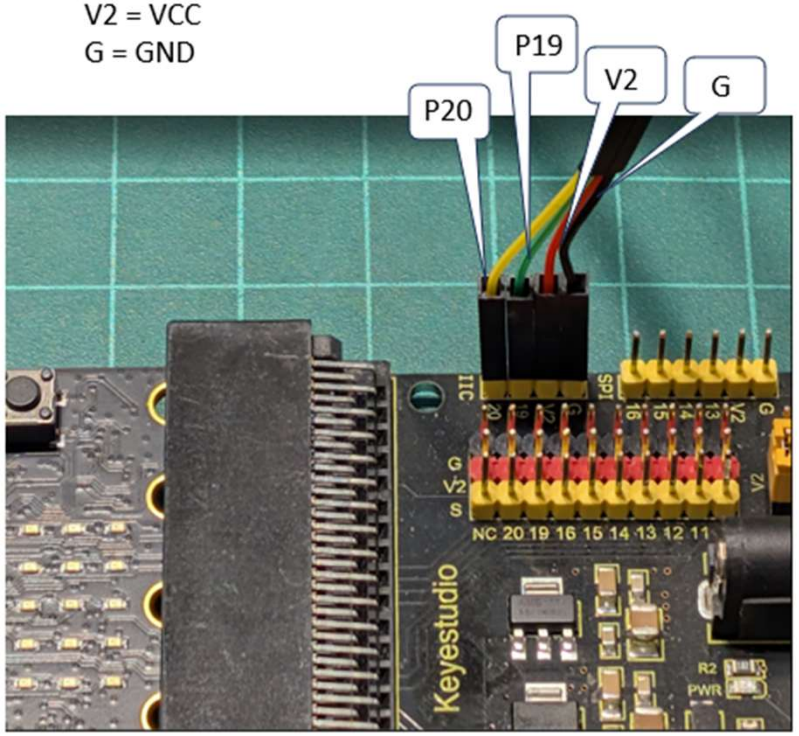


Lab: Characteristics Learn Device

Characteristic Learn Device Demonstrator Electrical Wiring Diagram

Note:
V2 = VCC
G = GND

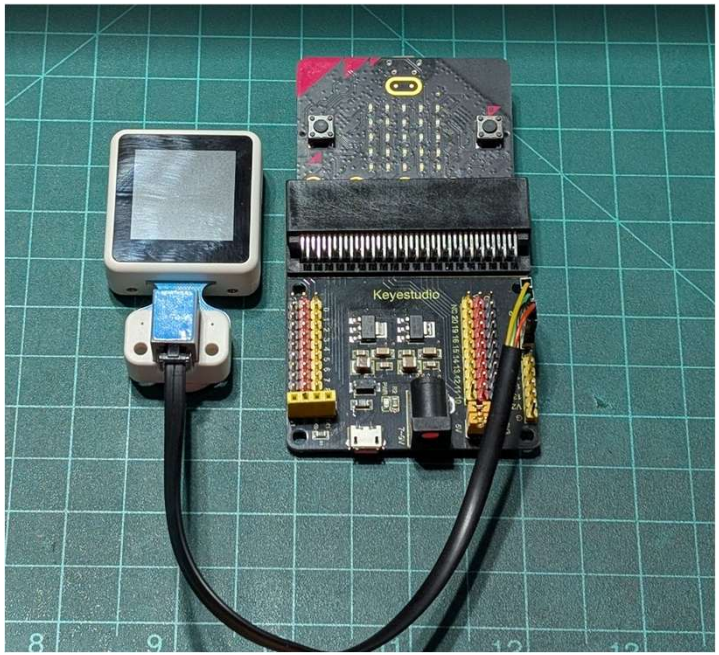
RJ11 wire harness connected to Micro:bit Sensor Shield



