AI Enhanced Cardboard Robotics with Microbit

12. Hisar Annual Coding Summit: Crossing The Lines

12-13 December 2025

Mehmet Bener, Arya Kesik, Yiğit Okutan

Table of Contents

Materials Used

During the workshop, the following materials will be used for participants to design and program their robots:

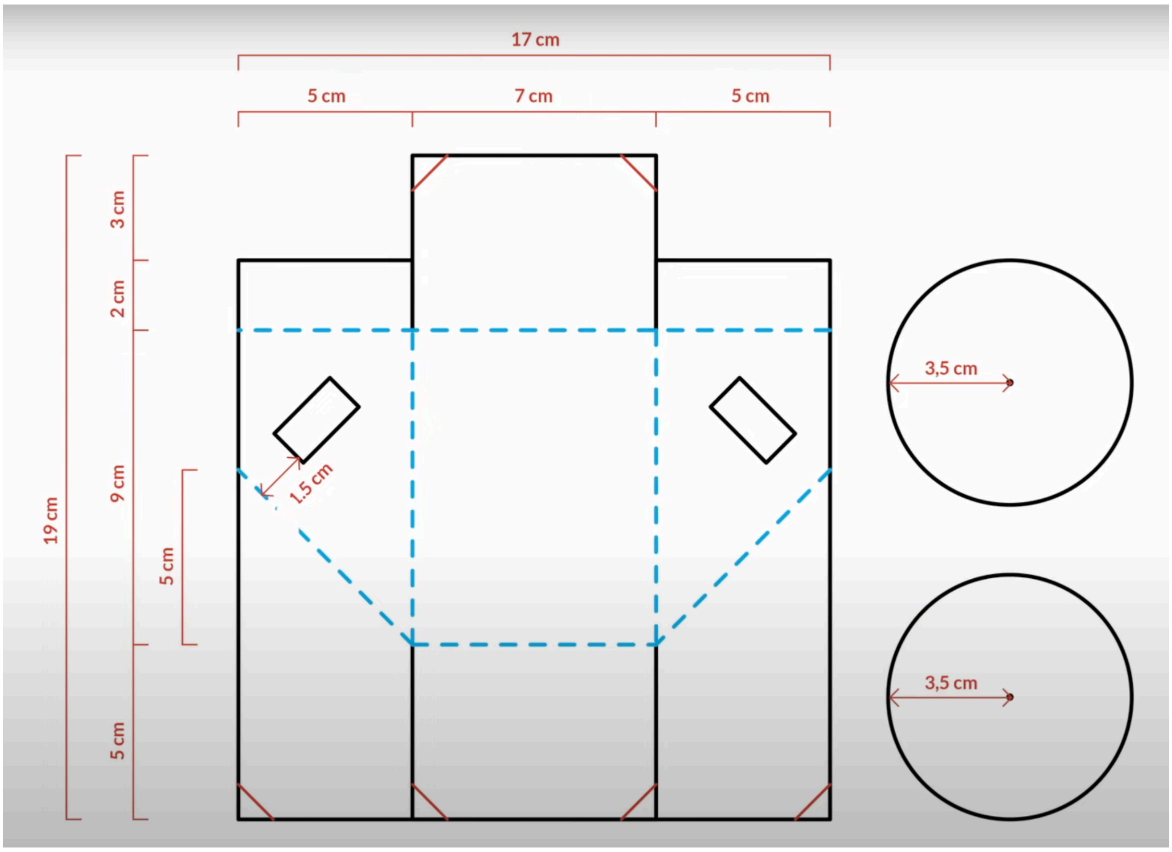
1. **Micro:bit**
- Used as the main control unit of the workshop.
- Programmed to control servo motors and run movement codes.
2. **Battery Holder (with 2 x AAA Batteries)**
- Used to power the Micro:bit and servo motors.
3. **Breadboard**
- Used to connect circuit components and organize servo motor connections.
4. **Servo Motors (x2)**
- Used to enable the robot's movement.
- Controlled via the Micro:bit.
5. **Cardboard**
- Used to create the robot's body.
- Allowed participants to design their robots.
6. **Connection Cables**
- Used to establish connections between the breadboard, servo motors, and Micro:bit.
7. **Various Adhesive Materials (tape, glue, scissors)**
- Used to assemble cardboard pieces and complete the robot's design.
8. **Scissors and Pen**
- Basic tools used for cutting and marking the cardboard.

