ACEBOTT

TinkerBott Smart Car Tutorial

Preface

Our Company

ACEBOTT STEM Education Tech Co.,Ltd
Founded in China's Silicon Valley in 2013, ACEBOTT is a STEM education solution leader We have a team of 150 individuals, including members from research and development, sales, and logistics. Our goal is to provide high-quality STEM education products and services to our customers. We are working together with STEM education experts and our business partners to produce successful STEM products together Our self-own factory also provides CEM services for our clients including Logo customization on product packaging and PCB.

Tutorial

This is a programming course that combines the micro:bit car with a detective suspense story. The story revolves around the "Lumi Detective and the Mysterious Case of ACE Town" where mysterious shadows and sounds appear. You will play the role of a detective assistant, using the sensors on the car to detect the environment, and navigate around the town to crack clues. Through programming, multiple modules can work together to finally uncover the truth behind the incident.

With this kit, you can:

1. Learn how to effectively use the micro:bit development board, including uploading code, understanding its features, and programming in MakeCode.
2. Learn graphical programming knowledge from scratch and master programming skills.
3. Explore various electronic components, such as RGB lights, sensors, and motors, and understand how they work together in the smart car project.
4. Design functions such as lights, automatic obstacle avoidance, intelligent line patrol, infrared remote control, and wireless communication for the smart car.

Overall, the TinkerBott Smart Car Learning Kit is designed for beginners to learn programming based on micro:bit. By using this kit, students will be able to independently design and program a feature-rich smart car, gain a deep understanding of the basic operating mechanisms of smart hardware, and develop the skills to apply what they have learned to solve real-world problems.

After-sales service

ACEBOTT is a dynamic and fast-growing STEM education technology company committed to providing excellent products and quality services to meet your expectations. We value your feedback and encourage you to send us any comments or suggestions at support@acebott.com.

Our experienced team of engineers is committed to quickly resolving any issues or questions you may have while using our products. During the working day, we guarantee to respond to you within 24 hours.

Contents

Prologue

ACE is a peaceful town, but the nights here have become uneasy recently. Residents often hear strange noises late at night, and some even claim to have seen mysterious shadows wandering the streets. The quiet town in the past is shrouded in a layer of uneasy haze, and people are panicking and no longer dare to sleep peacefully at night as usual.

Inspector Lumi is the best inspector in this small town. He decided to explore the reasons behind this and help the residents regain their peaceful lives. He visited the residents, collected clues, and did not miss any suspicious details.

After investigation, Inspector Lumi found that these mysterious shadows are not simple. Their appearance is not without rules, but hides a kind of "mysterious information". This information is transmitted through sound, light, and even hidden tracks. Only by precise analysis and tracking can the truth be revealed one by one.

As the most promising young detective in the school detective agency, you will assist Inspector Lumi to complete this mission. At the same time, you will also have a special helper - TinkerBott smart car. This car has an intelligent brain that can sense the surrounding environment and track hidden clues through sophisticated sensors and programming instructions.

"Hey guys, let's get started!" Inspector Lumi said, "The key to solving these mysterious phenomena lies in this car. We need to use the functions of the car to decipher the clues found in different scenes."

Come on, the adventure has begun, the mystery is about to be revealed, use your wisdom and Inspector Lumi to solve the secrets of the town!

Lesson 1: Mystical Symbols

On the first day of reporting to the police station, you and Inspector Lumi were surprised to find that the TinkerBott smart car had been dismantled. You and Inspector Lumi both had a question in mind: Who dismantled the TinkerBott smart car?

In order to solve this mystery, you and Inspector Lumi visited the residents around. Some residents revealed that the mysterious figure reappeared last night, and before the TinkerBott smart car was dismantled, various icons and symbols kept flashing on the screen, including love hearts and smiley faces.

After some thought, Inspector Lumi realizes that these icons belong to the icon library built into the TinkerBott smart car. He instructs you to try to repair the TinkerBott smart car first, and display the icons in the icon library on the screen to see if you can find any clues in the icons.

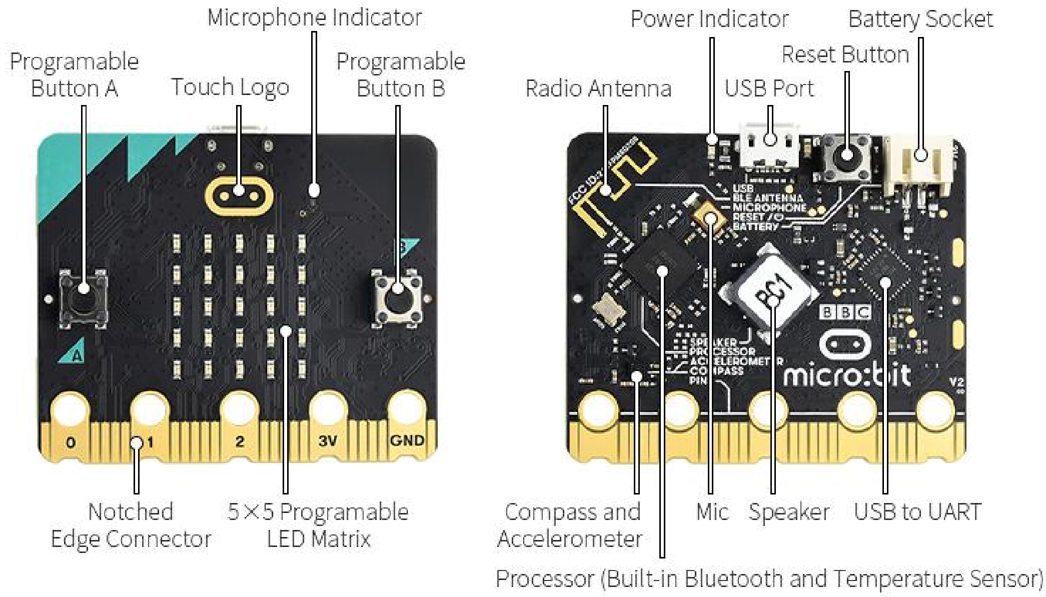
In order to complete these tasks, you need to know some basic knowledge about TinkerBott smart car.

1. Knowledge Learning

1. Get to Know TinkerBott Smart Car

TinkerBott is a smart car based on the micro:bit controller. It can realize multiple functions such as colorful lights, line patrol and obstacle avoidance, wireless communication, etc.

The brain of the car: All functions of the TinkerBott smart car are realized based on the micro:bit main control board (v2.0). The micro:bit main control board is the brain of the smart car, responsible for receiving programming instructions and processing information.

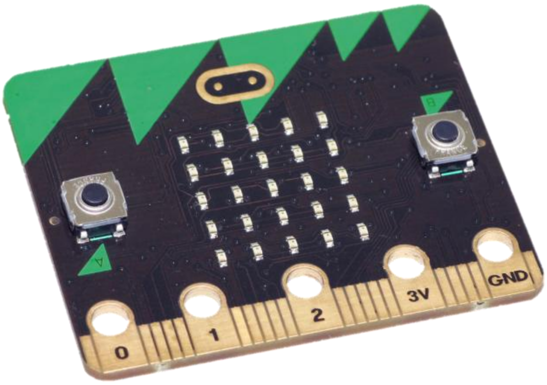


The car's perception system: Just as people can perceive the world through their five senses, the car also has its own "eyes and ears". Various sensors can receive different external information. For example, ultrasonic sensors can identify obstacles in front of them just like eyes and avoid collisions.

The power system of the car: The power system of the car consists of a motor and wheels. The motor is equivalent to the muscles of the car, providing power to drive the wheels to rotate, allowing the car to move forward or backward; the wheels are like the legs of the car, supporting the body of the car and using the power of the motor to help the car move on the ground.

2. Micro:bit Dot Matrix Screen

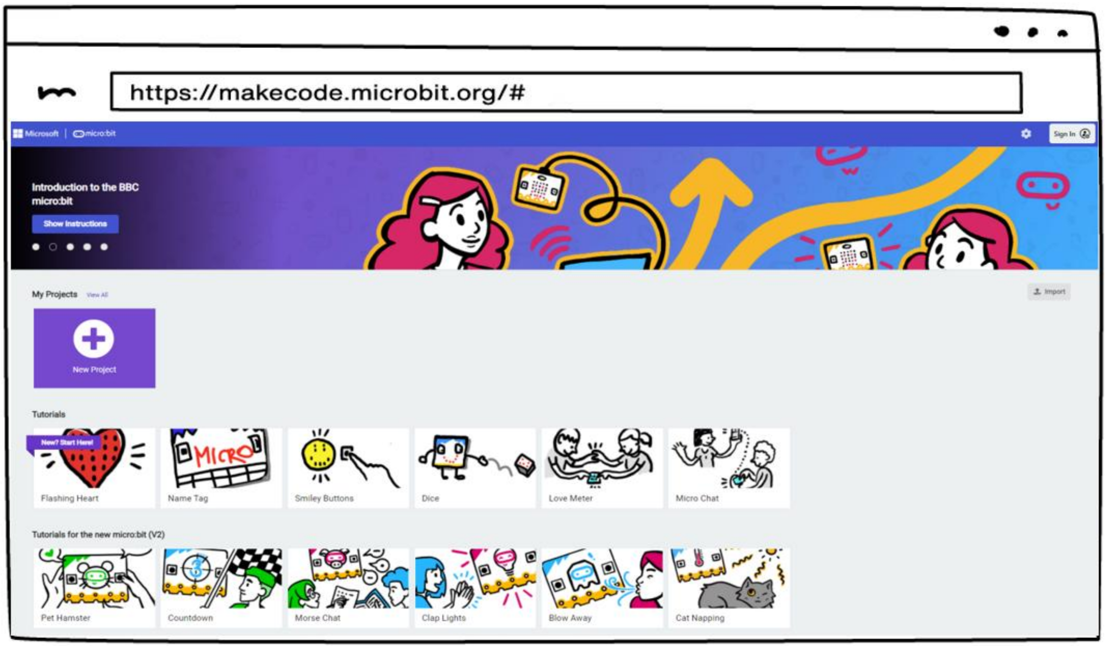
There is a 5x5 LED matrix on the Micro:bit mainboard. Each LED is an independently controllable light-emitting diode, arranged in a matrix structure. The 5×5 LED matrix in the center is like a magic drawing board composed of 25 small stars, on which various icons and symbols can be displayed.



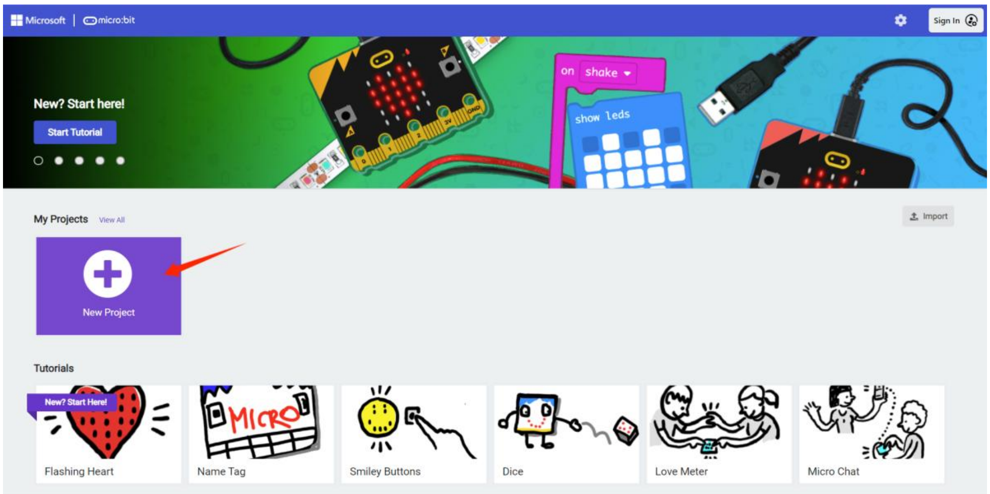
3. Understand the Programming Environment

If you want TinkerBott to achieve a certain function, there is an indispensable tool, that is the MakeCode programming platform, which is an online graphical programming platform. With it, you can program the car and set control instructions.

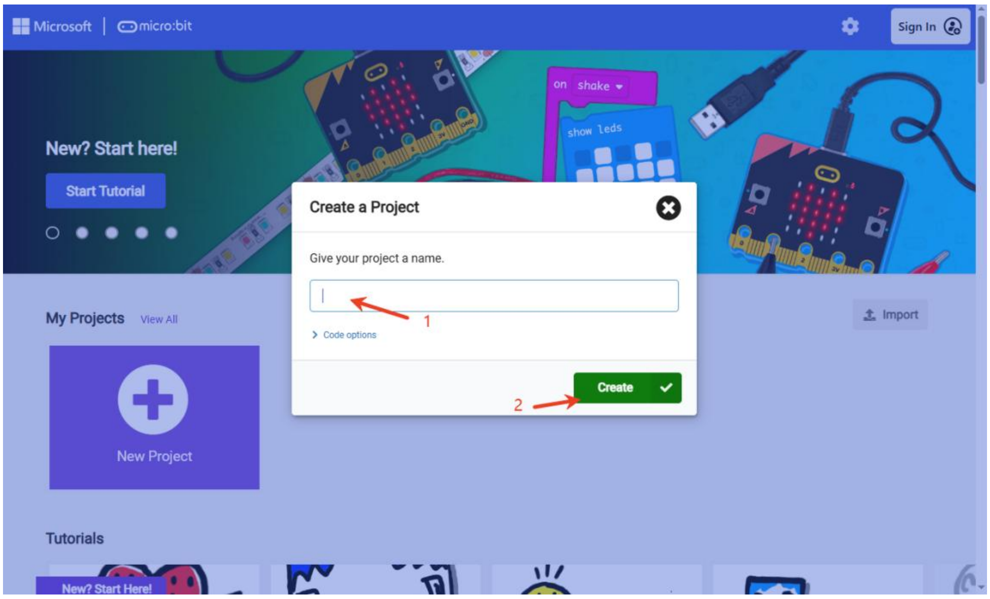
1. Open the MakeCode website on the web:https://makecode.micro:bit.org.