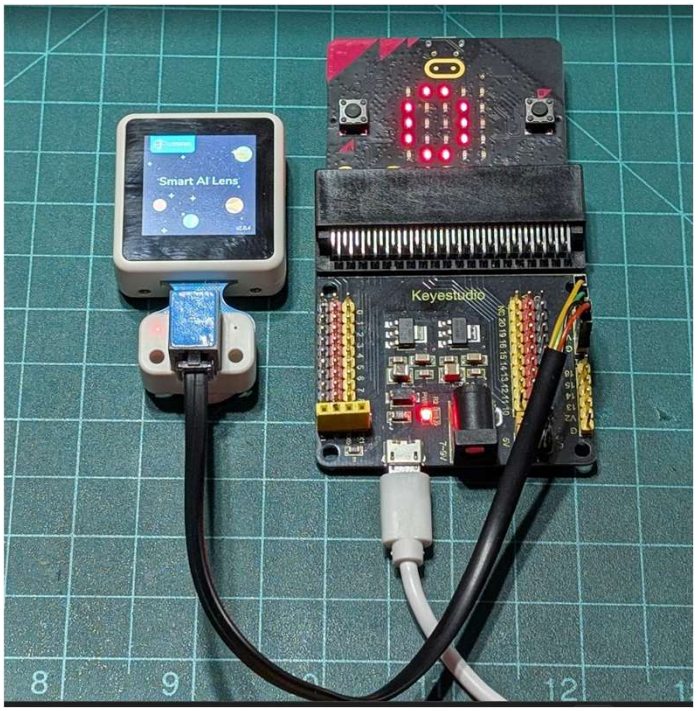
Lab: Characteristics Learn Device. . .

Characteristic Learn Device Demonstrator Electrical Wiring Diagram

Smart AI-Lens wired to the Micro:bit Sensor Shield

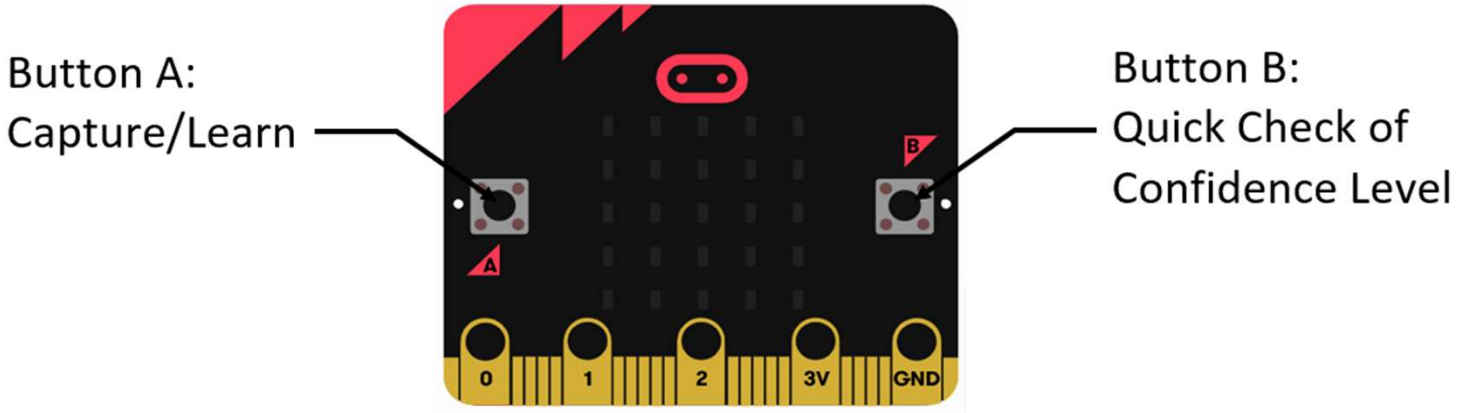


Smart AI-Lens powered ON



Lab: Characteristics Learn Device. . .

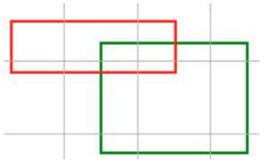
User Interface (UI) of Characteristic Learn Device



Operation:

A press/release on Button A, will allow the Smart-AI Lens to capture and store an object's image. Placing the object in front of the Smart-AI Lens will allow the micro:bit to display a Smiley face. No object present or the incorrect object will display a Sad face on the micro:bit's 5x5 LED matrix.

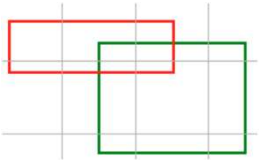
Lab: Characteristics Learn Device. . .



Setup Procedure

1. Go to Microsoft MakeCode: micro:bit and open a new project.
2. Click on the Python button to open a new editor.
3. Copy and paste the Characteristics Learn Python code from the text file into the editor.
4. Add the PlanetX-AI extension in the Make Code Python project before compiling. Note: The extension provides the PlanetX-AI Lens API to the programming environment.
5. Attach the micro:bit to the code development laptop computer or desktop PC USB port.
6. Download the Characteristics Learn code to the micro:bit (Click the Download button)
7. Insert the micro:bit into the Sensor Shield. Attach the USB cable to the Sensor Shield's USB connector.

Lab: Characteristics Learn Device. . .