Robot Design

In this part of the workshop, students will create their own robot bodies using cardboard and other materials. This stage enhances both creativity and problem-solving skills.

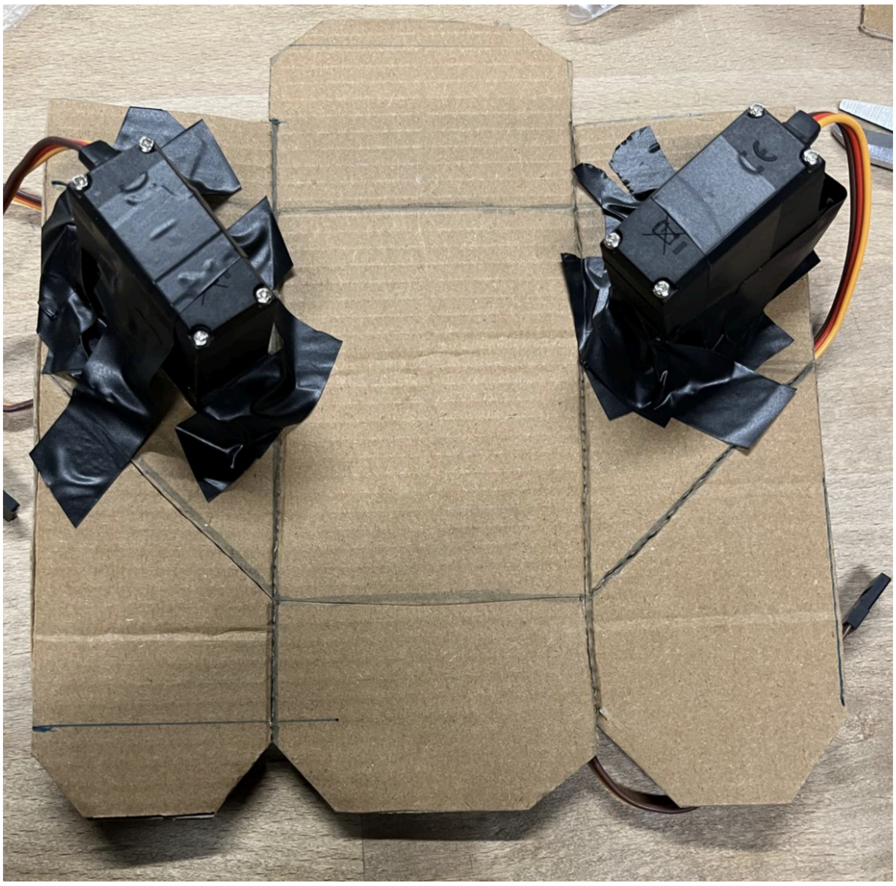
1. Body Design

- **Design Template:**
- Participants are provided with a template that they can draw on cardboard. The template has marked points for cutting, folding, and gluing.
- **Drawing and Preparation:**
- Participants will draw the provided template onto their own cardboard and mark the necessary cutting and folding points.
- **Cutting and Assembly:**
- The cardboard will be shaped using scissors and other cutting tools.
- The designated areas will be folded and glued.
- After folding, tape will be used to stabilize the robot's shape.



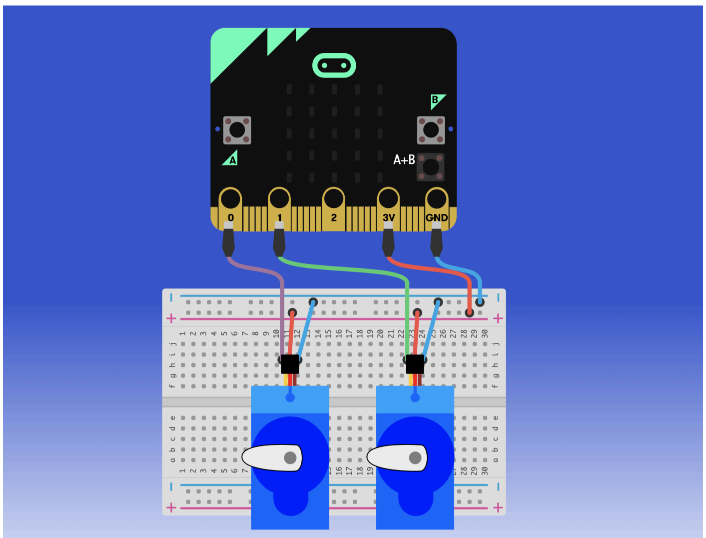
2. Wheel Assembly

- **Setting Up the Wheel with Servo Motors:**
- Servo motors will be placed into the holes designed for them and secured using tape.
- A hole will be made for the servo motor cables, and the cables will be routed to the area where the Micro:Bit will be placed.
- Wheels will be attached to the servo motors using tape, ensuring the wheels are positioned straight and symmetrically.



