Moving TPBot

Introduction to TPBot and micro:bit

The micro:bit TPBot is a smart car robot that helps students explore key concepts in the Digital Technologies curriculum. By combining the BBC micro:bit with the TPBot smart car, students design, program, and test digital solutions that bring coding and robotics to life.

The following activity challenges students to explore the TPBot and micro:bit (hardware) and choose 2 or more features of these devices to create and implement an algorithm using MakeCode (software).

Curriculum links

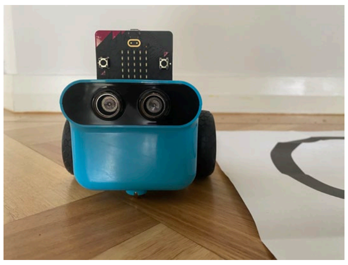
We encourage teachers to adjust the content and focus to suit the needs of their students. The activity can be adapted for students between **Year 3 and Year 6**. More details are provided in the supporting documents.

Digital Technologies

Required resources

You will need the following materials (most supplied in our Lending Library kit):

- BBC micro:bit V2
- TPBot smart car robot
- 4 x AAA batteries (providing a runtime of approximately 1.5hrs)
- USB cable (for downloading code to micro:bit)
- Laptop or tablet with Microsoft MakeCode editor



Pre-requisite knowledge

Prior to this session, students should:

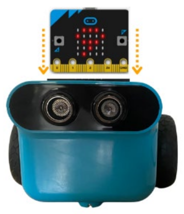
- have basic experience coding a micro:bit using MakeCode or block-based editors.
- understand how to download code to a micro:bit. (See Getting Started with a micro:bit – if required)

Technologies Core concepts	Systems – inputs, processes and outputs	Interactions and impacts
	Systems thinking	Computational thinking

For more information about the Australian Curriculum Technologies core concepts Understand this learning area - Technologies

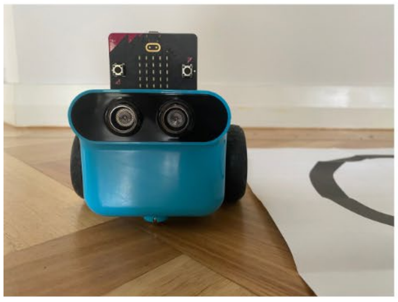
Attaching the micro:bit to TPBot

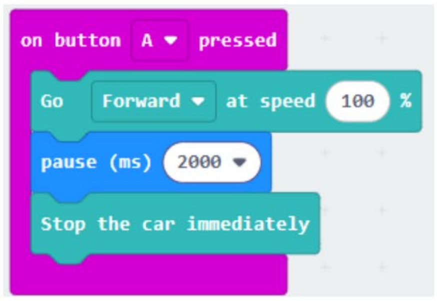
- The micro:bit slides into the dedicated slot on the top of the TPBot
- The gold connector pins MUST face downward into the TPBot
- The LED display should face upward and forward.
- Push firmly but gently until the micro:bit is fully seated. You should feel a slight click when properly connected.
- The micro:bit gets power from the TPBot.



Students will work together to use simple block coding to control a TPBot's movement. Some steps can be extended or shortened depending on the prior experience of students with block coding and micro:bits.

1. Explore the features of the TPBot, including wheels, batteries, input and output ports, RGB headlights, ultrasonic and line-following sensors etc. TPBot is also compatible with LEGO bricks which can be connected to the robot's body. Note: Our lending library lesson, how robots move: Exploring TPBot (no micro:bit required) provides additional information on the features of the TPBot.
2. Discuss the features of a micro:bit - explore its sensors, LEDs, and buttons. See micro:bit overview for details.
3. Open Microsoft MakeCode programming platform in a web browser and explain that we can use MakeCode blocks to code the micro:bit to control the TPBot. To access and code the features of the TPBot, students will need to add the Extension to their MakeCode environment.
    - Click 'New project'
    - Click 'Extensions' and type 'TPBot' in the search bar
    - The TPBot code blocks will now appear
4. Demonstrate creating a simple program in MakeCode (see image) - move the TPBot forward when button_A is pressed on the micro:bit. Students could also add code to change the colour of the RGB headlights or show an icon on the micro:bit's LED display. Ensure students create an algorithm (for example, pseudocode, flowchart or symbols) describing what they want the TPBot to do before creating and implementing their program in MakeCode.
5. Ask students to read their program and predict what the TPBot will do before downloading and running the code.





```
on button A pressed
    Go Forward at speed 100 %
    pause (ms) 2000
    Stop the car immediately
```

To operate with a micro:bit

1. Create code with MakeCode
2. Download code to micro:bit
3. Insert micro:bit into TPBot
4. Run code
   (in example provided - Press Button A)

Sample ideas available on Elecfreaks - Learn page.
The Line-following and Obstacle-avoidance TPBot Create a TPBot smart car that drives along with a black line and stops while it detects obstacles. Note: use the TPBot line tracking map supplied in the lending library kit. ➔ MakeCode direct link
Line tracking Programme to set the TPBot driving along with a black line. ➔ MakeCode direct link

6. Set students a challenge to write their own program which uses 2 or more features of the TPBot and micro:bit. Encourage experimentation with loops (repeat) and sensors (if distance < 10 cm). Students can demonstrate their programs to each other and make changes based on feedback received.

7. Have students share an overview of their challenge, their algorithm, final code and describe how they adapted or improved their algorithms and/or programs.

Pose questions

Use these example open-ended questions to prompt student reflection, check for understanding, and encourage discussion to help students explain their thinking and build on each other's ideas.

- What code blocks did you use? Explain why you selected these.
- How did you fix any errors or issues with your code?
- What was the most difficult part of coding the TPBot?
- What advice would you give another group starting this activity?
- Describe a situation where a robot might need to make decisions based on sensors.
- How can robots help people in real life?

MORE RESOURCES	This is one is a series of TPBot lesson ideas. See the CSER National Lending library resources for more micro:bit and TPBot lesson ideas.

Why is this relevant? (Real world connections)

This lesson connects to the real world by showing students how robots and automated systems can be programmed to follow instructions, just like machines used in warehouses, hospitals, and farms. It helps them understand that digital systems, sensors, and coding are part of many everyday technologies, preparing them for future jobs and problem-solving in a digital world.

Assessment

Observation can be used to check students' ability to carry out tasks aligned to the Australian Curriculum. We have included some suggested questions for teachers to reflect on and to guide these observations.

Checking for understanding

- Can students explain their algorithm using everyday language?
- Does the MakeCode program reflect the algorithm correctly?
- Did they use feedback from testing to refine the algorithm or MakeCode program?
- Is there evidence of branching and iteration in the algorithm and MakeCode program?

For more assessment resources we recommend the Assessment resources on the Digital Technologies Hub.

Australian Curriculum

This activity is suitable for students between Year 3 and Year 6 **Digital Technologies:**

Students in Years 3 and 4 learn to:

- follow and describe algorithms involving sequencing, comparison operators (branching) and iteration (AC9TDI4P02)
- implement simple algorithms as visual programs involving control structures and input (AC9TDI4P04).

By the end of Year 4 students follow and describe simple algorithms involving branching and iteration and implement them as visual programs.

Students in Years 5 and 6 learn to:

- design algorithms involving multiple alternatives (branching) and iteration (AC9TDI6P02)
- implement algorithms as visual programs involving control structures, variables and input (AC9TDI6P05).

By the end of Year 6 students design algorithms involving complex branching and iteration and implement them as visual programs.