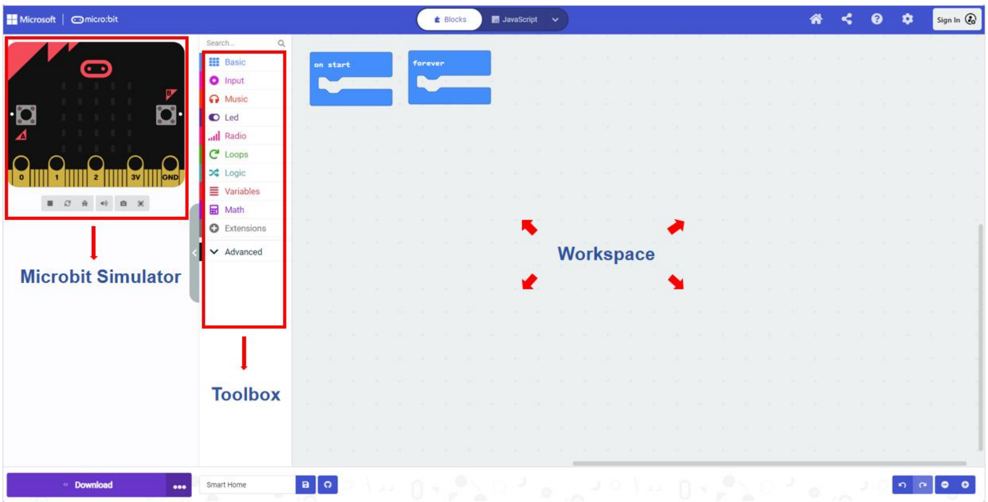
2. Click "New Project" to enter the programming interface.



3. Fill in the project name and click "Create"



4. Understand the programming interface



The operation interface of MakeCode is mainly divided into three parts:

micro:bit Simulator, Toolbox and Workspace.

●Micro:bit Simulator: Used to preview the state of the board when the code is running

●Toolbox: stores code blocks and drags code blocks from Toolbox to Workspace

●Workspace: You can stitch code blocks together to build your program

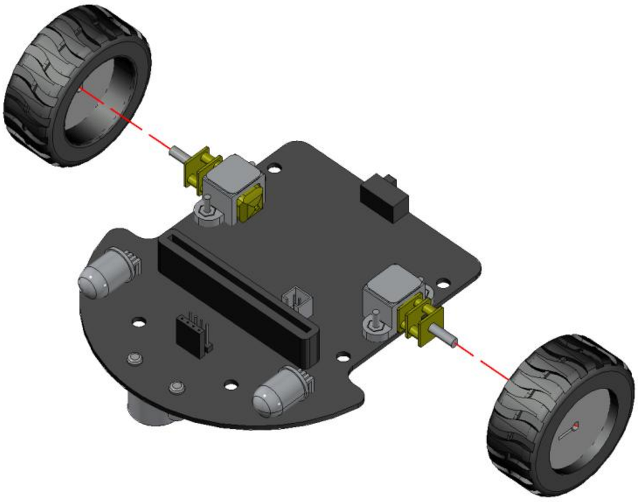
2. Mission Decryption

Task 1: Build TinkerBott Smart Car

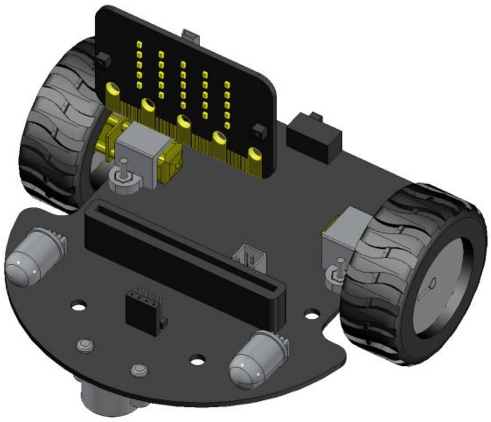
To help Detective Lumi solve the case, you need to assemble the TinkerBott smart car first. The following is the parts list of the TinkerBott smart car. Please check if there are any missing parts in the list.

Once you have completed the checklist, follow the steps below to build your TinkerBott smart car.

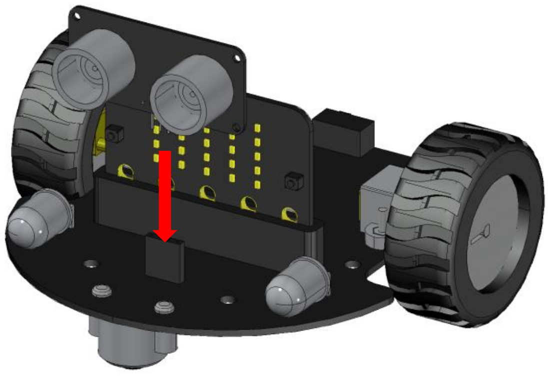
Step 1: Install the wheels.



Step 2: Install the micro:bit mainboard.

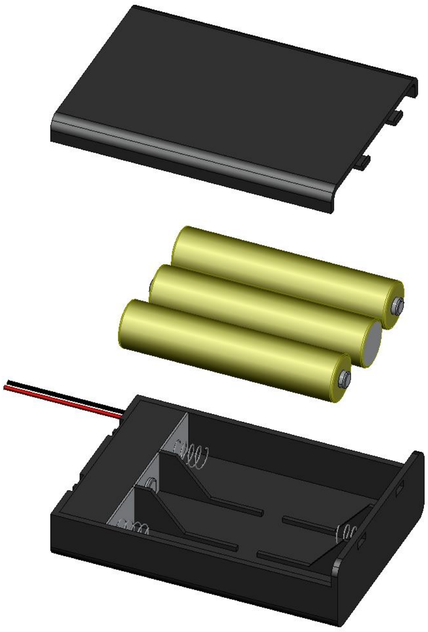


Step 3: Install the ultrasonic module.

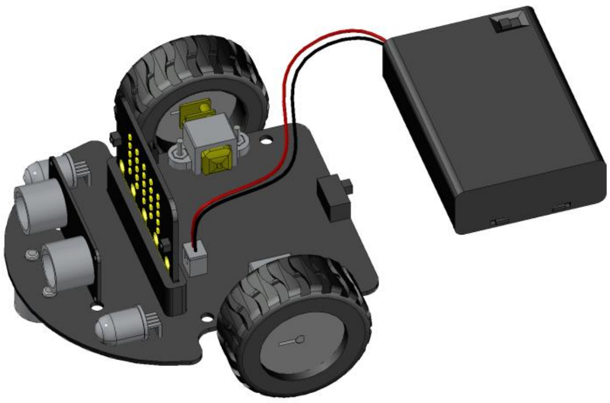


Step 4: Install the battery.

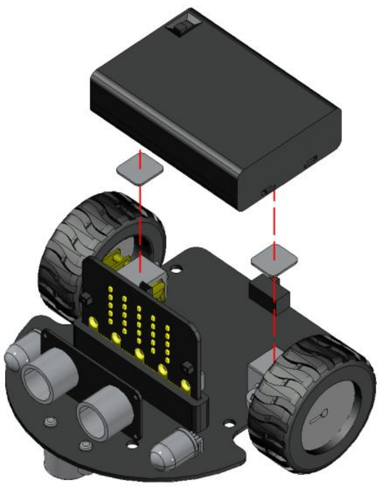
(1) Place three AA batteries in the battery compartment.



(2) Connect the battery power cable to the power interface of the car.



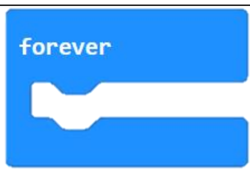
(3) Use double-sided tape to stick the battery to the motor.



Task 2: Display the Built-in Icons on the Micro:bit Dot Matrix Screen

(1) Command Learning

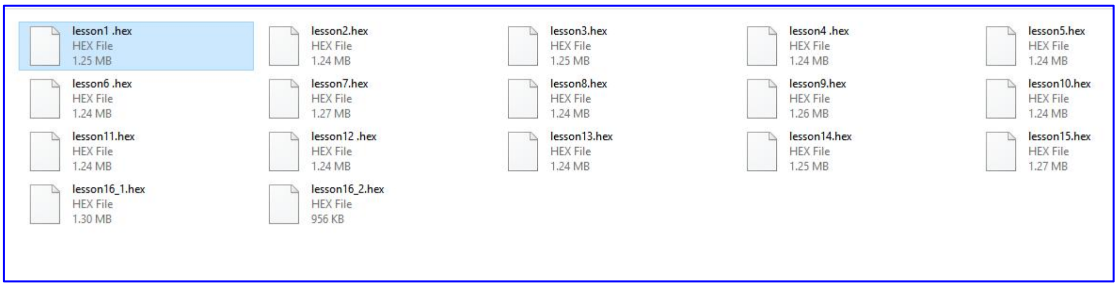
Before programming, we need to understand the instructions related to controlling the dot matrix screen. Find the building blocks below in the "Basic" instruction set.

Building Blocks	Description
	Initialization instruction, executed once at the beginning of the program
	Permanent loop instruction, forming an infinite loop structure, repeatedly executing the internal code block
	Display a preset icon or symbol, such as a heart, a smiley face, and so on.

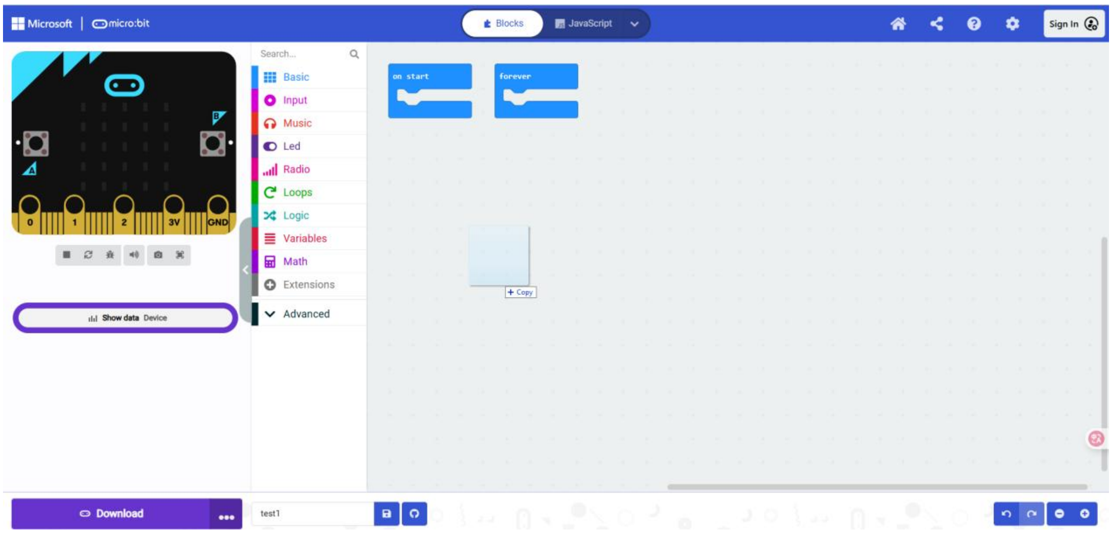
(2) Write a Program and Upload It to the Micro:bit

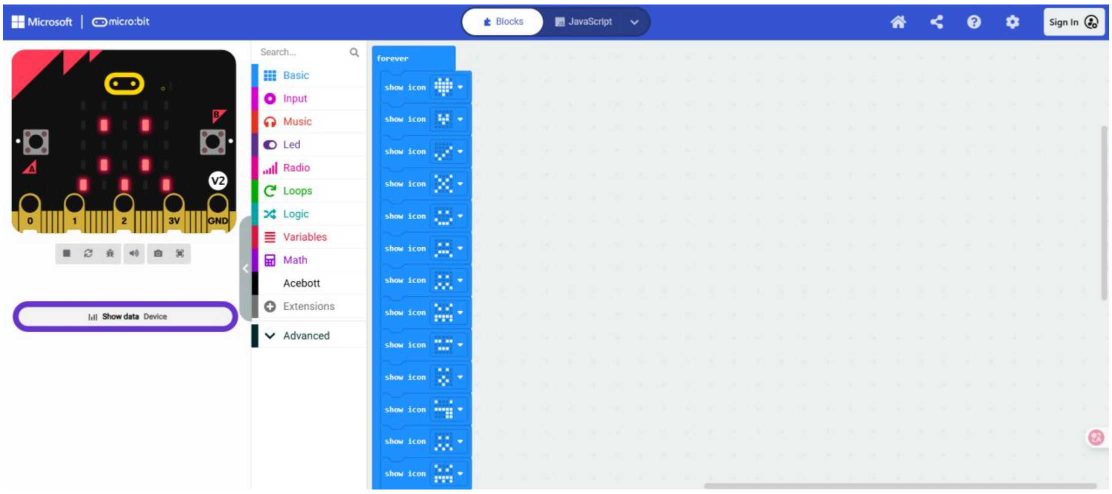
Step 1: Open the reference program file.

Find the program file and open it. [Click here to open the program file for Lesson 1](#).



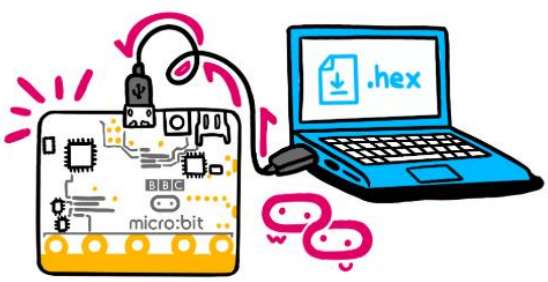
Open the Makecode programming interface, click the reference program file of this lesson with the left button of the mouse, and drag it to the workspace area to open the reference program.





Step 2: Download the program to micro:bit.

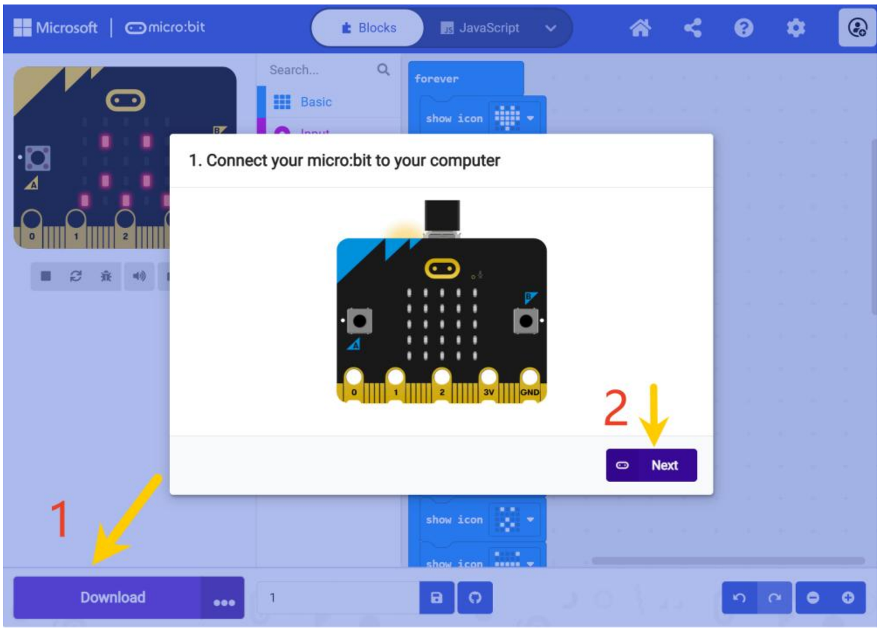
① Connect the micro:bit motherboard to the computer via a MicroUSB cable. The power indicator on the back of the micro:bit motherboard will light up when the connection is correct.