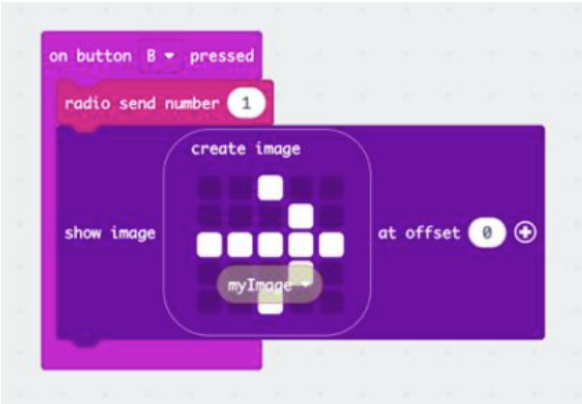
Circuit Assembly

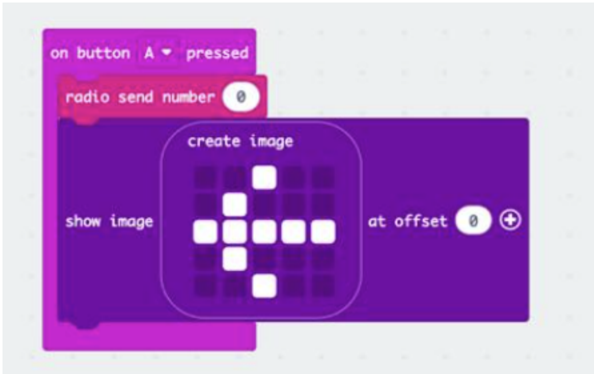
- First, connect the "alligator clip" cables to the 0, 1, 3V, and GND ports on the micro:bit as shown in the diagram below. The 3V port is the power port, and GND is the "ground," meaning the grounding port. The 0, 1, and 2 ports are data ports, which instruct the servos on what to do.
- Next, attach jumper cables to the ends of the alligator clips. Insert the jumper cable coming from the GND into the hole labeled "negative" (-) on the breadboard. Perform the same step for the jumper cable coming from the 3V wire by inserting it into the hole labeled "positive" (+).
- This section of the breadboard is called the "power rail," and it helps to distribute the connected power cables.
- Then, connect jumper cables to the wires coming from the servos, and place these cables into the following holes on the breadboard:
- **Brown wire**: Connect to hole H25.
- **Red wire**: Connect to hole H24.
- **Yellow wire**: Connect to hole H23.
- Next, connect jumper cables to holes J25 and J24. While ensuring proper polarity (+ and -), connect the brown servo wire to the (-), or "negative," power rail and the red wire to the (+), or "positive," power rail. This completes the power supply for the servos.
- Finally, connect the third yellow wire to the alligator clip coming from port 0.
- Now, repeat the same steps for the second servo motor , but use different holes to make the connections.

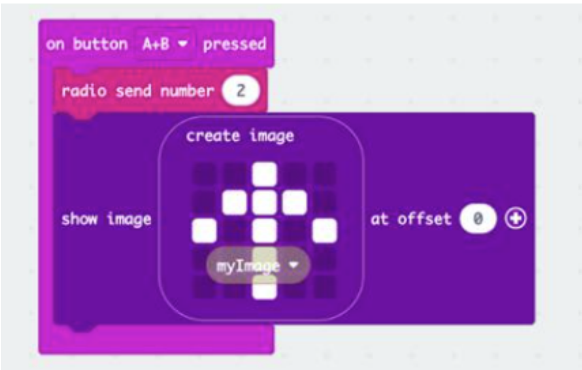


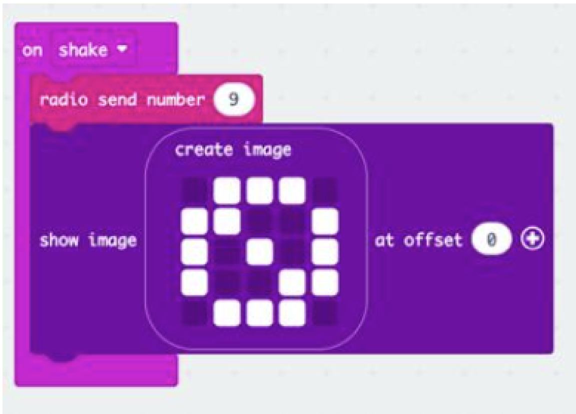
Remote Control of the Robot

- There are various methods for remotely controlling a robot. However, our preferred method is to establish a wireless connection between the micro:bit on the robot and another micro:bit.
- We use a program to manage the servo motors on the microbits

