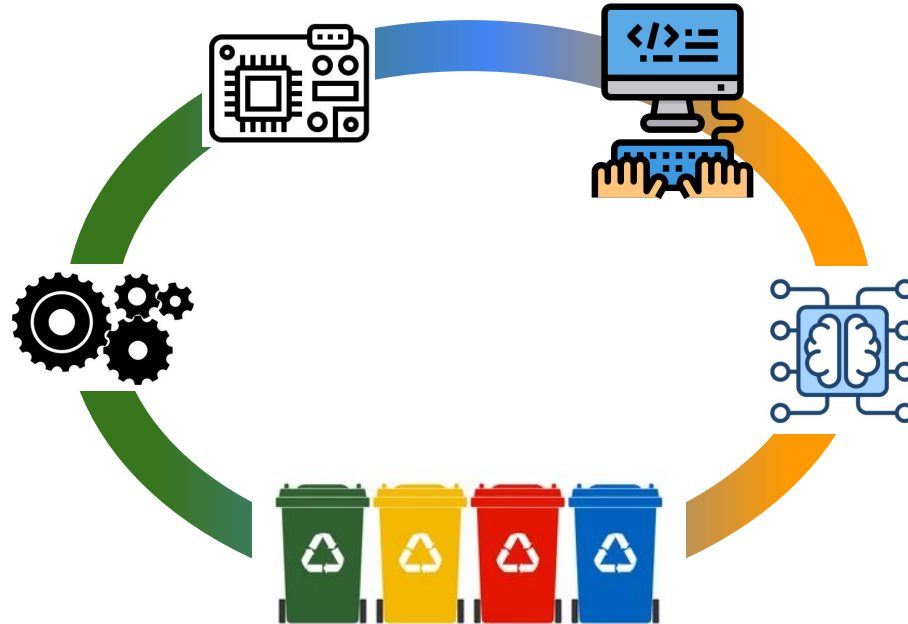

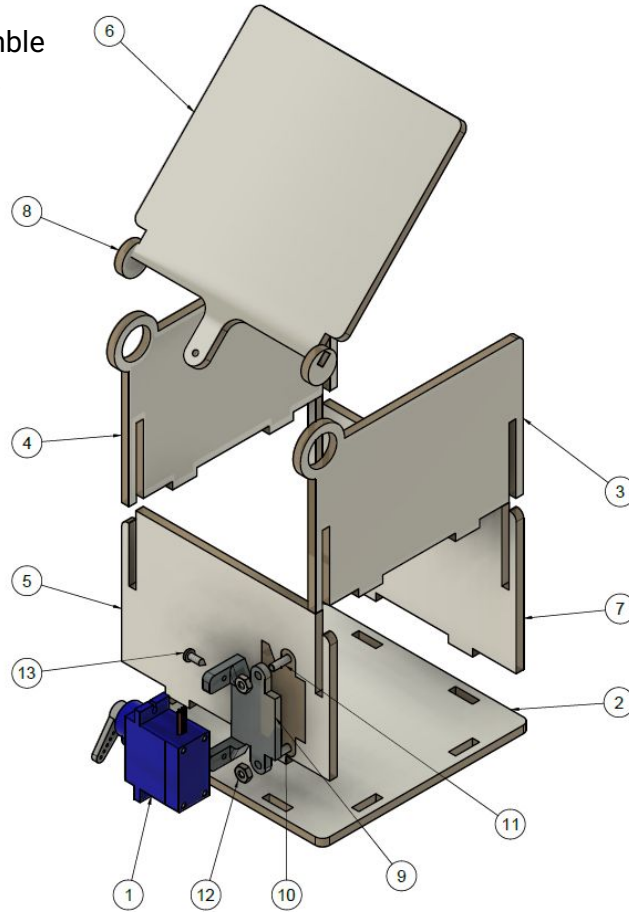


Smart Recycling System: Building and AI-powered bin

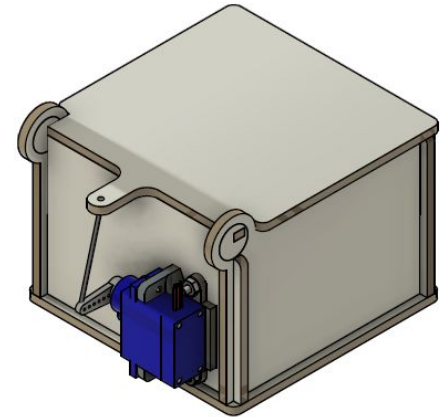


ASSEMBLING THE BIN AND CONNECTING ALL THE COMPONENTS


 **TASK 1:** Follow the drawing to assemble the box and attach the servo motor at the back

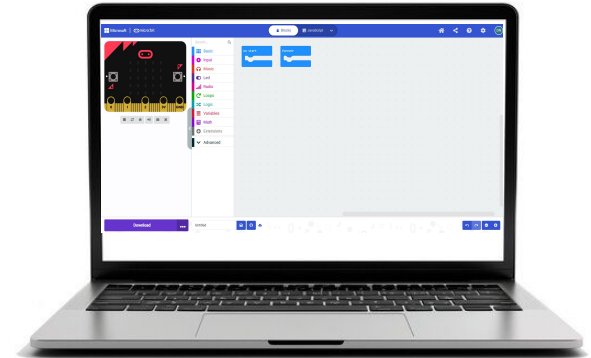
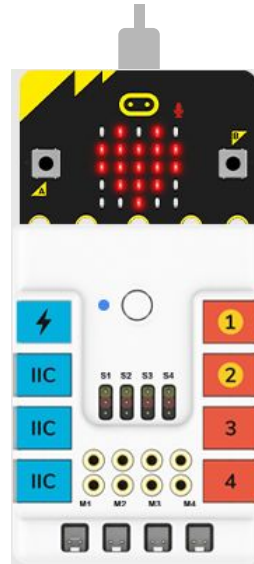
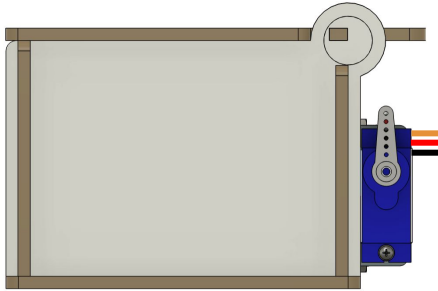


Parts List			
Item	Qty	Part Number	Material
1	1	9g Micro Servo SG90	Plastic
2	1	Base	MDF Medium Density Fiberboard
3	1	Side	MDF Medium Density Fiberboard
4	1	Side (1)	MDF Medium Density Fiberboard
5	1	Back	MDF Medium Density Fiberboard
6	1	Lid	MDF Medium Density Fiberboard
7	1	Front	MDF Medium Density Fiberboard
8	2	Hinge	MDF Medium Density Fiberboard
9	1	Mini_Servo_Bracket	Plastic
10	1	DIN 7985 - M2.5x8-H H Steel 4.6 Plain	Steel 4.6, Plain
11	1	DIN 7985 - M2.5x10-H H Steel 4.6 Plain	Steel 4.6, Plain
12	2	DIN 934 - M2.5 x 0.45 Steel 6 Plain	Steel 6, Plain
13	2	DIN EN ISO 7049 - ST2.2 x 6.5 - C - H Steel 4.6 Plain	Steel 4.6, Plain