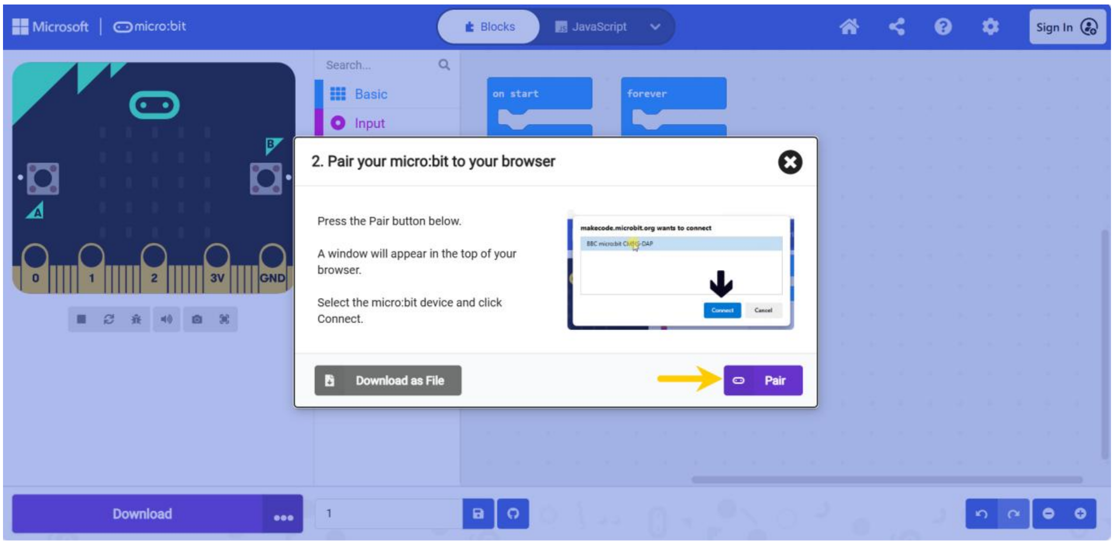
And there will be an extra disk in your computer folder. As shown below:

② The next step is to pair the micro:bit device. If this is the first time you are pairing the micro:bit with your computer, you will need to complete a few setup steps. The system will guide you through the first-pairing steps (subsequent use will automatically connect).

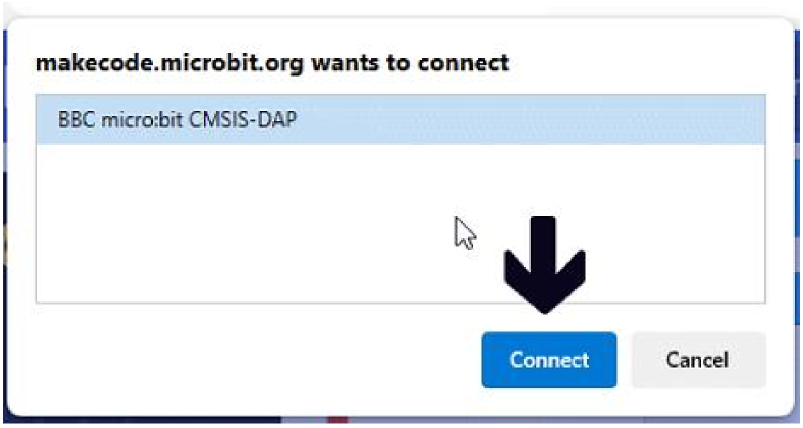
- Click the "Download" button at the bottom of the editor window. Then click "Next" in the message window.



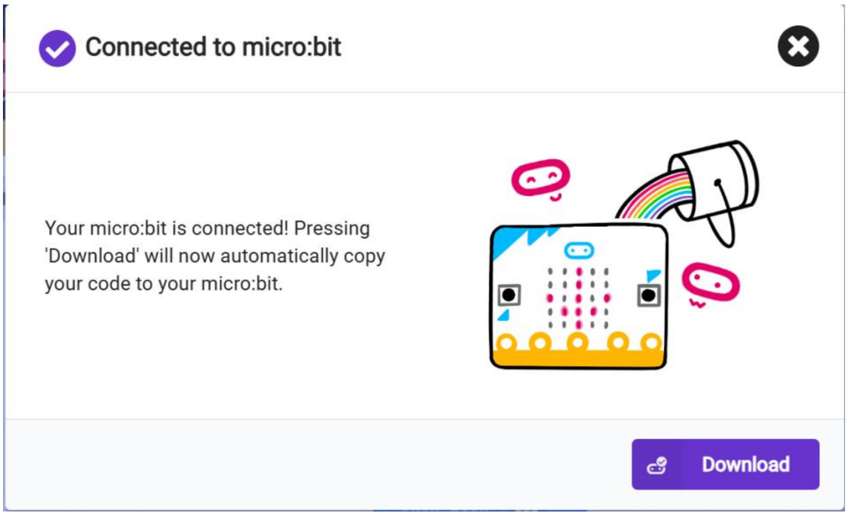
- Another message window will prompt you to pair with the micro:bit device. Click the "Pair" button to view the device list.



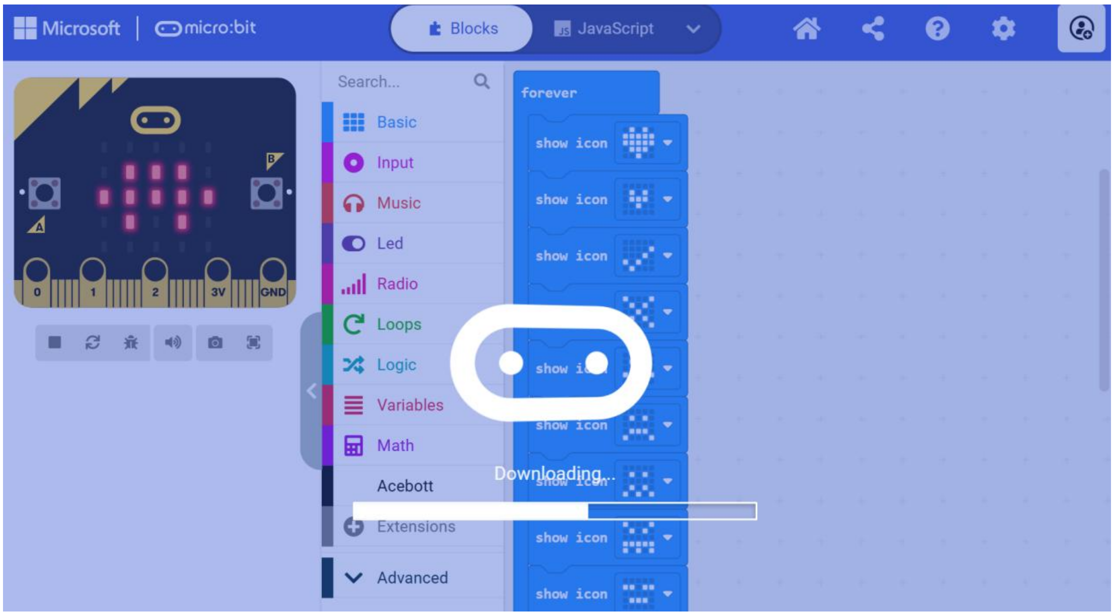
- In the device list, micro:bit will be displayed as the device to be connected. Select the corresponding device and click "Connect".



- Then a window will pop up indicating that the device is connected successfully. Click "Download" in the lower right corner.



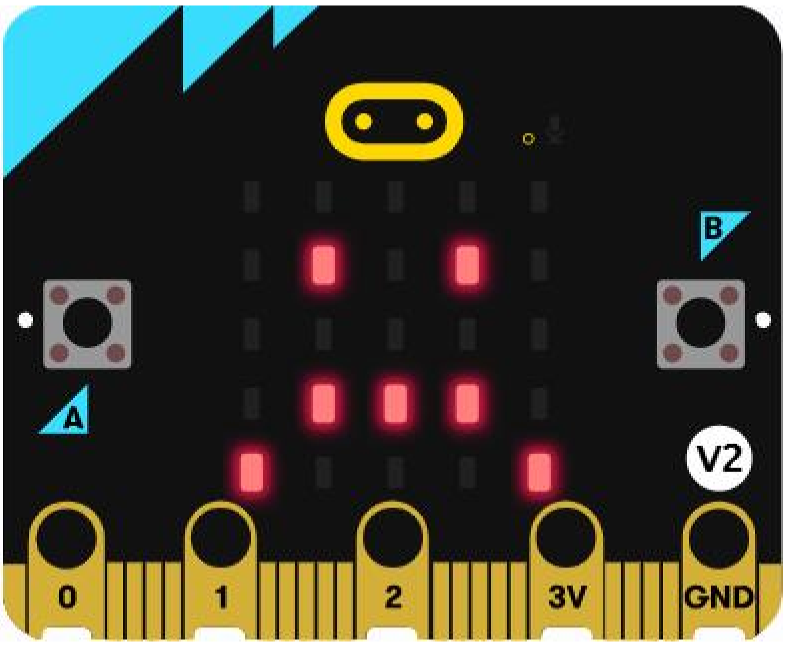
● Next, you can wait for the browser to download the program to the micro:bit controller! If you need to download the program later, just click the "Download" button at the bottom of the editor window to download the program directly to your device.



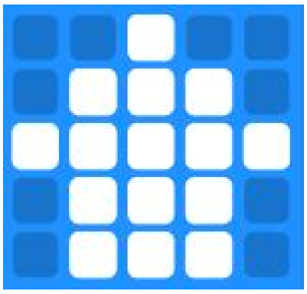
Note: Here the program displays the first 4 rows of built-in patterns of the TinkerBott smart car. You can also display all the built-in patterns.

(3) Run the Program

After the program is downloaded to the main control, multiple icons will be displayed on the dot matrix screen of micro:bit.



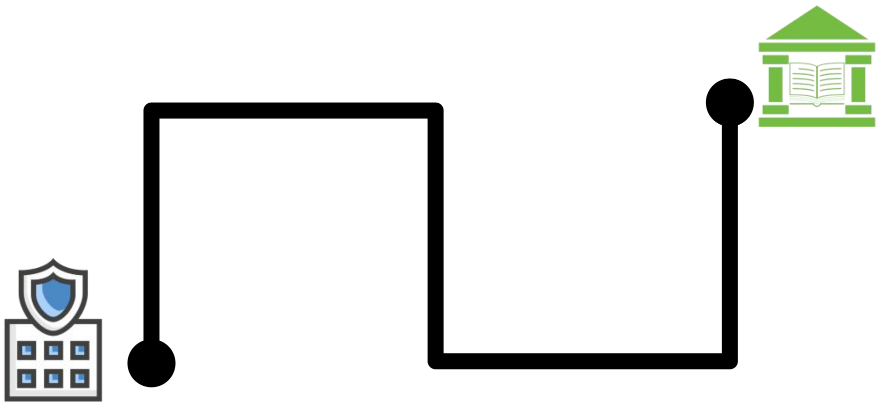
Among the many icons, Inspector Lumi found a suspicious icon, which was an icon of a house. Its shape was very familiar, as if he had seen it somewhere before.



Detective Lumi lowered his head in thought, and suddenly said: "I see, this is the town library. The town library is the same shape as this house. We should go to the library to see if there are any new clues there."

Lesson 2: Action! Car

Inspector Lumi looks at the built TinkerBott smart car and says to you: "We need to let the smart car follow us to the library to help us solve the clues. Now we need you to program it to drive automatically to the library." Then, Inspector Lumi gives the route from the police station to the library, and you need to program the smart car to complete the movement of the following route.



Before completing the task, you need to understand the car's power system and master the motion control method of the car's motor.

1. Knowledge Learning

1. DC Motor

A DC motor is a device that converts electrical energy into mechanical energy. The current drives the motor to rotate, thus achieving the movement of the car. In the TinkerBott smart car, an N20 DC motor is used to drive the wheels, allowing the smart car to move forward, backward, or turn.

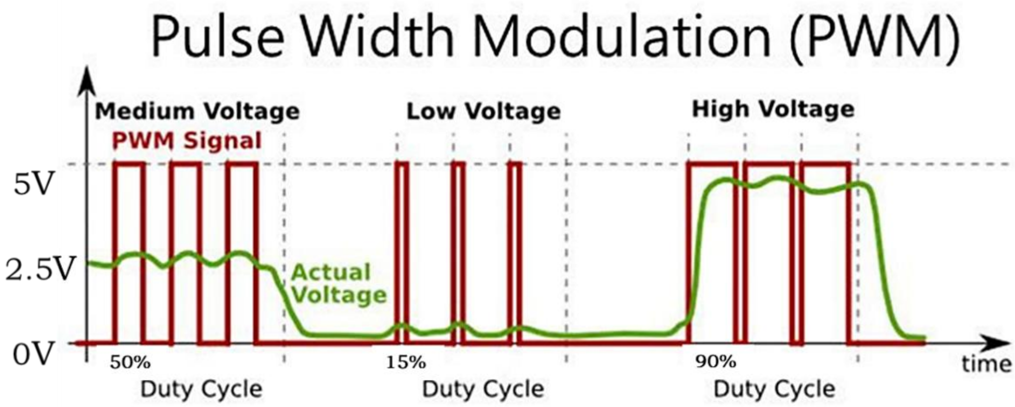
(1) Working Principle of DC Motor

The DC motor drives the internal coil of the motor to generate a magnetic field through the change of current, thereby driving the rotor to rotate. By changing the direction of the current, the motor can be rotated forward (the car moves forward) or reverse (the car moves backward). The speed of the motor can be controlled by a PWM signal.

By adjusting the duty cycle of the PWM signal, the output voltage of the micro:bit is adjusted, thereby changing the motor speed. In the micro:bit, the PWM signal is output through the 

instruction, and the PWM value range is 0-1023.

- When the PWM value is 1023: it means 100% duty cycle, the output voltage is the maximum value, and the motor runs at full speed.
- When the PWM value is 512: it means 50% duty cycle, the output voltage is half of the maximum value, and the motor runs at medium speed.
- When the PWM value is 0: it means 0% duty cycle, the output voltage is 0, and the motor stops.



(2) Parameters of N20 motor

Parameter	Description
Voltage	12V
Speed	200rpm
Control method	PWM wave signal

2. Basic Movement of the Car

The movement of the car is driven by two DC motors, which control the rotation direction and speed of the left and right wheels respectively.