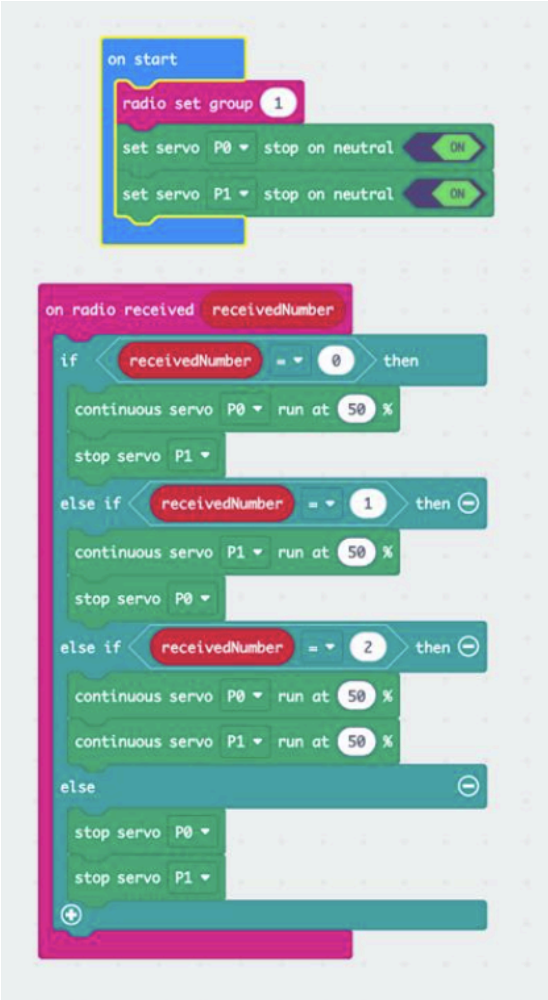








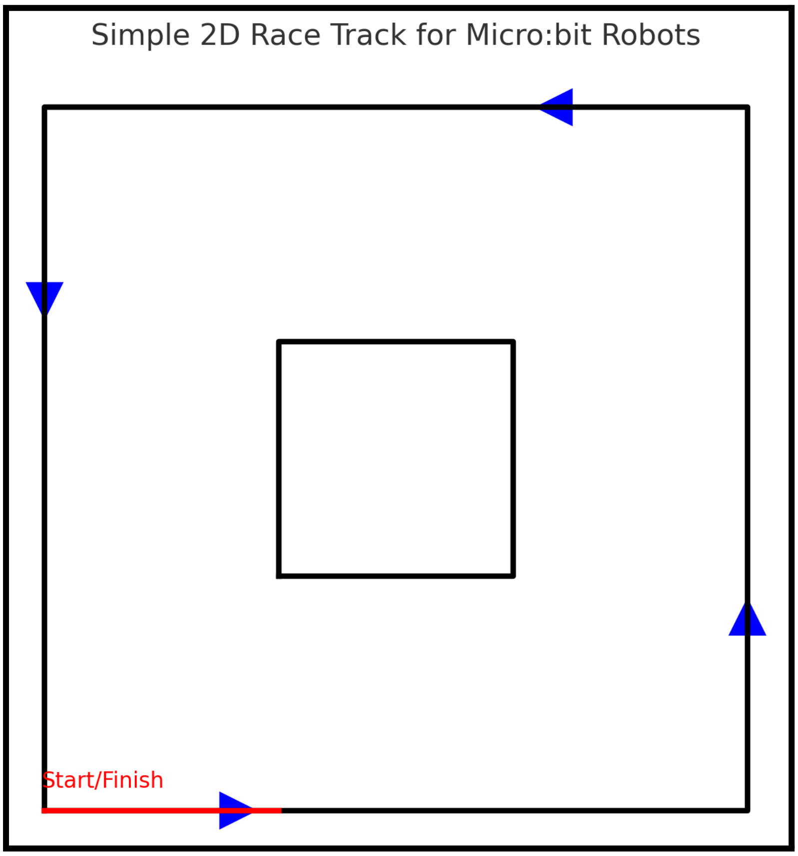
- When "On button B pressed," pressing the B button sends "1" via radio signal, and an arrow pointing to the right is displayed, causing the robot to move right.
- When "On button A pressed," pressing the A button sends "0" and displays an arrow pointing left, causing the robot to move left.
- When "On button A+B pressed," pressing A and B simultaneously sends "2" and displays an arrow pointing upward, causing the robot to move forward.
- When "On shake," shaking the robot sends "9" and displays a circle, causing the robot to stop or respond.
- Commands are transmitted to the robot via radio signals, while visuals indicate the movement status.