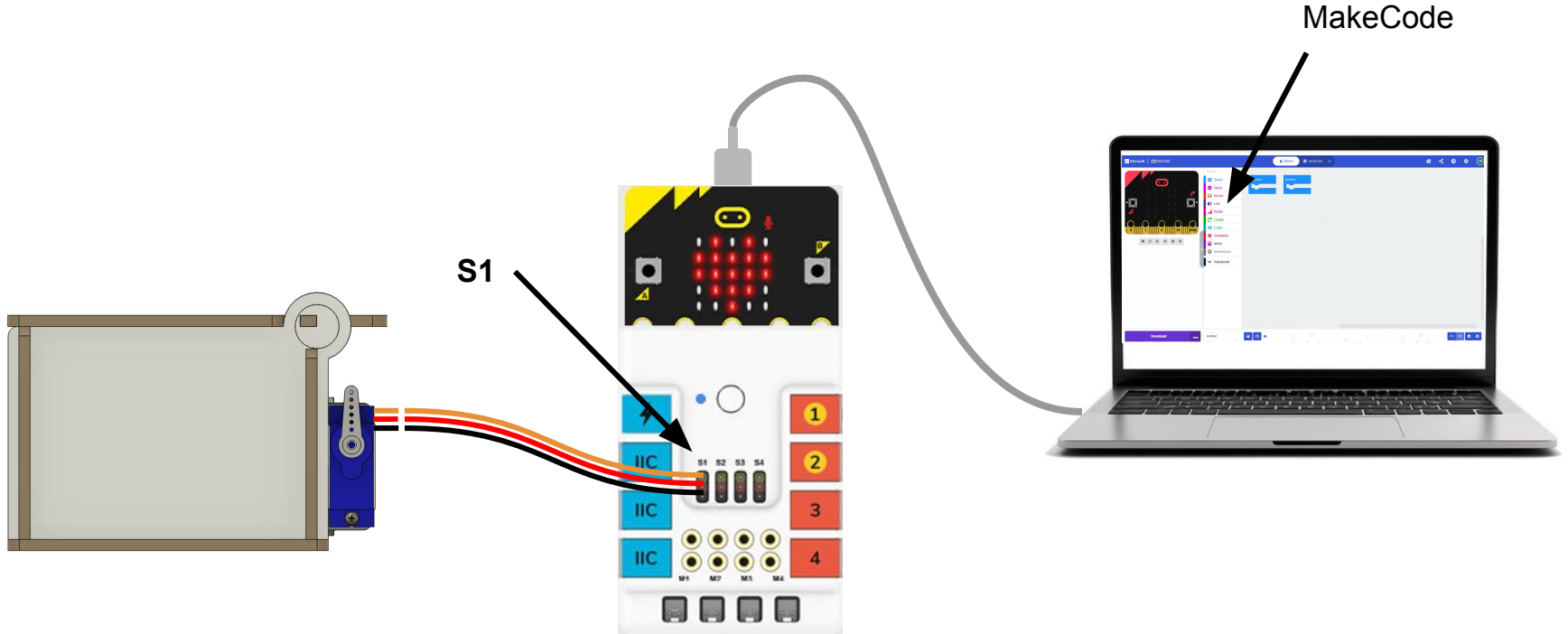
ASSEMBLING THE BIN AND CONNECTING ALL THE COMPONENTS

 **TASK 2:** Follow the teacher's instructions on how to connect the servo to the expansion board and draw the correct connections on the image below.

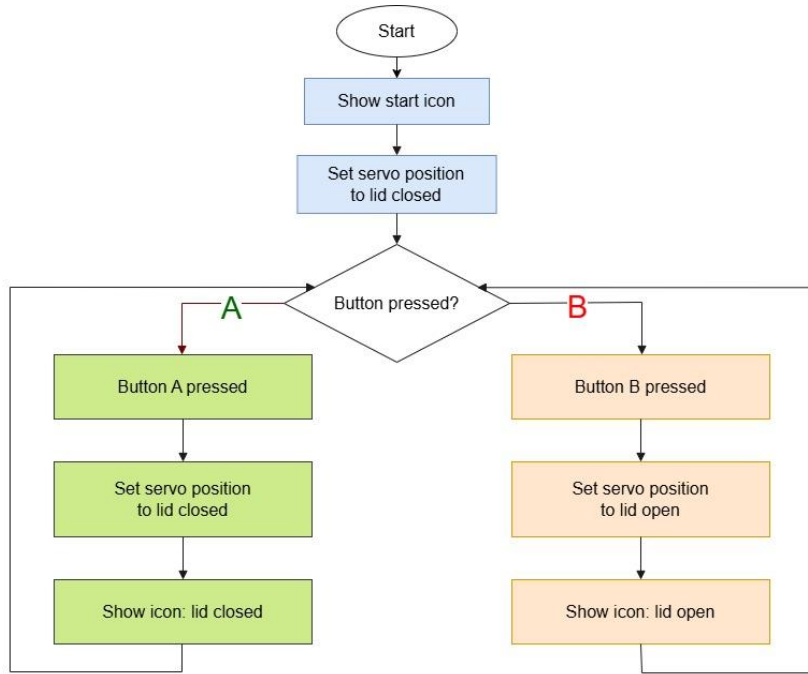



ASSEMBLING THE BIN AND CONNECTING ALL THE COMPONENTS

 **TASK 2:** Follow the teacher's instructions on how to connect the servo to the expansion board and draw the correct connections on the image below.