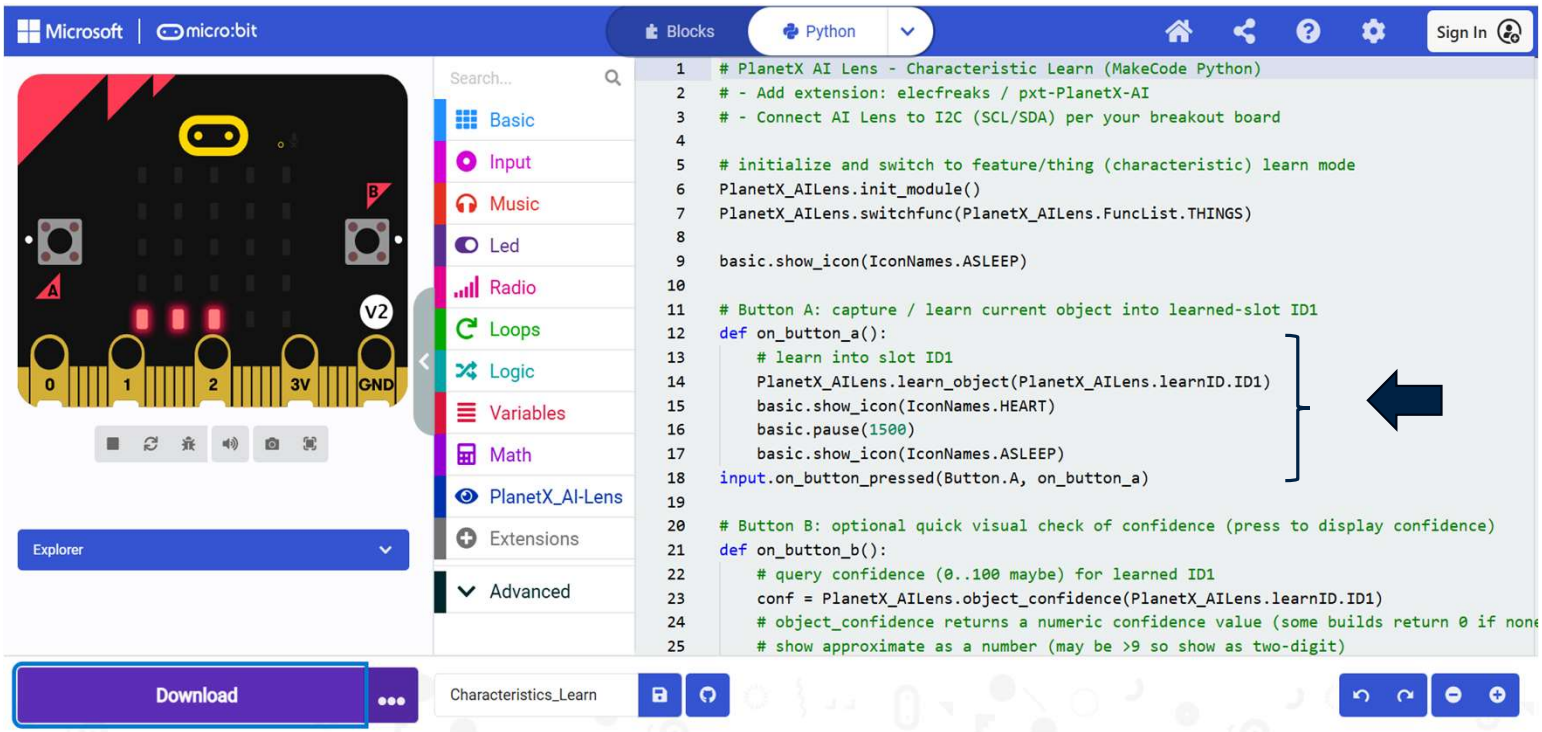


Learn Mode

1. Place an object in front of the Smart AI-Lens.
2.Press/release Button A. A heart will be displayed on the micro:bit's 5x5 LED display.
   Note: the object is stored as an image on the micro:bit.
2. Press Button B. The number 79 will scroll on the LED matrix. This number is the confidence level..
4. Place the learned object from Step 1 in front of the Smart AI-Lens.
5. A smiley face will be displayed on the micro:bit's LED matrix.

Lab: Characteristics Learn Device. . .

Characteristics Learn Python Code pasted onto the editor



Learn Mode – Step 3

Question 4

In reviewing slide 25, which API aligns with the description "Get the info of the cards from the AI Lens to a list?