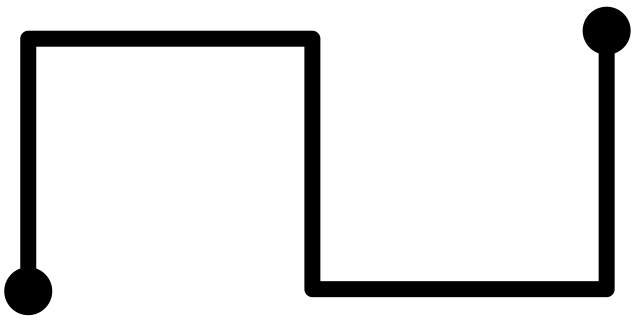
By combining the forward and reverse rotation and speed of the two motors, the following basic actions can be achieved:

Action	Left wheel motor	Right wheel motor
Forward	Forward	Forward
Backward	Reversal	Reversal
Turn left	Reversal	Forward
Turn right	Forward	Reversal
Stop	Stop	Stop

2. Mission Decryption

1. Task Description

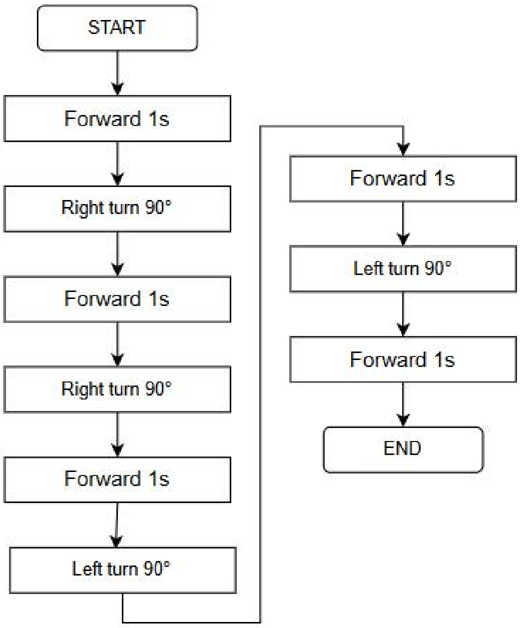
Program the TinkerBott smart car to follow the path below.



2. Programming

(1) Draw a Program Flow Chart

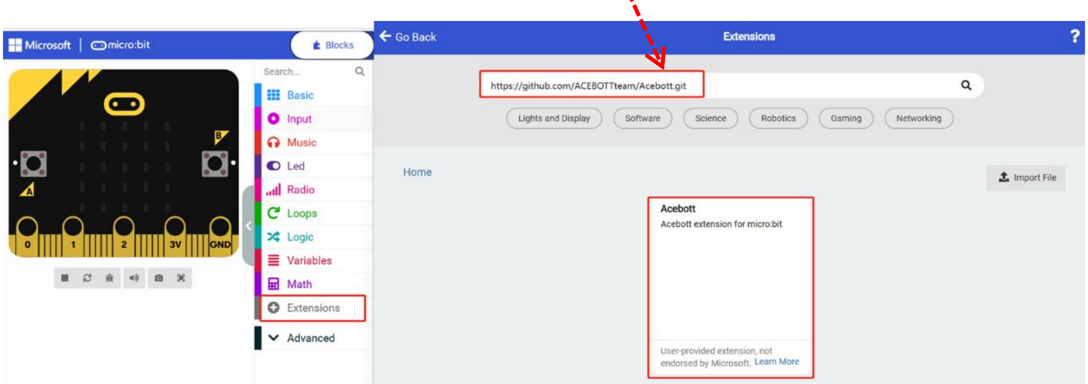
According to the path on the map, the car can automatically drive to the library according to the following process.



(2) Command Learning

To control the TinkerBott car, you need to first add the ACEBOTT extension pack. In the programming interface, click on "Extensions". In the search box of the pop-up interface, enter the **extension pack address in the text box below**, and then click on the car extension pack. At this time, the car's extension pack will be loaded into the programming area.

https://github.com/ACEBOTTteam/Acebott.git



After adding the expansion pack, click Acebott—> Executive—> Microbit car to find the motor control instructions below:

Building Blocks	Description
	Set the driving direction and speed of the car. The driving direction can be "Forward", "Backward", "Left", "Right" and so on.
	Set the speed of the left and right wheels of the car respectively. The first parameter indicates the speed of the left wheel, and the second parameter indicates the speed of the right wheel. The range is from 0% (stop) to 100% (full speed).
	Stop the movement of the car.

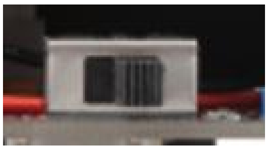
(3) Reference Program

Click here to view the program "Lesson 2" for the car movement.

Note: The delay command in the program determines the distance the TinkerBott smart car moves forward and the angle it turns. You can modify the parameters to control the movement of the car. The longer the delay time, the longer the car moves forward and the greater the angle it turns. Considering the different friction between the wheels and the ground, you need to set it according to the specific situation.

(4) Run the Program

After downloading the program to the main control, place the car on the ground, turn the switch on the back of the TinkerBott smart car from "OFF" to "ON", and observe whether the car can move along the set route.



After the program was uploaded, the TinkerBott smart car drove autonomously along the predetermined route and arrived at the library smoothly.

Lesson 3: Operation Deciphering Sound Wave Codes

You and Inspector Lumi come to the library and look for clues in the library. After a while, Inspector Lumi finds a combination lock behind a music book and a note in the music book with two lines of numbers written on it, "262 262 392 392 440 440 392 -" and "349 349 330 330 294 294 262 -".

Could these numbers be the combination lock's password? Inspector Lumi tried to enter the numbers on the paper into the combination lock, but none of them worked. Inspector Lumi looked at the numbers on the paper and wondered, "What do these numbers mean?"

You look at the music book and ask, "Could these numbers be related to music?"

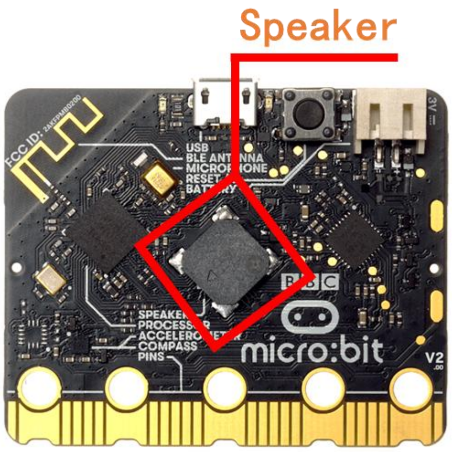
Detective Lumi's eyes lit up and he said to you, "Yes, I understand. These numbers are the vibration frequencies corresponding to certain notes. These two lines of numbers may be the melody of a certain song. Next, let the TinkerBott smart car play this melody and let's listen to see which song it is."

To play this melody, you need to use a buzzer module. Before completing the task, you need to master some knowledge about buzzers and music, and then program the TinkerBott smart car to play this melody.

1. Knowledge Learning

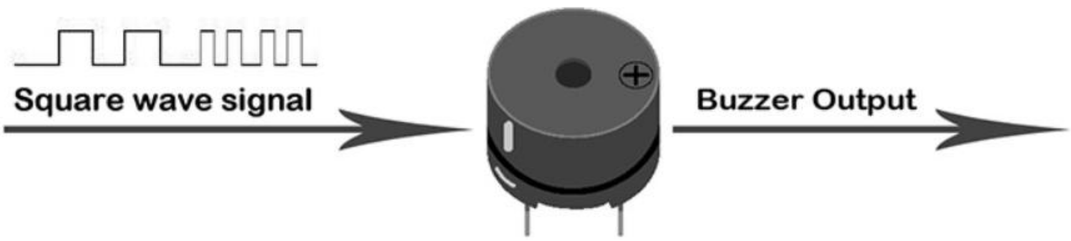
1. Buzzer

A buzzer is an audio signaling device that emits sound by converting electrical signals into sound. It is widely used in alarm systems, timing equipment, alert tone generation, music creation, and other fields. Buzzers are divided into two types: active buzzers and passive buzzers. Micro:bit integrates a passive buzzer that can emit sounds of different frequencies, thereby producing various tones and melodies.



(1) Working Principle of Passive Buzzer

The working principle of the passive buzzer is based on vibration. When the electrical signals of different frequencies are input, the piezoelectric ceramic inside the buzzer will vibrate at the corresponding frequency, thus emitting different tones. By controlling the frequency and duration of the signal, a melody can be combined. In Micro:bit, we can call the music module through the graphical programming tool, convert the notes into signals of the corresponding frequencies, and drive the buzzer to play music.



(2) Features of Microbit Buzzer

① **Adjustable volume:** The volume of the buzzer can be controlled by programming to meet the needs of different scenarios.

② **Diverse tones:** It can produce sounds of different frequencies, produce various tones, and combine to create rich melodies.

③ **Frequency range:** The buzzer can emit sounds in the frequency range of tens of Hz to tens of thousands of Hz, covering common tones.

2. Music Basics

(1) Correspondence between Notes and Frequencies

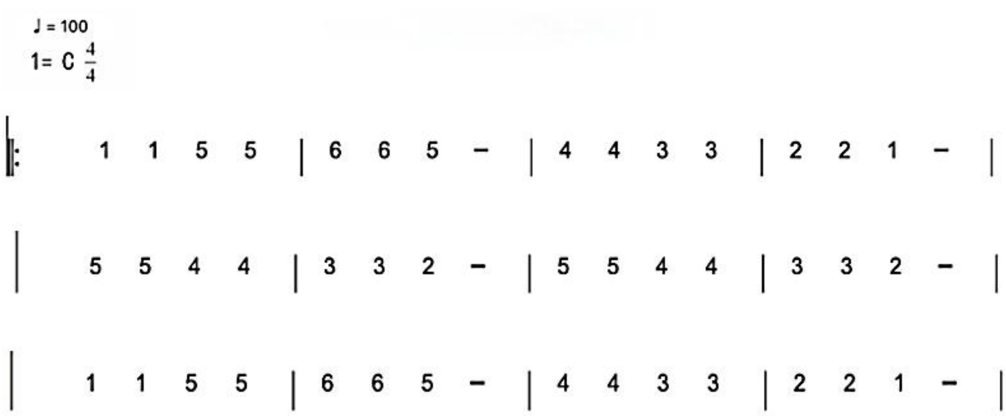
Each note is a sound of a specific frequency. By making the buzzer work at a fixed frequency, the sound of the corresponding note can be produced. The following table shows the relationship between notes and frequencies.

Note	C3(1)	D3(2)	E3(3)	F3(4)	G3(5)	A3(6)	B3(7)
Freq	131	147	165	175	196	221	248
Note	C4(1)	D4(2)	E4(3)	F4(4)	G4(5)	A4(6)	B4(7)
Freq	262	294	330	350	393	441	495
Note	C5(1)	D5(2)	E5(3)	F5(4)	G5(5)	A5(6)	B5(7)
Freq	525	589	661	700	786	882	990

(2) The Beat of the Music

The melody and beat of a song can form beautiful music. The beat determines the speed of each note. In the same song, the duration of each beat is fixed. BPM (Beats Per Minute) is usually used to indicate the speed of music beats, which indicates the number of beats per minute and is closely related to the music control of the buzzer. If the BPM value is N, the duration of each beat is 60/N seconds. For example, the beat duration of 120BPM is 0.5s.

Below we will explain the specific beats using the simplified notation as an example.



In the above simplified notation, ♪ =100 means that the BPM of this song is 100.

1. Ordinary notes, such as the first note 1, occupy one beat.

2. Underlined notes represent 0.5 beats.

3. Some notes are followed by a ·, which means adding 0.5 beats, that is, one note is 1+0.5=1.5 beats.

4. Some notes are followed by a —, which means adding one more beat, that is, one note is 1+1=2 beats.

2. Mission Decryption

1. Task Description

Program the TinkerBott smart car to play the melody provided in the clue, BMP=120:

262 262 392 392 440 440 392 —

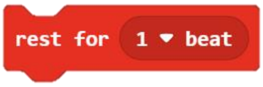
349 349 330 330 294 294 262 —

2. Programming

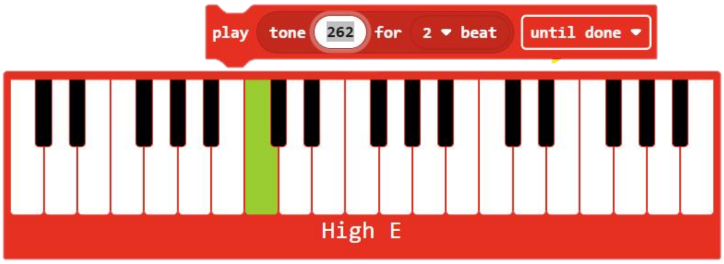
(1) Command Learning

On the programming platform, you can use the building blocks in the "Music" instruction set to program this song. In order to play the notes of the corresponding frequency, you need to master the following building blocks.

Building Blocks	Description
	Plays a note. The default value plays middle C for 1 beat. The parameter indicates the specific note to play.
	Set the tempo of the music. For example, here, the value 120 means there are 120 beats per minute.

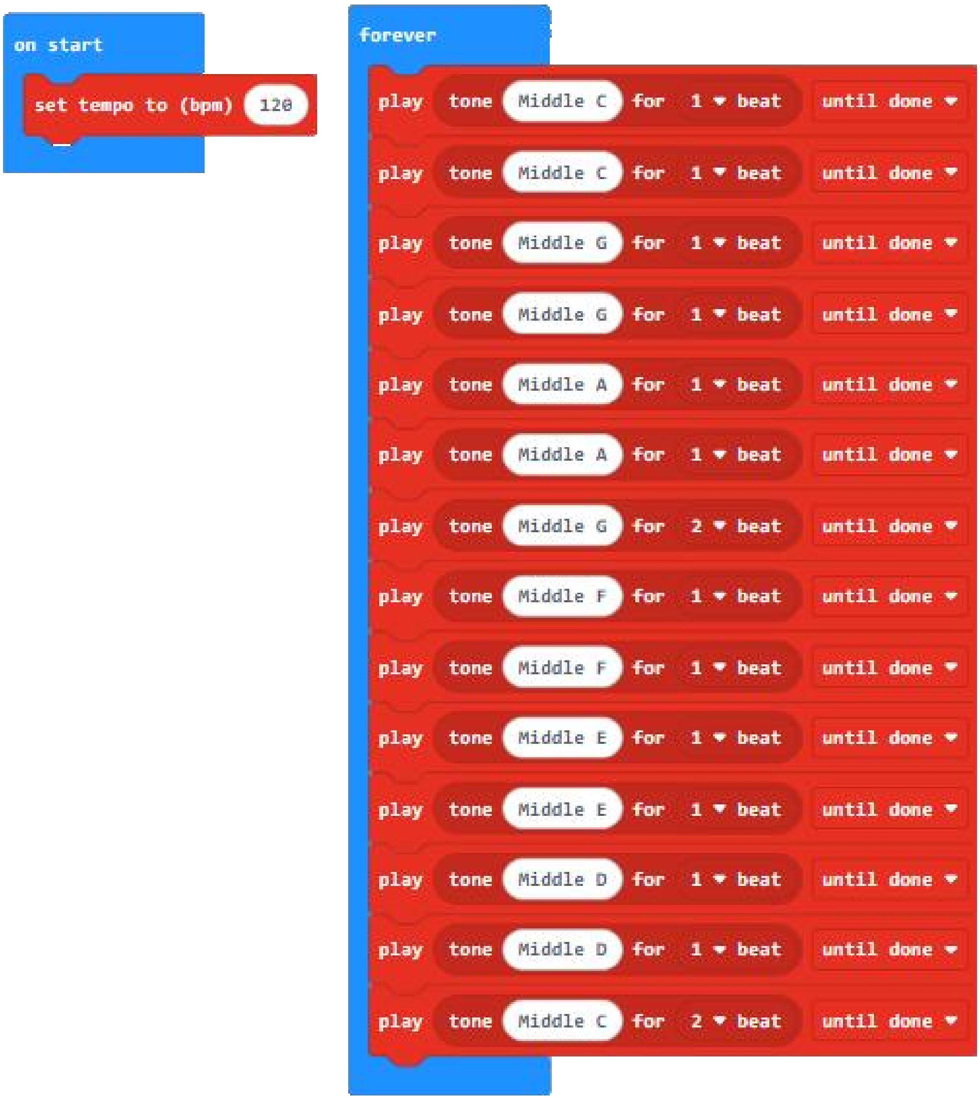
	Insert a 1-beat silence (rest)

When you select the parameters of the note block to play, a piano icon and a frequency value that can be filled in will appear, indicating that you can play music according to a specific note frequency.