PROGRAMMING THE MANUAL OPENING OF THE BIN



 **TASK 3:** Based on the flowchart, complete the following explanation of what the code is doing:

On **start**:

- Show an _____.
- Set servo position to lid _____

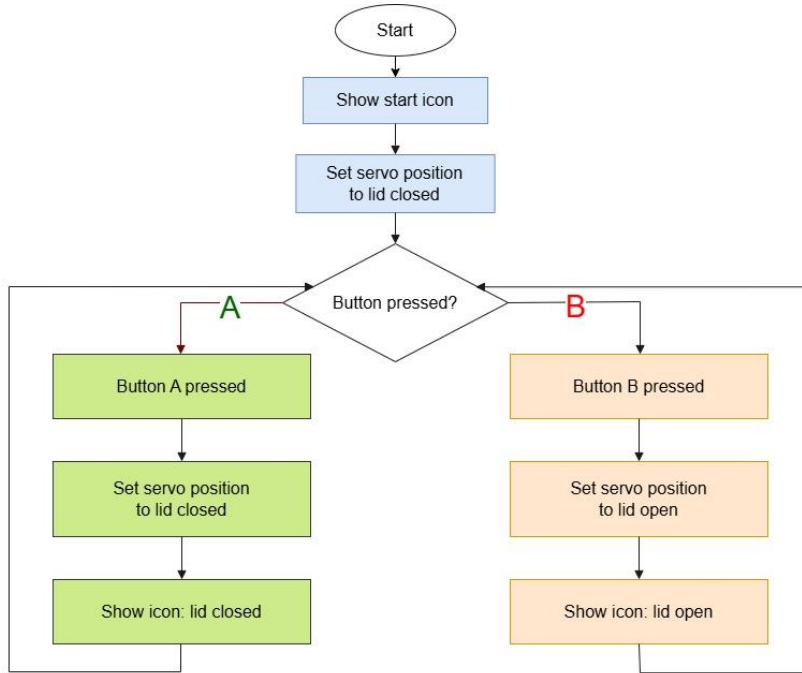
On **button _____ pressed**:

- set _____ position to _____
- Show an _____ that represents _____

On **button _____ pressed**:

- Set _____ to _____
- Show _____

PROGRAMMING THE MANUAL OPENING OF THE BIN



On **start**:

- Show an start icon
- Set servo position to lid closed

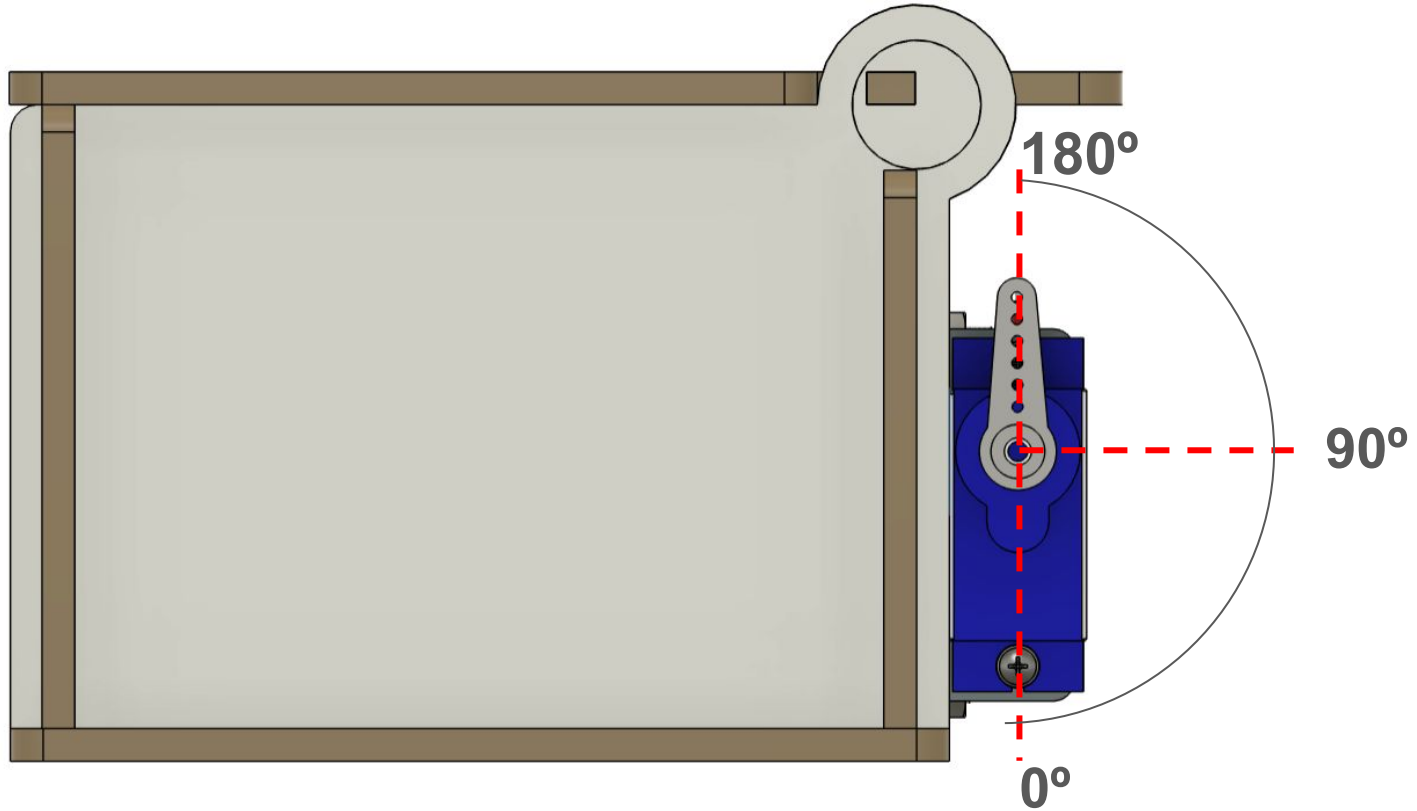
On **button A pressed**:

- set servo position to lid closed
- Show an icon that represents the lid closed

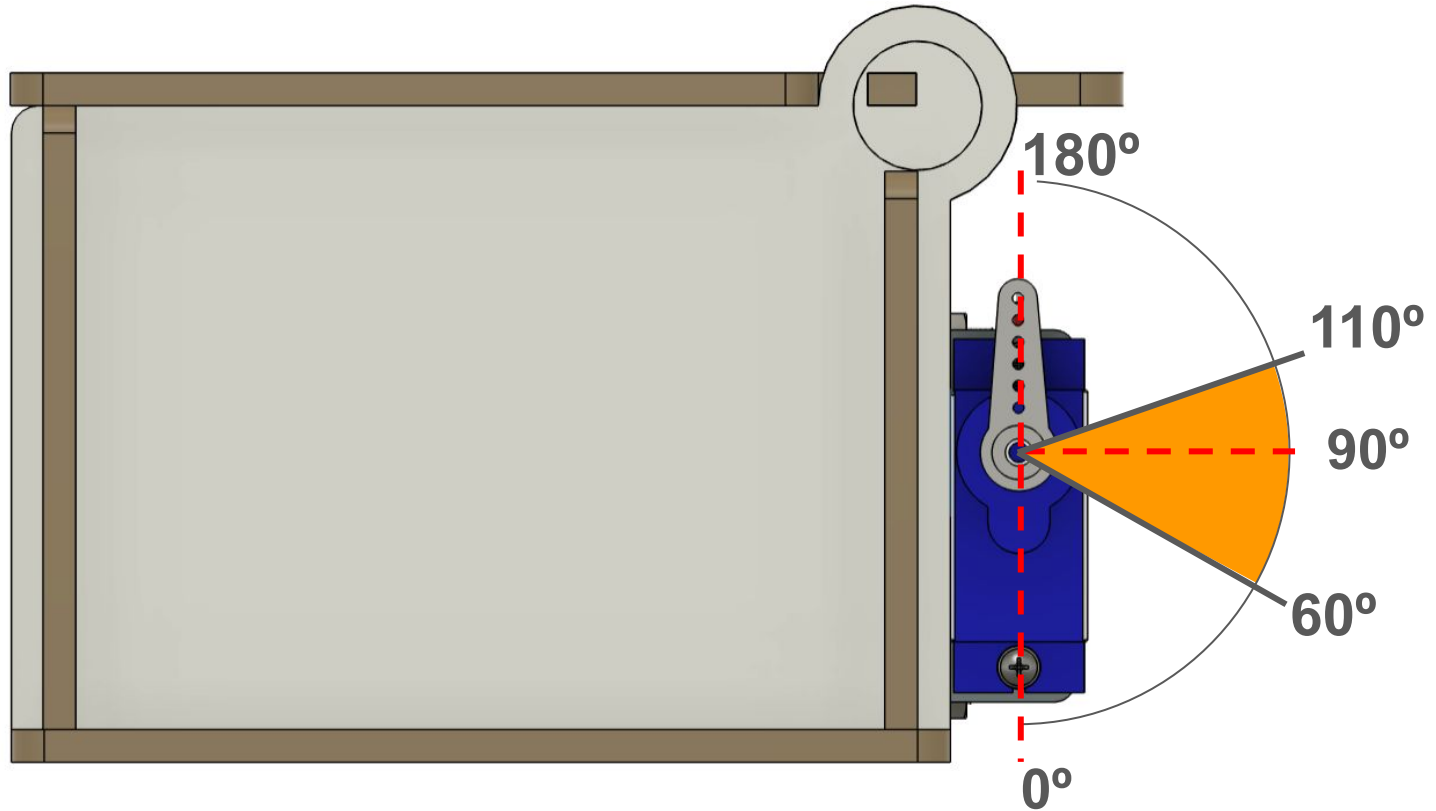
On **button B pressed**:

- Set servo to lid open
- Show an icon that represents the lid open

PROGRAMMING THE MANUAL OPENING OF THE BIN



PROGRAMMING THE MANUAL OPENING OF THE BIN



PROGRAMMING THE MANUAL OPENING OF THE BIN


The screenshot shows the Microsoft MakeCode editor interface for a micro:bit. The top navigation bar includes the Microsoft logo, 'micro:bit', and tabs for 'Blocks' and 'Python'. On the left, a sidebar lists various categories: Basic, Input, Music, Led, Radio, Servos, Loops, Logic, Variables (highlighted), Math, NeZha, Extensions, and Advanced. A red arrow points from the 'Variables' category in the sidebar to the 'Variables' panel in the center. The 'Variables' panel displays 'Make a Variable...' and 'Your Variables' with two variables: 'lid_closed' and 'lid_open'. The main workspace shows a script starting with an 'on start' block, followed by 'show icon' (LED matrix), 'set lid_closed to 110', 'set lid_open to 60', and 'Set 180 degrees servo S1 angle to lid_closed'.

The screenshot shows the 'Extensions' page in the MakeCode editor. It features a search bar with the text 'necha'. Below the search bar are several category buttons: 'Lights and Display', 'Software', 'Science', 'Robotics', 'Gaming', and 'Networking'. The main area displays a list of extension cards. Two cards are visible: 'necha2' and 'necha'. The 'necha' card is highlighted with a red box. The 'necha' card description reads: 'necha (NEB) The micro:bit extension board Necha with I2C connection port by ELECTREACKS Co Ltd'. There is also an 'Import File' button on the right side of the page.


PROGRAMMING THE MANUAL OPENING OF THE BIN

```
on start
  show icon [LED Matrix]
  set lid_closed to 110
  set lid_open to 60
  Set 180° servo S1 angle to lid_closed °
```


```
on button A pressed
  Set 180° servo S1 angle to lid_closed °
  show leds
```

A 5x5 grid of 25 blue squares, representing all LEDs being lit.

```
on button B pressed
  Set 180° servo S1 angle to lid_open °
  show leds
```

A 5x5 grid of 25 squares. The top row has 4 blue squares and 1 white square. The second row has 2 blue squares, 1 white square, and 2 blue squares. The third row has 3 blue squares, 1 white square, and 1 blue square. The bottom two rows have 5 white squares each.

PROGRAMMING THE MANUAL OPENING OF THE BIN

 **TASK 4:** Complete the following table with the final values you used in the code and the inputs and outputs used so far:

Variable name	Variable type	Variable Value

Inputs	Outputs

TRAINING THE LEARNING MODEL

Teachable Machine

<https://teachablemachine.withgoogle.com/>

no_item
cardboard
glass
containers
organic

The screenshot displays the Teachable Machine web interface. At the top left, the 'Teachable Machine' logo is visible. Below it, there are three class containers: 'containers', 'glass', and 'cardboard'. The 'cardboard' class is currently selected and active, showing a 'Webcam' view on the left and a grid of '163 Image Samples' on the right. A 'Hold to Record' button is located at the bottom of the webcam view. To the right of the main interface, there are two panels: 'Training' and 'Preview'. The 'Training' panel has a 'Train Model' button and a dropdown menu set to 'Advanced'. The 'Preview' panel has an 'Export Model' button and a message that reads: 'You must train a model on the left before you can preview it here.'

TRAINING THE LEARNING MODEL

The interface displays a list of training categories on the left, each with a 'Webcam' and 'Upload' button and a row of image samples:

- glass**: 179 Image Samples
- organic_waste**: 96 Image Samples
- containers**: 134 Image Samples
- cardboard**: 163 Image Samples

In the center, a **Training** panel shows the status **Model Trained** and an **Advanced** dropdown menu.

On the right, a **Preview** panel shows the model's output for a webcam input. The input is set to **ON** and **Webcam**. A **Switch Webcam** button is visible above the video feed. The video feed shows a hand holding a yellow sticky note. Below the video, the **Output** section displays classification results:

Category	Confidence
no_ite...	
glass	
organ...	
contai...	100%
cardb...	