- Initially, the "on start" block sets the radio group to "1" and ensures that the servo motors are in a neutral position. This keeps the servo motors inactive in the starting state and prevents the robot from moving.
- The "on radio received" block is triggered when a number (receivedNumber) is received via radio signals:
- When receivedNumber = 0: The continuous servo motor on "P0" operates at 50% speed, causing the robot to turn left. The motor on "P1" is stopped.
- When receivedNumber = 1: The continuous servo motor on "P1" operates at 50% speed, causing the robot to turn right. The motor on "P0" is stopped.
- When receivedNumber = 2: Both "P0" and "P1" servo motors operate at 50% speed, making the robot move forward.
- If a different number is received (else condition): Both "P0" and "P1" servo motors are stopped, and the robot does not move.
- This setup is used to control the direction of a robot via radio signals. Depending on the signal received, the robot can turn left, turn right, move forward, or remain stationary.

Racing the Robots

- At the end of the workshops the robots will race against each other on a racetrack.
- You can find the drawing of the racetrack below.



AI Integration: The Smart Robot's Decision Making

In this section, we will give your robot the ability to make decisions based on a signal from its environment. Simply put, your robot will decide for itself whether it should walk slowly on a dark path or speed up on a bright path.

The **LED screen** on the Micro:bit also functions as a **light sensor**. We will teach your robot a simple rule using this light sensor:

Rule 1 (Dark Environment): If the light level is **low** (dark), the robot should move **slowly**. (For example, moving carefully to avoid an obstacle).

Rule 2 (Bright Environment): If the light level is **high** (bright), the robot should move **quickly**. (For example, everything is fine, it can speed up).

Required Code Blocks

In addition to the previous remote control code, we need to make changes to the program that will be uploaded to the robot.

1. Initial Settings

First, let's add a new variable to the **"on start"** block that will allow the robot to read the surrounding light level. You need to create a variable in this block, and this variable will continuously read the ambient light sensed by the Micro:bit.

2. AI Decision Block

We will use the **"forever"** block to have the robot continuously check this decision.

```
None
// This block will be added to the second Micro:bit loaded
onto the robot.

on forever
  // Read the Micro:bit's light level
  set light_level to light level

  if light_level < 50 then
    // Rule 1: If light is low (Dark) - Move Slowly
    continuous servo P0 run at 30%
    continuous servo P1 run at 30%

    // Display 'K' (Karanlık/Dark) on the surface
```

```
    show string "K"

  else
    // Rule 2: If light is high (Bright) - Move Quickly
    continuous servo P0 run at 80%
    continuous servo P1 run at 80%

    // Display 'A' (Aydınlık/Bright) on the surface
    show string "A"
```

How Does it Work?

Sensing: The Micro:bit continuously reads the amount of ambient light as a number between 0 (total darkness) and 255 (very bright) and assigns this value to the light_level variable.

Decision Making: The robot checks the light_level value. If the value is less than 50 (light_level < 50), Rule 1 (Move Slowly) is executed. If the value is greater than or equal to 50, Rule 2 (Move Quickly) is executed.

Action: According to the selected rule, the servo motors run at different speeds (30% or 80%), and the robot moves in a way that is consistent with this decision.

This simple structure uses the principle of conditional logic (If-Then), which is one of the fundamentals of artificial intelligence, to allow your robot to react to its environment. This means the robot observes and changes its behavior on its own!

For more projects visit:
event.hisarcs.com

If you have any questions you can contact us from:
idealab@hisarschool.k12.tr

To contact our workshop teachers:
mehmet.bener@hisarschool.k12.tr
arya.kesik@hisarschool.k12.tr
yigit.okutan@hisarschool.k12.tr