(2) Reference Program

[Click here to view the program "Lesson 3" for playing music with a buzzer.]



(3) Run the Program

After downloading the program to micro:bit, the buzzer will play the melody of the corresponding song in a loop.

As the music played, the name of the song came out. It was the melody of "Twinkle Twinkle Little Star". Then, Inspector Lumi entered the first letter of the song name in the combination lock. Suddenly, with a "click", the bookshelf slowly opened to reveal a secret passage. Inspector Lumi lowered his voice and said, "Be careful. The key clue to solving the puzzle may be hidden at the end of the passage. Let's go and have a look."

Lesson 4: Directional Lights

A topographic map is displayed on the wall of the secret passage, and the topography is the same as the path from the police station to the library. When you look at the dim secret passage, express your concerns: "Although we have mastered the topography of the passage, the light inside the passage is very dim and we can't see the road conditions clearly. Will there be any danger?"

Inspector Lumi replied: "The TinkerBott smart car is equipped with RGB lights, which can lead the way ahead, guide us through the lights, and help us eliminate danger."

Your next task is to implement the lighting system of TinkerBott's car. Before completing the task, you need to understand the working principle of RGB.

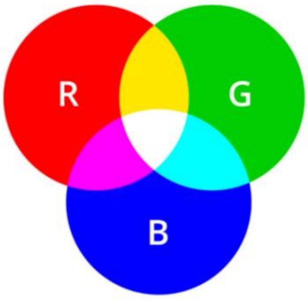
1. Knowledge Learning

1. RGB Lights

RGB lights are LED lights that can emit multiple colors of light. By adjusting the brightness ratio of the three primary colors of red (R), green (G), and blue (B), rich colors can be mixed. TinkerBott's two headlights are RGB lights, located on the left and right front of the car respectively.

(1) Working Principle of RGB Light

The color of RGB light is achieved by adjusting the brightness ratio of the three primary colors of red (R), green (G), and blue (B).



The brightness range of each color is 0-255. When the RGB value is 0, the corresponding color does not emit light. When the RGB value is 255, the corresponding color emits the brightest light. The combination of different color brightness can produce a variety of colors, for example:

- Red light (255, 0, 0)
- Blue light (0, 0, 255)
- Green light(0, 255, 0)
- White light (255, 255, 255)
- Yellow light(255, 255, 0)

By controlling the color of RGB lights, we can make the car's lights emit different prompt lights in different scenarios, such as turn signals, warning lights, etc.

(2) Hardware Parameters

Parameter	Description
Supply voltage	3.3-5V
Color Control	Full Color
Control Signal	Digital Signal

2. Mission Decryption

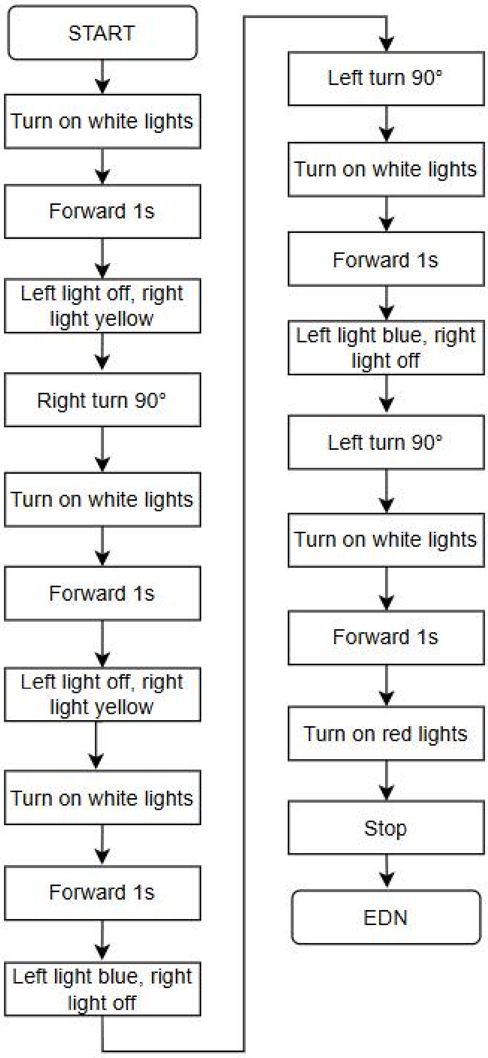
1. Task Description

Programming to implement TinkerBott's lighting system, the specific functions are as follows:

- When the car moves forward, both lights are on white.
- When the car turns left, the left headlight turns blue and the right headlight turns off.
- When the car turns right, the right headlight turns yellow and the left headlight turns off.
- When the car stops, both lights turn red.

2. Programming

(1) Program Flow Chart



(2) Command Learning

In the Micro:bit programming platform, we can use the following building blocks to control the RGB lights:

Building Blocks	Description
Set LED headlights Right ▾ color ●	Set the color of the RGB light. You can directly select the set color (16 colors in total)
Set the Right ▾ color R: 0 G: 0 B: 0	Set the color of the RGB light (specified by the brightness value of the R, G, B channels 0-255)

Through these building blocks, we can flexibly control the color changes of the lights and combine them with the movement of the car to achieve rich effects.

(3) Reference Program

[Click here to view the program "Lesson4" for the car light lighting effect.]


on start
Set LED headlights ALL color
Set direction Forward speed 50 %
pause (ms) 1000
Set LED headlights Right color
Set LED headlights Left color
Set direction Right speed 100 %
pause (ms) 400
Set LED headlights ALL color
Set direction Forward speed 50 %
pause (ms) 1000
Set LED headlights Right color
Set LED headlights Left color
Set direction Right speed 100 %
pause (ms) 400
Set LED headlights ALL color
Set direction Forward speed 50 %
pause (ms) 1000
Set LED headlights Left color
Set LED headlights Right color
Set direction Left speed 100 %
pause (ms) 400
Set LED headlights ALL color
Set direction Forward speed 50 %
pause (ms) 1000
Set LED headlights Left color
Set LED headlights Right color
Set direction Left speed 100 %
pause (ms) 400
Set LED headlights ALL color
Set direction Forward speed 50 %
pause (ms) 1000
Set LED headlights ALL color
Stop

(3) Run the Program

After downloading the program to the main control, turn the switch of the car to "ON" to start the car movement, and observe the driving path of the car and the changes of the headlights.

Forward

Turn right

Turn left

Stop

With the help of Tinker Bott, you and Inspector Lumi successfully walked out of the passage.

Lesson 5: Follow the Light, Tinker Bot!

When you and Inspector Lumi walked out of the passage, you saw a forest. The sun shone through the dense leaves and fell on the ground. Some places were bright, some were dim, and some areas were even completely covered by shadows. You looked at the forest in front of you and asked: "There are no road signs or guides here. Which direction should we go?"

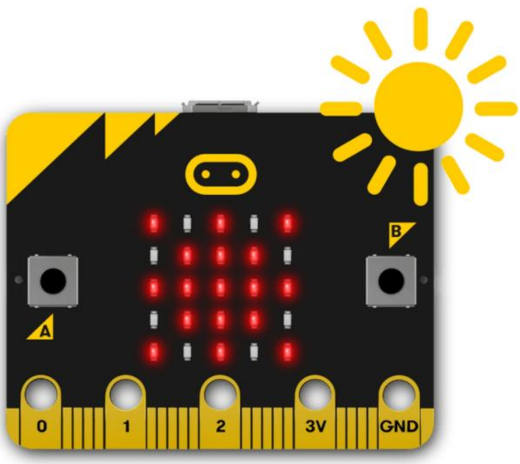
Inspector Lumi observed the surrounding environment, thought for a moment, and replied: "The light shining through the leaves seems to be guiding us somewhere. Let's go to the bright place first. Next, you need to let the TinkerBott smart car automatically follow the light source and continue to lead us forward."

Before you can complete the task, you need to understand how the TinkerBott smart car detects the surrounding light.

1. Knowledge Learning

1. Micro:bit Light Sensor

The light sensor of Micro:bit is not a dedicated photosensitive device, but uses the LED lights and receiving circuits in its LED matrix screen to work in reverse to achieve the light sensing function.



2. Micro:bit Light Detection Working Principle

A general light sensor is a device that can convert ambient light intensity into electrical signals, just like our eyes can perceive light and darkness. Its core principle is the **photoelectric effec**t: when light shines on the photosensitive element in the sensor (such as a photoresistor or photodiode), the element will change its resistance or current characteristics according to the intensity of the light, thereby outputting a corresponding electrical signal.

Micro:bit cleverly reuses the LED dot matrix screen to realize the light detection function. Its design inspiration is similar to that of general light sensors. It switches 25 LEDs to photodiode mode and uses the reverse current characteristics of LED to detect light. When light shines on the LED, photons excite the semiconductor to generate current, and the current intensity is proportional to the light intensity. The Micro:bit car can represent the detected light intensity value in the range of 0-255, and the larger the value, the stronger the light.

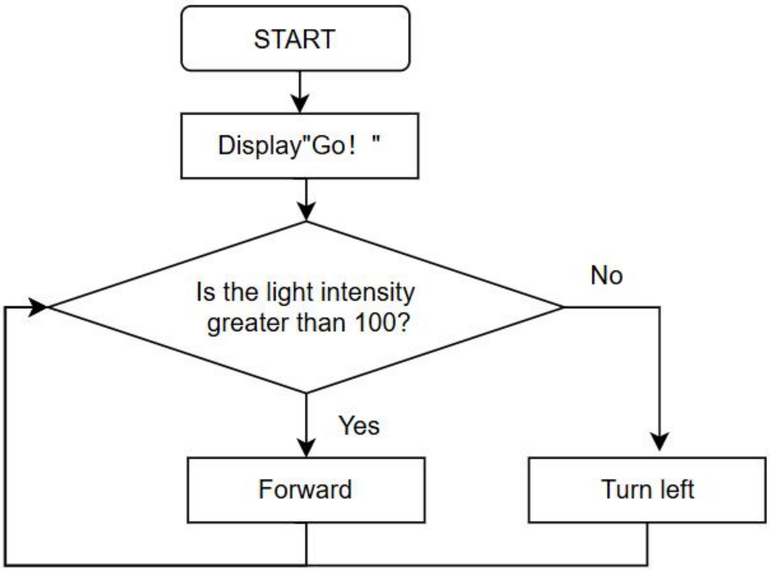
2. Mission Decryption

1. Task Description

Programming realizes the function of TinkerBott smart car to automatically find light. When the car does not detect the light source, the car rotates in place to find the light source. When the car detects the light source, the car automatically moves forward.

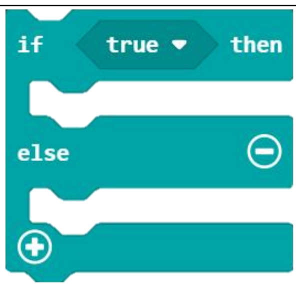
2. Programming

(1) Draw a Program Flow Chart



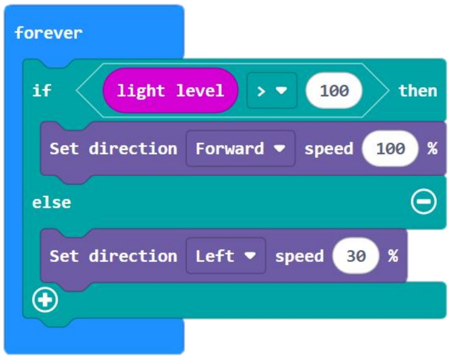
(2) Command Learning

In the Micro:bit programming platform, we can find the statement block for reading "light intensity" in the "Basic" instruction. In the "Logic" instruction, find the conditional judgment statement block. The block description is described in the table below.

Building Blocks	Description
	Read the value of light, the value range is 0-255
	**Branch statement block:** If the condition is true, the code block after then is executed; if the condition is false, the code block after else is executed.
	**Comparison operation instructions:** Used to compare the size of two numbers and return true or false based on the comparison result.

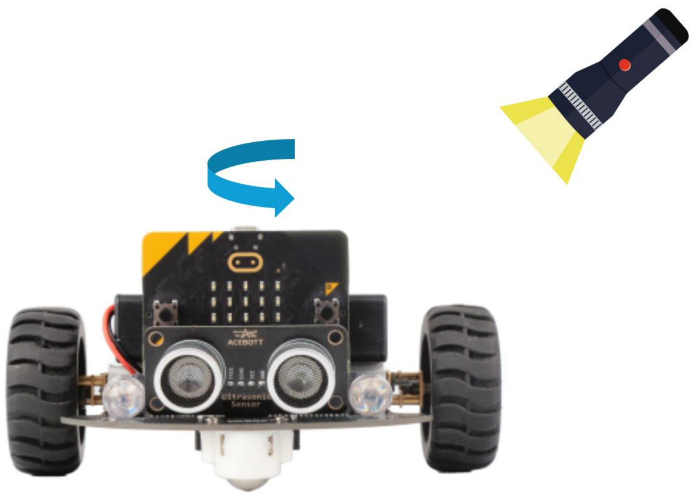
(3) Reference Program

[Click here to view the light sensor detection program "Lesson 5".]



(4) Run the Program

After downloading the program to the main control, in a relatively dark environment, use a mobile phone's flashlight or other light source to shine the light on the car and observe whether the car follows the light source.



With your help, the car leads Inspector Lumi to the brighter areas in the forest. What new problems will they encounter next?