TRAINING THE LEARNING MODEL

The image shows a screenshot of the Teachable Machine web interface. A modal dialog box titled "Export your model to use it in projects." is open in the center. The dialog has three tabs: "Tensorflow.js" (selected), "Tensorflow", and "Tensorflow Lite". Below the tabs, there are three options: "Upload (shareable link)" (selected with a radio button), "Download", and "Upload my model" (a button). Under "Your sharable link:", there is a text input field containing a partial URL: "https://teachablemachine.withgoogle.com/models/[...]". Below this, a note states: "When you upload your model, Teachable Machine hosts it at this link. (FAQ: [Who can use my model?](#))". Under "Code snippets to use your model:", there are two tabs: "Javascript" (selected) and "p5.js". To the right of these tabs is a "Contribute on Github" link with a GitHub icon. Below the tabs, there is a link: "Learn more about how to use the code snippet on [github](#)." At the bottom of the dialog is a code editor with a "Copy" button. The code is as follows:

```
<div>Teachable Machine Image Model</div>
<button type="button" onclick="init()">Start</button>
<div id="webcam-container"></div>
<div id="label-container"></div>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest/dist/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@latest/dist/teachablemachine-image.min.js"></script>
<script type="text/javascript">
  // More API functions here:
  // https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/image

  // the link to your model provided by Teachable Machine export panel
```

The background interface shows a "Webcam" section with a "Switch Webcam" button and a video feed area. Below the video feed is a "Hold to Record" button. There is also a "background" section with a "73 Image Samples" label and "Webcam" and "Upload" buttons. On the right side, there are "Preview" and "Export Model" buttons, and a "Webcam" dropdown menu.

PROGRAMMING THE MICRO:BIT TO USE THE LEARNING MODEL

The image shows the Microsoft MakeCode IDE for the Micro:bit. The interface is split into a left sidebar with a block palette and a main workspace for writing code. The top bar shows 'Microsoft | micro:bit' and tabs for 'Blocks' and 'JavaScript'. The bottom bar includes a 'Download' button, a file name 'AI_recycle_bin', and navigation icons.

Block Palette (Left Sidebar):

- Servos
- Loops
- Logic
- Variables
- Math
- NeZha
- PlanetX_Base
- PlanetX_Display
- PlanetX_IoT
- PlanetX_AI-Lens
- Extensions
- Advanced
- Functions
- Arrays
- Text
- Game
- Images
- Pins
- Serial
- more

Main Workspace (JavaScript):

```
on start
  serial redirect to USB
  show icon
  set lid_closed to 110
  set lid_open to 60
  Set 180° servo S1 angle to lid_closed °

on button A pressed
  Set 180° servo S1 angle to lid_closed °
  show leds

on button B pressed
  Set 180° servo S1 angle to lid_open °
  show leds
```

Block Palette (Right Side):

- serial write value "x" = 0
- serial write string ""
- serial write numbers array of 0 1 - +
- serial read line
- serial read until new line ()
- serial on data received new line ()
- serial read string
- serial
- redirect to
- TX P0
- RX P1
- at baud rate 115200
- serial redirect to USB

PROGRAMMING THE MICRO:BIT TO USE THE LEARNING MODEL

TASK 5: This is part of the code you need to automate the system. Can you add the missing blocks?

The image shows the 'Logic' category in the Micro:bit IDE's block palette. Under the 'Comparison' sub-category, three blocks are visible: an equals sign (=), a less than sign (<), and a double equals sign (==). A green arrow points from the double equals sign block to the 'else if' block in the main code editor.

```
serial on data received new line ( )
set SerialData to serial read until new line ( )
if SerialData = "glass" then
  show string "Glass"
  pause (ms) 2000
else if SerialData = "cardboard" then
  show string ""
  pause (ms) 2000
else if SerialData = "containers" then
  //
else if SerialData = "organic" then
  //
else if SerialData = "no_item" then
  //
show icon [ ]
```

[]

[]

[]

PROGRAMMING THE MICRO:BIT TO USE THE LEARNING MODEL



TASK 6: Only the assigned recycling category should open the bin; the others should only display feedback.

We have been assigned the following category : _____

You need to add the code to open the lid under the correct conditional:

- opens the lid
- waits 5 seconds
- then closes it again

Write the blocks with the variables, pins you will be using to make the system work:

Else if SerialData = ...

Set 180° servo S1 angle to...