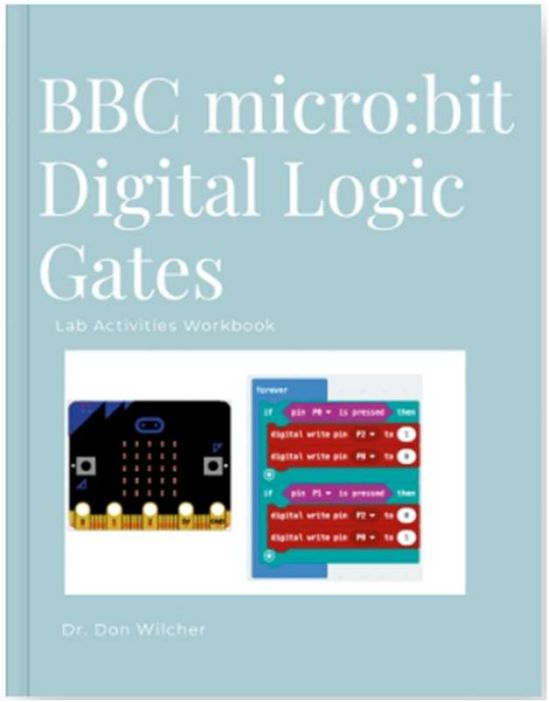
a) RJ15
b) RJ45
c) RJ11
d) none of the above

Lab: Characteristics Learn Device. . .

BBC Micro:bit Digital Logic Gate Lab Activities Workbook Challenge

Modify the Characteristics Learn Device code where pressing a digital pushbutton switch will turn on an external red LED.

Page 34 provides an electrical wiring diagram on attaching the traffic light unit and the digital pushbutton switch to the sensor shield.

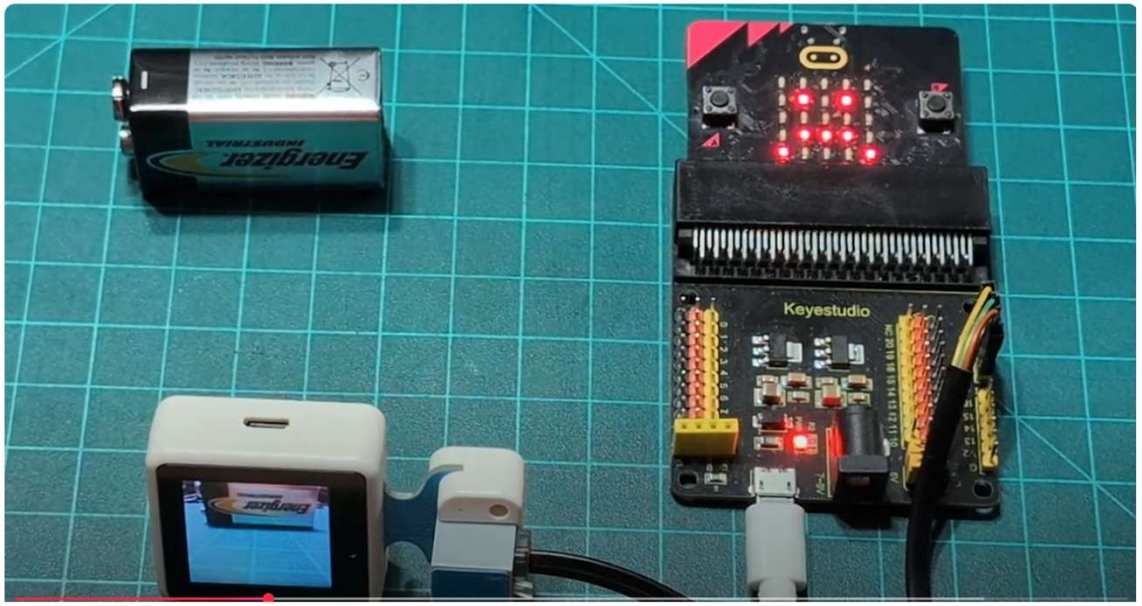


Lab: Characteristics Learn Device. . .

- Assembled and Functionally Characteristics Learn Device Demonstrator

Watch the Video Clip!

https://youtu.be/xpPdI8WaAo4



Question 5

Which page number in the BBC Digital Logic Gate workbook provides an electrical wiring diagram for attaching the digital pushbutton switch and the traffic light unit to the sensor shield?

a) 26
b) 39
c) 35
d) 34

Thank you for attending

Please consider the resources below:

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *arXiv*:1506.02640 [cs.CV], Jun. 2016. [Online]. Available: https://arxiv.org/abs/1506.02640
[2] [1] D. Wilcher, “Designs News September 25 webinar code,” GitHub repository, Sep. 2025. [Online]. Available: https://github.com/DWilcher/DesignNews-WebinarCode/blob/main/September_25_Webinar_Code.zip

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

DigiKey

informa markets