Lesson 6: Sound Speed Controller

After passing through the brightest light spot in the forest, Inspector Lumi and the smart car came to the entrance of an abandoned mine tunnel. Suddenly, intermittent "beep-beep" sounds came from the depths of the mine tunnel, the rhythm was sometimes fast and sometimes slow, as if a machine was sending out signals. Inspector Lumi whispered: "These sound wave signals may point to secret devices deep in the mine tunnel. We need to let the car automatically adjust its speed according to the sound intensity so that we can approach quickly without missing every signal prompt."

Next, you need to understand sound signals and sound sensors. With the help of sound sensors, the car can learn to "adjust speed by listening to sounds", track mysterious signals, and reach the target directly!

1. Knowledge Learning

1. Learn about Sound

Sound is a sound wave generated by the vibration of an object, which can propagate in the air. The characteristics of sound are mainly as follows:

●**Amplitude**: The amplitude of a sound determines how loud it is. The greater the amplitude, the louder the sound.

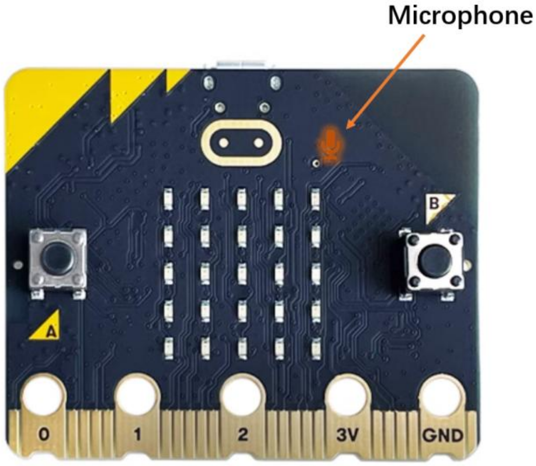
●**Frequency**: The frequency of a sound determines the pitch of the sound. The higher the frequency, the higher the pitch.

●**Waveform**: Different sounds have different waveforms, and the waveform reflects the complexity of the sound.

2. Sound Sensor

(1) Sound Sensor Hardware Description

A sound sensor is an electronic component that can sense the strength of ambient sound, and mainly detects changes in sound through a microphone component. Micro:bit has a built-in microphone sensor. It can respond to loud or quiet sounds, and can also measure the size of the sound around you.



(2) Working Principle of Sound Sensor

The working principle of sound sensors is based on the physical properties of sound. When the sound sensor detects sound, the sound waves contact the diaphragm of the microphone, causing the diaphragm to vibrate accordingly. This vibration is converted into an electrical signal, whose amplitude is related to the loudness of the sound, and whose frequency is related to the pitch of the sound. By analyzing these electrical signals, the sound sensor can determine the intensity, frequency and other characteristics of the sound, thereby realizing the detection and identification of the sound.

The microphone of the Micro:bit V2.0 controller can detect sounds in the range of **30-120dB** (covering the audible range of the human ear), and convert these signals into values of 0-255 to represent the intensity of the sound.

3. Sound Sensor Parameters

Parameter	Description
Supply voltage	3.3V-5V
Sound intensity range	30-120dB
Signal Type	Analog signal (0-255)

2. Mission Decryption

1. Task Description

Programming TinkerBott smart car to detect surrounding sounds:

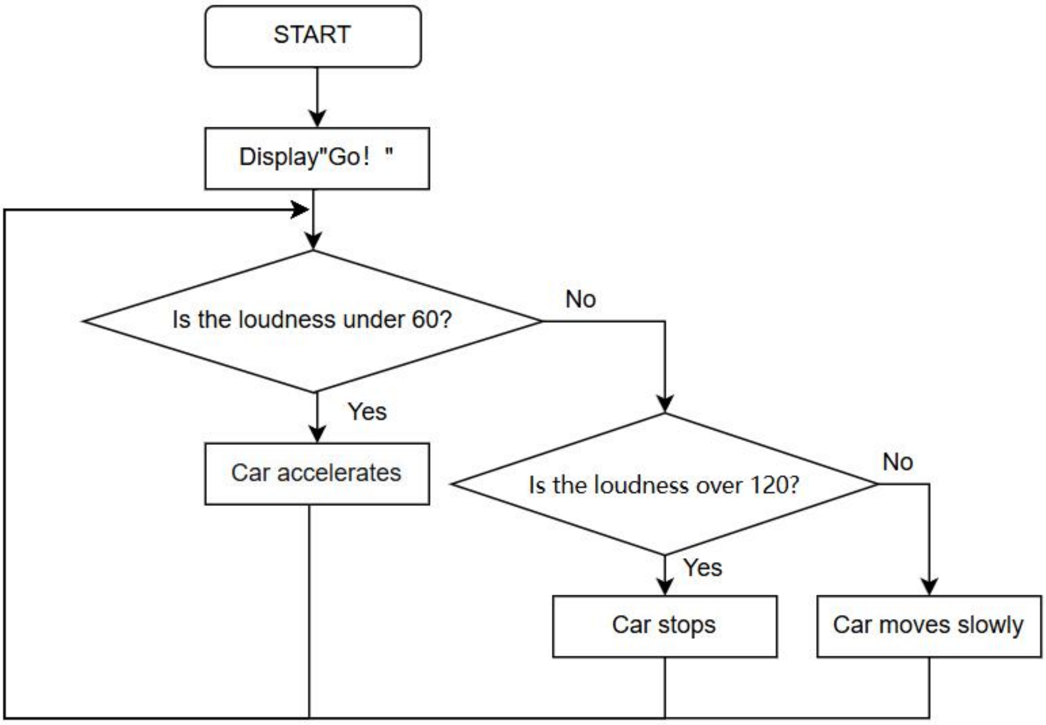
Sound < 60 (weak signal): The car accelerates (such as 60% PWM) and gradually approaches the sound source.

Sound 60–120 (medium signal): The car slows down (such as 30% PWM) and moves forward cautiously.

Sound > 120 (strong signal): The car stops and the screen shows that it has reached the vicinity of the sound source.

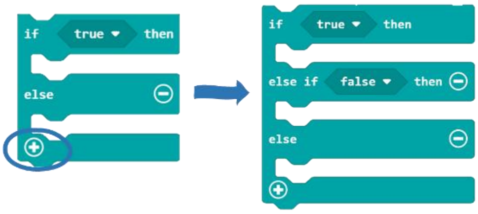
2. Programming

(1) Draw a Program Flow Chart



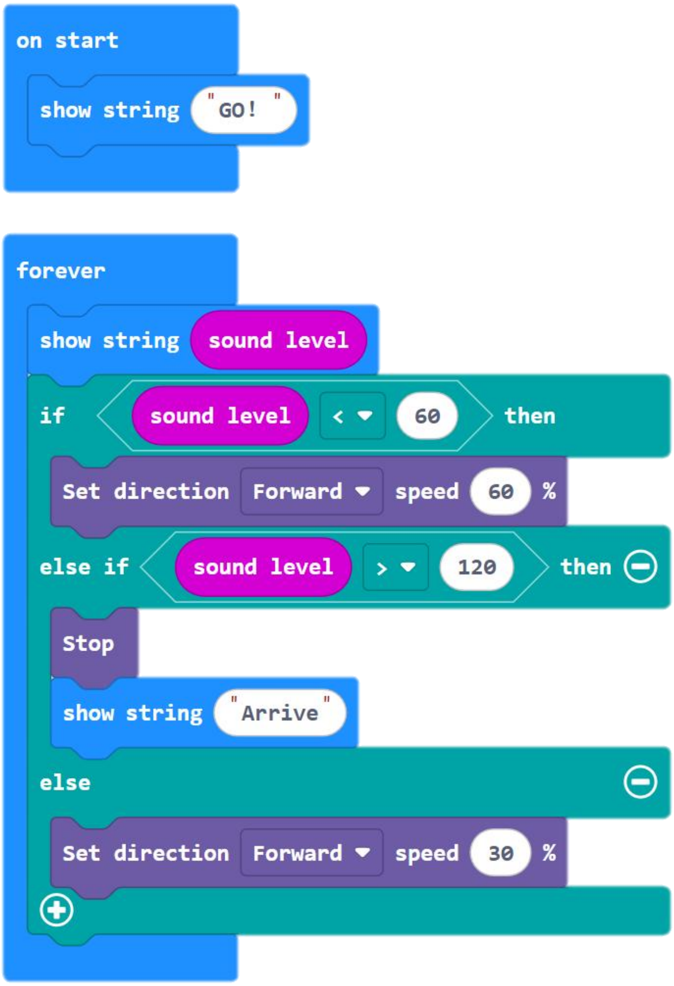
(2) Command Learning

In the Micro:bit programming platform, you can find the programming statements for the sound building blocks in the "Basic" instructions.

Building Blocks	Description
	Get the sound loudness value detected by the sound sensor, the value range is 0-255
	Multi-branch conditional statements: used to execute different code blocks according to different conditions. Find the branch statement block in the "Logic" block and click the "+" in the statement block to get the multi-branch conditional statement block.

(3) Reference Program

Click here to view the sound detection procedure "Lesson6".



(3) Run the Program

After downloading the program to micro:bit, turn on the power of the car and place it in a relatively quiet environment. The car will then accelerate. When you sing loudly towards the car, it will slow down. When the sound is loud enough, observe whether the car stops moving.

The car approached step by step according to the sound signal, and the car displayed "Arrive" and stopped moving. Inspector Lumi saw the device in front of him that made the sound. There was a button on it. When he pressed the button, the gate of the mine passage next to him suddenly opened. He approached the entrance of the passage and suddenly felt a heat wave coming towards him. Inspector Lumi stretched his hand close to the wall and shrank it as soon as he touched it. "The temperature here has obviously risen. This is definitely not normal. Maybe this is the next level of the puzzle."

Lesson 7: Temperature Perception

The hot air rushed towards him from deep in the mine tunnel. Inspector Lumi narrowed his eyes and took a few steps closer, saying, "The temperature ahead is abnormally high, so we can't move forward rashly. We need to let the TinkerBott smart car measure the surrounding temperature. When the temperature is too high, we need to stop and rest to prevent excessive loss of physical strength."

Next, you received a new task from Inspector Lumi. You need to let the TinkerBott smart car use the temperature sensor to detect the temperature changes in the environment and let the car make an alarm decision based on the abnormal ambient temperature.

1. Knowledge Learning

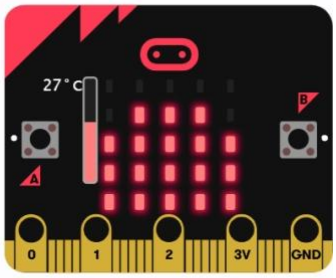
1. Temperature

Temperature is a physical quantity that measures the hotness or coldness of an object. It reflects the intensity of the molecular movement inside the object. The unit of temperature is usually expressed in degrees Celsius (℃) or degrees Fahrenheit (℉). From human body temperature measurement, weather to agricultural production, industrial production, to circuit boards, chips, fire alarms, etc., all aspects cannot be separated from temperature measurement and control.

2. Temperature Sensor

(1) Temperature Sensor

A temperature sensor is an electronic component that can detect ambient temperature. The Micro:bit main control board comes with a temperature sensor that can measure the ambient temperature in real time and output the temperature value in digital form (in degrees Celsius).



(2) Working Principle of Temperature Sensor

The temperature sensor of Micro:bit senses the ambient temperature by detecting the temperature of the main control board chip. The temperature sensor senses the thermal motion of air molecules in the environment and converts it into electrical signals. By processing the electrical signals, the temperature value can be output in the form of degrees Celsius. These temperature data can be read in real time and used for programming operations.

3. Temperature Sensor Parameters

Parameter	Description
Signal Type	Digital Signal
Measuring range	-5℃ ~ 50℃
Operating voltage	3.3V

2. Mission Decryption

1. Task Description

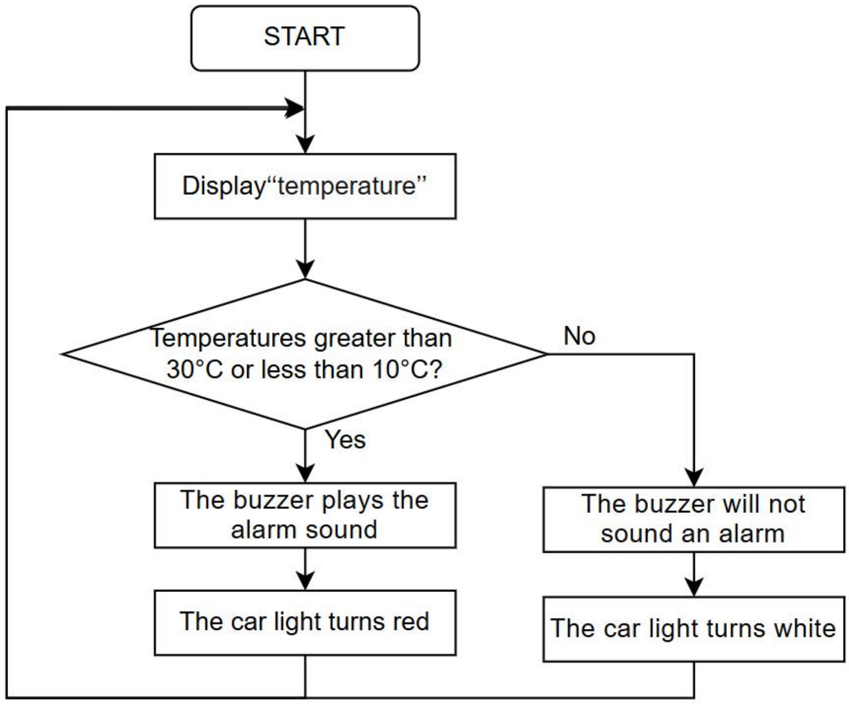
The TinkerBott smart car can be programmed to take different actions according to the changes in ambient temperature. The specific functions are as follows:

- When the ambient temperature is abnormal (such as greater than 30℃ or less than 10℃), the car lights up the red light and sounds a buzzer alarm.
- When the ambient temperature is within the normal range (not within the above range), the car lights remain bright white and the buzzer does not sound.

2. Programming

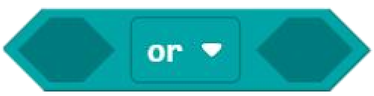
(1) Draw a Program Flow Chart

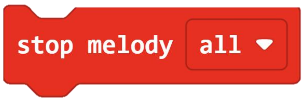
According to the task requirements, you need to design a program that allows the car to monitor the input of the temperature sensor in real time and respond accordingly based on the temperature value. The following is the flow chart of the program.