CONNECTING THE MICROBIT TO THE LEARNING MODEL

MAKE: AI Robots (<https://makeairobots.com/>)

English

MAKE: AI ROBOTS

NEW Oct 27, 2024: We have released a new AI powered chatbot for your AI Robot projects! Use the power of a full large language model to chat with your project and make AI powered decisions. You can access the new tool at

Chatbot Imagination Lab

Give your micro:bit project AI superpowers! This site will allow your micro:bit to use your computer's webcam to recognize images. You can recognize facial expressions, poses, and any object you can think of to put in front of your webcam! What will your micro:bit do when it can tell if you're smiling, or if you've deposited the right amount of coins?

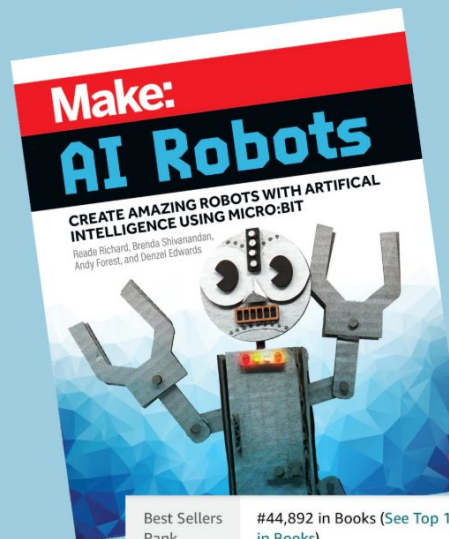
Use the links below to get your Teachable Machine image recognition model and your micro:bit starter code. Then click on the third button to connect everything together!!

1. Google Teachable Machine AI Model

2. Starter micro:bit code

3. Connect your micro:bit to your AI!

Select the Microbit in the popup screen. Psst, it might be called mbed Serial Port.



Best Sellers Rank	#44,892 in Books (See Top 100 in Books)
	#1 in Technology Books for Young Adults
	#5 in Computers for Young Adults
	#9 in Microprocessor & System Design

CONNECTING THE MICROBIT TO THE LEARNING MODEL

Paste your Google Teachable machine model link here:

Choose Camera:

Choose Audio:

Ready!

IMAGE MODEL


This is a Recognition Project - where the AI will be able to identify the classes you made based on the input you give it!

RESULTS!

no_item	0%
containers	99%
cardboard	0%
glass	0%
organic	0%

Reset the AI recognition

Downloaded new code? Reconnect now



OPEN MESSAGE LOG

HELP!

**TASK 7:**

Did the system work the first time? Explain what problems you had and how you solved them. Think about problems using the apps, wiring, coding, hardware, etc.

Some questions to think about	Words to use in your explanation	Some prompts to help you complete the task
<p>What problem did you have? What do you think caused the problem? How did you fix it? Did the system work after the change?</p>	<p>wiring, servo, variable, code, connection, class name, serial data, camera, model, test, error, solution, Learning model</p>	<p>“At first, our system did not work properly. The main problem was _____. We checked _____ and realised that _____. We solved it by _____. After that, the system _____.”</p> <p>“The lid did not open properly because...”</p> <p>“The servo angles were not correct, so...”</p> <p>“The micro:bit did not connect because...”</p> <p>“The model did not recognise the object because...”</p> <p>“The class names did not match the code, so...”</p> <p>“We fixed the problem by changing...”</p> <p>“After testing again, we found that...”</p>



TASK 8: Reflection questions:

1. Does this system actually reduce waste contamination?
2. Is recycling the best solution, or should we reduce and reuse first?
3. What are the limits of AI in solving environmental problems?

POSSIBLE PROGRAMMING SOLUTION

NOTE:

In this example the bin only opens if the data received is "containers"

The angles in the variables may need to be adjusted for correct operation

```
on start
  serial redirect to USB
  show icon [grid]
  set lid_closed to 110
  set lid_open to 60
  Set 180° servo S1 angle to lid_closed °

on button A pressed
  Set 180° servo S1 angle to lid_closed °
  show leds [grid]

on button B pressed
  Set 180° servo S1 angle to lid_open °
  show leds [grid]
```

```
serial on data received [new line ()]
  set SerialData to serial read until [new line ()]
  if SerialData = "glass" then
    show string "Glass"
    pause (ms) 2000
  else if SerialData = "cardboard" then
    show string "card/paper"
    pause (ms) 2000
  else if SerialData = "containers" then
    show string "containers"
    show icon [grid]
    Set 180° servo S1 angle to lid_open °
    pause (ms) 5000
    Set 180° servo S1 angle to lid_closed °
  else if SerialData = "organic" then
    show string "organic"
    pause (ms) 2000
  else if SerialData = "no_item" then
    show icon [grid]
```