(2) Command Learning

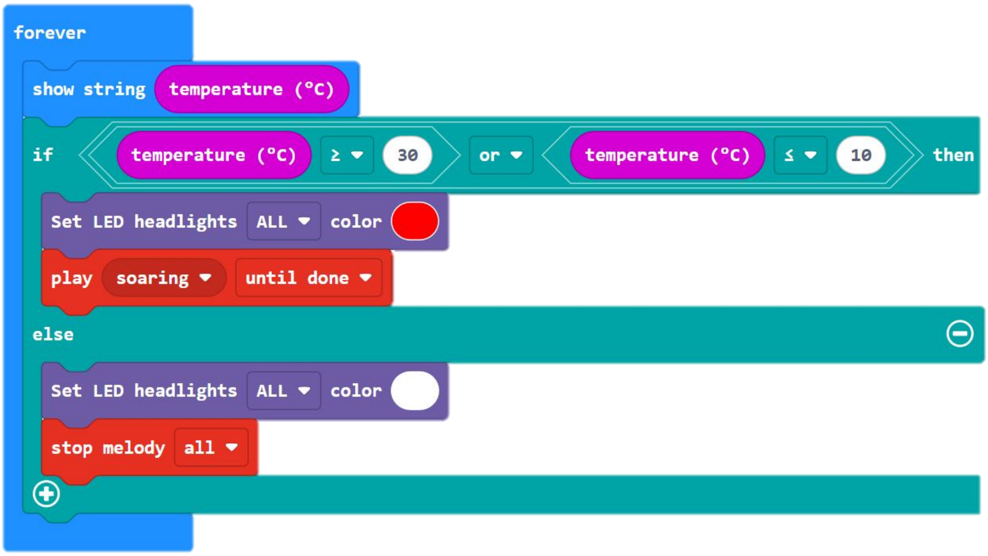
In the Micro:bit programming platform, you can find the building blocks for obtaining temperature in "Basic".

Building Blocks	Description
 	Get the current temperature value (unit: ℃)
 	Logical operators: connect two or more conditions, optional parameters are **and** and **or** **and statement block:** The result of the entire expression is true only when all given conditions are true. **or statement block:** As long as any one of

	the given conditions is true, the result of the entire expression is true.
	Play music, the parameter is to select the melody to play
	Stop music playing. The parameter means to stop all the melodies being played (you can also stop a specific melody separately).

(3) Reference Program

Click here to view the program "Lesson7" for the car temperature detection.



(4) Run the Program

You need to adjust the temperature parameters above according to the ambient temperature of your location. After downloading the program to the main controller, turn on the power switch of the car, the main controller will display the initial temperature, and observe the movement of the car. During the test, you can use a fan to cool the main controller or warm it up with your hands.

With your help, the car successfully avoided the area with abnormal temperature. However, as the exploration went further and further, Inspector Lumi had lost his way in the target area, and new challenges were coming one after another.

Lesson 8: Finding Direction

As the route progressed, Inspector Lumi suddenly realized that they seemed to have been to this place before. Inspector Lumi stopped, carefully observed the surrounding environment, and then said: "There is a lack of obvious references and guidance here, and we are obviously lost. But don't worry, we can rely on the TinkerBott smart car to take us out of here. It is equipped with a compass to help us determine the direction." Next, Inspector Lumi needs your help to implement the compass function in TinkerBott.

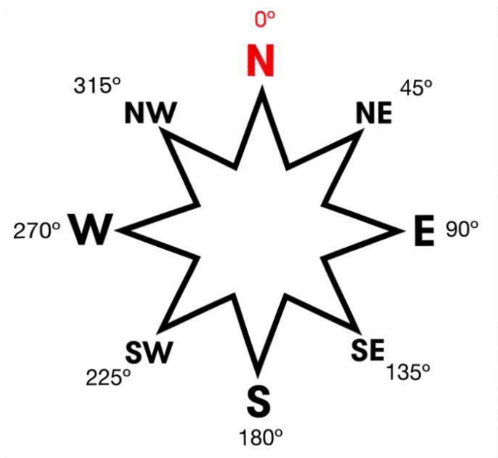
Before completing the task, you need to understand the knowledge related to the compass.

1. Knowledge Learning

1. Compass

A compass is a navigation tool that works based on the principle of the earth's magnetic field. The direction is determined by pointing the magnetic needle to the earth's magnetic north pole. In Micro:bit, the compass sensor can detect the direction of the device relative to the earth's magnetic field and output the result in the form of an angle.

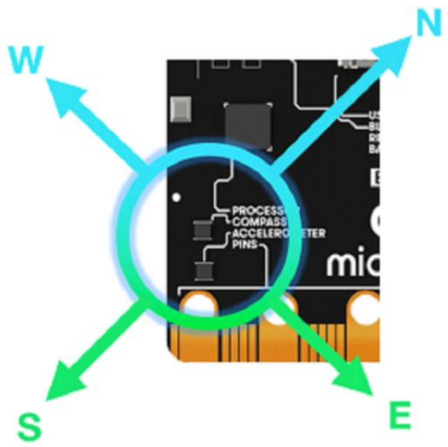
Divide the circumference into four equal 90° intervals, with each main direction (N/E/S/W) occupying a 90° range. With due north (0°) as the starting point, the four directions correspond to:



2. Compass Sensor

(1) Compass Sensor

The Micro:bit main controller board has a built-in electronic compass (magnetometer) sensor, which can accurately determine the direction according to the earth's magnetic field (depending on the initialization situation), detect the direction of the car and output the angle value of 0° ~ 360°. The module needs to be initialized before each call.



(2) Working Principle of Compass Sensor

The three-axis magnetometer inside the compass sensor senses the strength and direction of the Earth's magnetic field, and calculates the angle of the device relative to the Earth's magnetic north direction by processing the magnetic field data. It can also measure the direction of the device within a 360-degree range and convert this direction into an angle value (0-360 degrees), where 0 degrees represents due north, 90° represents due east, 180° represents due south, and 270° represents due west.

3. Compass Sensor Parameters

Parameter	Description
Communication	I2C
Accuracy	±5°
Output Units	angle

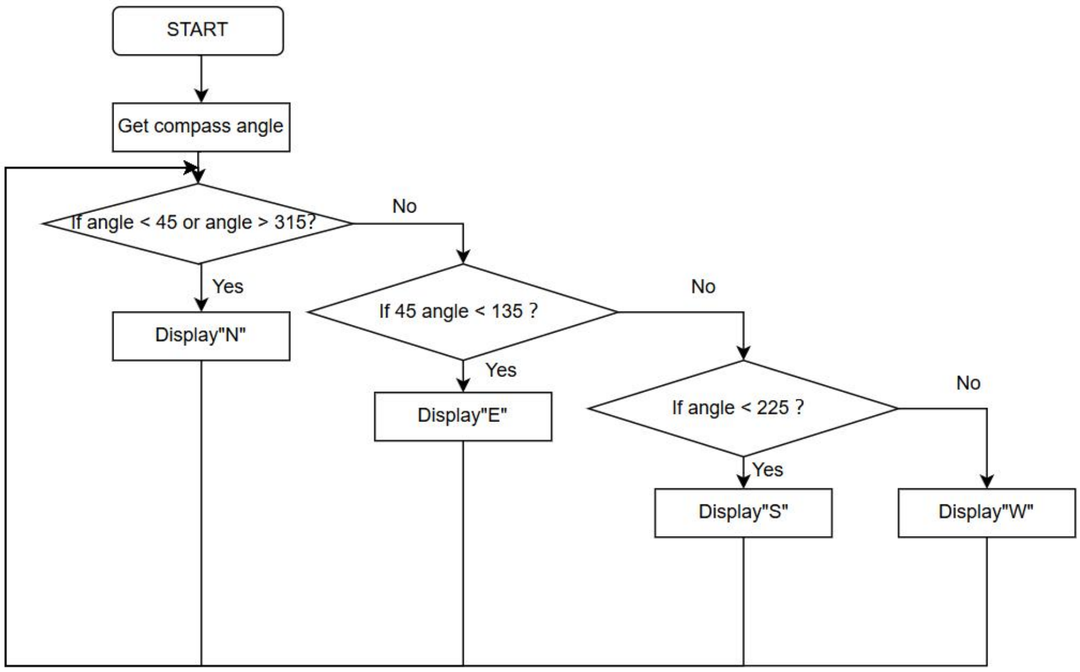
2. Mission Decryption

1. Task Description

Program the TinkerBott smart car to display its current position on the screen, and indicate the position of the car by displaying "N", "E", "S", and "W".

2. Programming

(1) Draw a Program Flow Chart



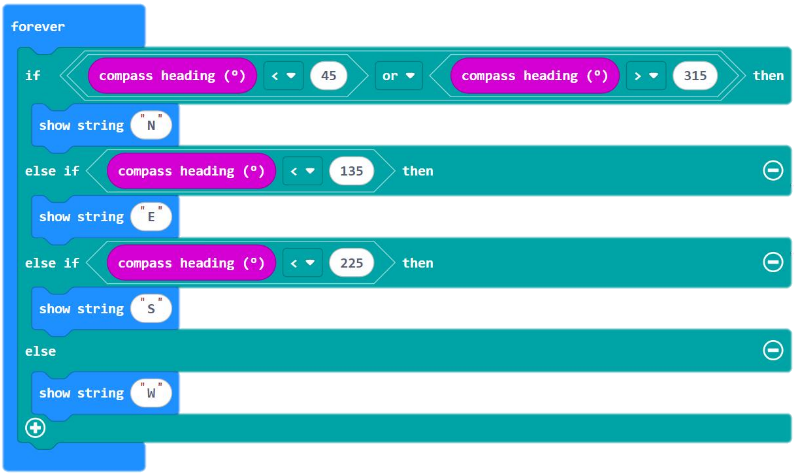
(2) Command Learning

Before implementing this task, you need to find the compass direction block in the "Basic" instruction.

Building Blocks	Description
compass heading (°)	Call the compass sensor to measure and return the current heading angle in degrees (°)

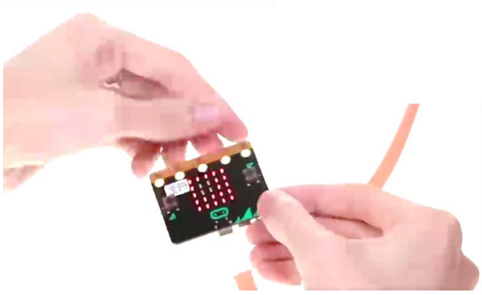
(3) Reference Program

Click here to view the program "Lesson8" for the car compass display.



(4) Run the Program

Take the main control off the car, connect the main control to the computer with a data cable, and burn the program of this lesson to the micro:bit main control. When running for the first time, the screen will display the English initialization prompt "TILT TO FILL SCREEN". Then some bright spots will appear on the screen. You need to flip the Micro:bit main control board back and forth and use the gravity sensor to light up all the LED lights. This step is to calibrate the compass of the main control. After all lights are lit, a smiley face pattern will be displayed, and then all LED lights will go out, which means the calibration is complete. Next, you will see the main control screen showing the current direction.



With your help, TinkerBotth can now use the compass sensor to detect the current direction and adjust its actions according to the direction. Inspector Lumi realized that the environment here is far more complicated than he imagined. If the car faces multiple challenges such as insufficient light, abnormal temperature, and sound interference while driving here, how will it coordinate the data from various sensors and make the best response?

Lesson 9: Multi-Sensor Collaboration

Detective Lumi used the car's compass sensor to find the direction and continued to move forward. But he soon found that the surrounding environment was becoming more and more complex, the light was getting darker, the temperature was fluctuating, and there were all kinds of noisy sound waves. Detective Lumi said: "Our car has learned how to use a single sensor to deal with a specific environment, but in this complex target area, a single capability may not meet the needs. We need to let the car integrate the functions of all sensors and make comprehensive judgments based on multiple environmental factors like a real detective assistant."

That’s right, our next task is to teach the car to integrate and use a variety of sensors (light, temperature, sound) and actuators (motors, speakers, LED dot matrix screens) so that it can respond flexibly in complex environments.

1. Knowledge Learning

1. Concept of Multi-Sensor Application

Multi-sensor application refers to the integration and analysis of data from multiple sensors to make more accurate and intelligent decisions in complex environments. In practical applications, different sensors can make up for the limitations of a single sensor, and integrating data from multiple sensors can allow the car to respond more intelligently in complex scenarios.

2. Variables

In programming, variables are containers used to store data. You can store the values detected by sensors in variables to facilitate subsequent processing and judgment. For example, you can create a variable to store the brightness value detected by a light sensor, another variable to store the temperature value detected by a temperature sensor, and so on. Through variables, you can easily calculate and compare these data, thereby achieving a comprehensive judgment of the environment.

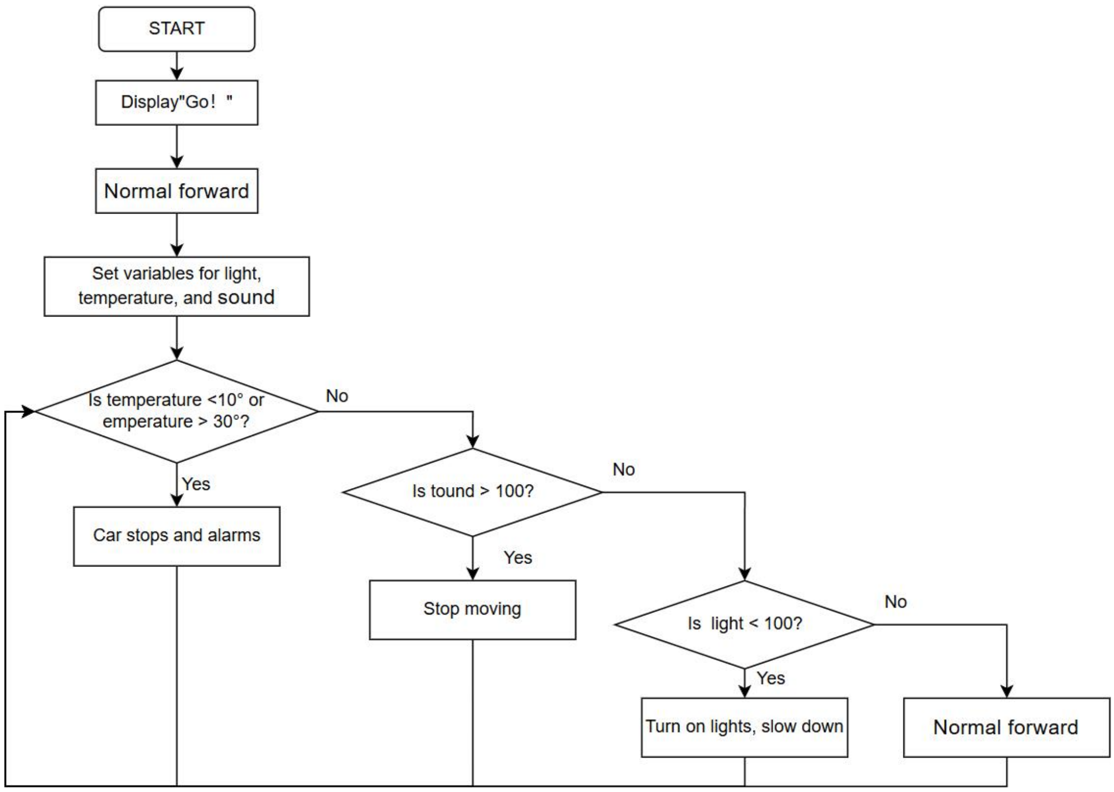
2. Mission Decryption

1. Task Description

Programming realizes the TinkerBott smart car to integrate the functions of multiple sensors and respond intelligently according to environmental changes. In abnormal environmental conditions, when the temperature sensor detects that the temperature is greater than 30° or less than 10°, the car can sound an alarm and stop moving. When the sound sensor detects that the sound interference is too large, the car can stop moving. When the light sensor detects that there is insufficient light, the car can automatically turn on the lights and slow down.

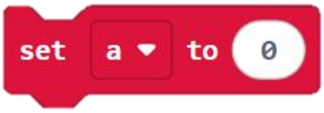
2. Programming

(1) Draw a Program Flow Chart



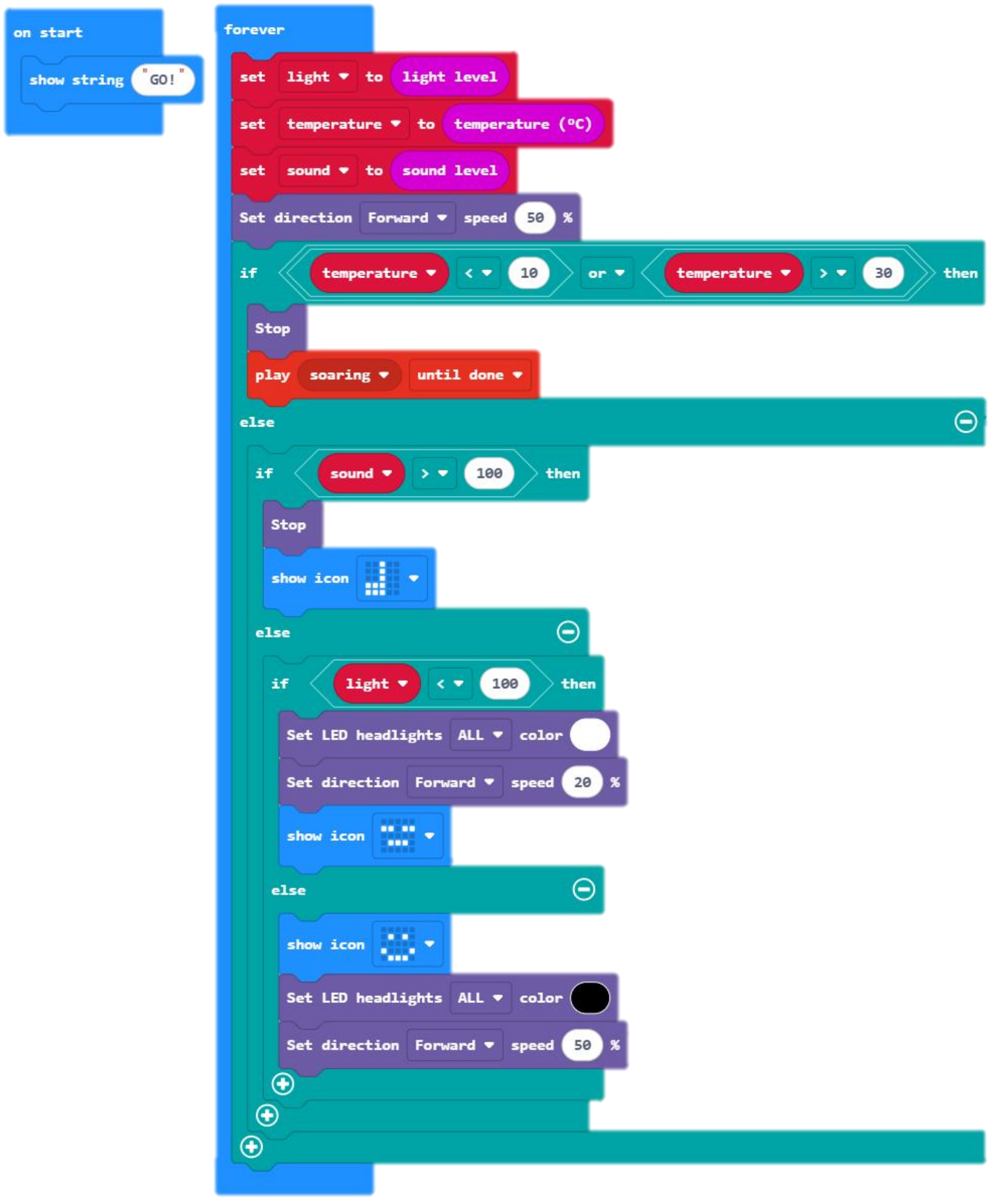
(2) Command Learning

In this task, there are three sensor detection values: temperature, sound loudness, and light intensity. These values will change with the environment. You can use variables to store the values detected by the sensors. In the programming platform, find the "Basic" instruction. Next, you can use variables according to the following methods and statement blocks.

Building Blocks	Description
	Click "Make a Variable" to create one or more variables. You need to name different variables. For example, the variable name is "a". After setting the variable, you can assign values and perform operations on the variable.
	Get or manipulate the current value of the variable
	Set the value of a variable to a specific number
	Increase or decrease the current value of the variable by the specified value, for example, increase the value of variable a by 1.

(3) Reference Program

Click here to view the program "Lesson9" for multi-sensor control of the car.



(4) Run the Program

After downloading the program to the main control board, turn on the power of the car, the car will move forward at 50% speed and display "GO!". Please follow the following process to test:

① **Temperature response tes**t: Cover the temperature sensor with your hand, and check whether the car stops moving and plays the "Soaring" music when the temperature is >30℃;

② **Sound response test**: In a quiet environment, suddenly shout loudly (or play music on your phone and turn up the volume), and observe whether the car stops immediately and displays the corresponding icon;

③ **Light response test**: Use a flashlight to illuminate the light sensor. When the brightness is ≥100, the car drives at normal speed. After removing the light source, the lights turn white and the speed drops to 20%. Press the reset button after each test to reset the status.

You test the car excitedly: "Now the TinkerBott smart car can deal with problems such as light, temperature and sound at the same time!" Led by the TinkerBott smart car, you and Inspector Lumi continue to move forward to find the truth.

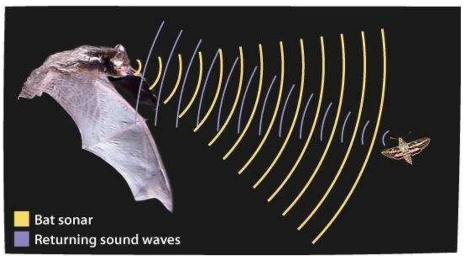
Lesson 10: Intelligent Obstacle Avoidance

You and Inspector Lumi finally walked out of the forest. A rocky area appeared in front of you. You found a suspicious person moving in the rocky area ahead. You need to follow this suspicious person, but the rocks everywhere block the progress of TinkerBott. You need to let TinkerBott automatically avoid these rocks to pass through this rocky area. Inspector Lumi thought for a while and said: "TinkerBott has an ultrasonic sensor that can help the car avoid these rocks. Next, you will implement the obstacle avoidance function of TinkerBott." Before completing the task, you need to understand the knowledge related to ultrasound.

1. Knowledge Learning

1. Ultrasonic waves

Ultrasonic waves are sound waves with a frequency higher than the human hearing range (usually more than 20kHz). They propagate in the air, water or solids in the form of waves. Due to its strong directionality, strong penetration and concentrated energy, ultrasonic waves are often used in medical examinations (such as B-ultrasound), industrial inspections, distance measurement, automatic obstacle avoidance and other fields. In daily life, many smart devices also use ultrasonic waves to achieve sensing and control functions.



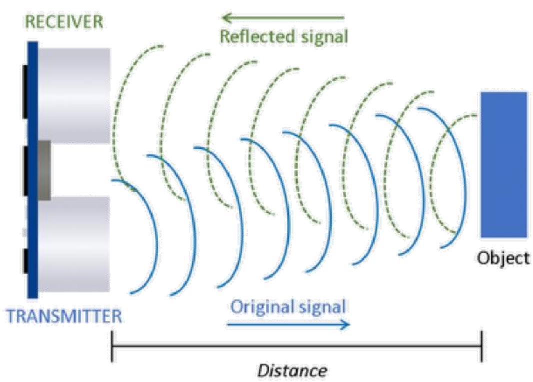
2. Ultrasonic sensor

(1) Ultrasonic Sensor

An ultrasonic sensor is an electronic device that uses ultrasonic waves to measure the distance of an object or detect the presence of an object. It emits high-frequency ultrasonic signals, and when these sound waves encounter obstacles, they are reflected back. After the sensor receives the echo, it calculates the distance to the object based on the propagation time of the sound waves. Ultrasonic sensors are widely used in scenarios such as automatic obstacle avoidance, smart parking, robot navigation, and liquid level detection. They are a common and practical intelligent sensing device.

(2) Principle of Ultrasonic Ranging

The ultrasonic sensor consists of a transmitter and a receiver. The sensor sends ultrasonic pulses through the Trig pin and receives echoes through the Echo pin. The distance can be calculated based on the round-trip time of the ultrasonic wave. The distance formula is: distance = speed of sound × time/2 (the propagation speed of ultrasonic waves in the air is about 340 m/s)



(3) Hardware parameters of ultrasonic sensor

Parameter	Description
Operating voltage	3.3V-5V
Measuring range	3cm ～500cm
Measurement accuracy	±0.1cm
Measurement blind area	<3cm

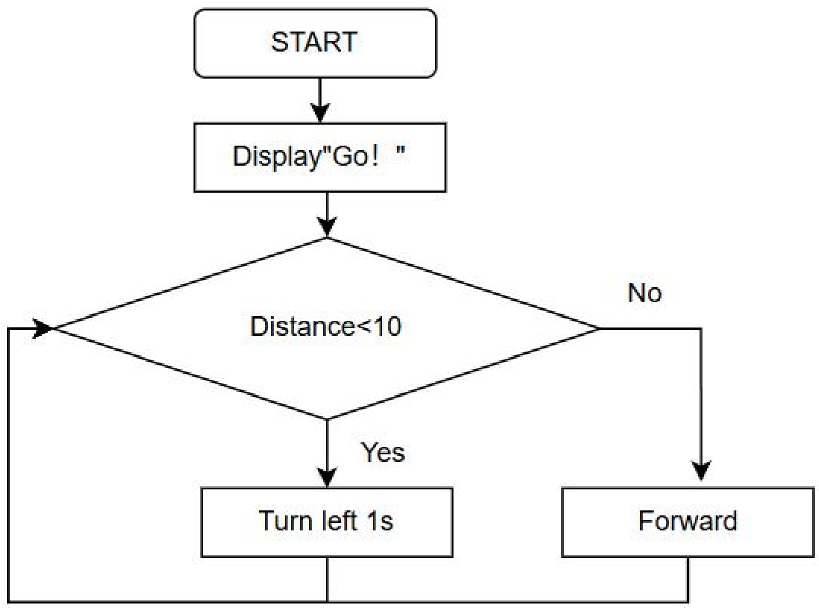
2. Mission Decryption

1. Task Description

Program to implement TinkerBott's automatic obstacle avoidance function. When an obstacle appears within 10cm in front of it, it will automatically avoid the obstacle.

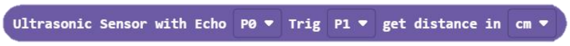
2. Programming

(1) Draw a Program Flow Chart



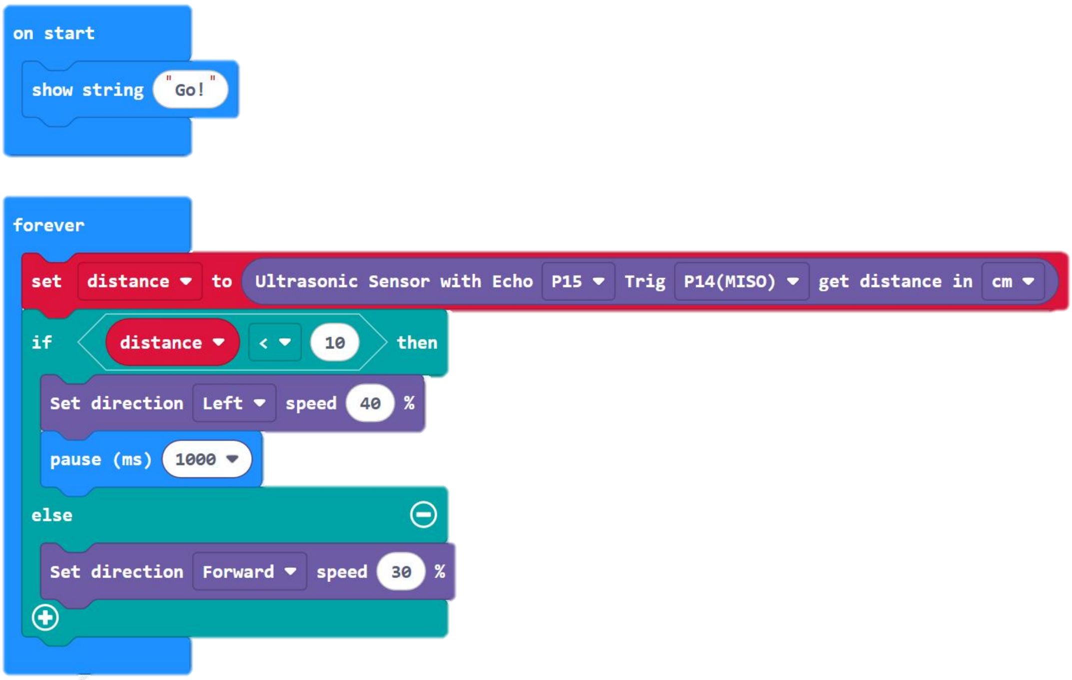
(2) Command Learning

In the Micro:bit programming platform, find the Acebott car expansion package, and in the "Sensor" command, find the block for obtaining the distance value measured by the ultrasonic sensor.

Building Blocks	Description
	Read the distance value measured by the ultrasonic sensor. The parameter Echo pin should be changed to P15, the Trig pin to P14, and the measurement unit can be cm or inch, so as to realize the distance detection of the ultrasonic sensor.

(3) Reference Program

[Click here to view the program "Lesson 10" for the car to avoid obstacles.]



(4) Run the Program

After downloading the program to the main control, turn on the power of the car, cover the ultrasonic sensor in front of the car with your hand, and observe whether the car can turn left. After removing your hand, observe whether the car can move forward normally.

With your help, Tinker Bott successfully crossed the rocky area. Now you have to follow the suspicious person in front to see where he is going.

Lesson 11: Intelligent Following

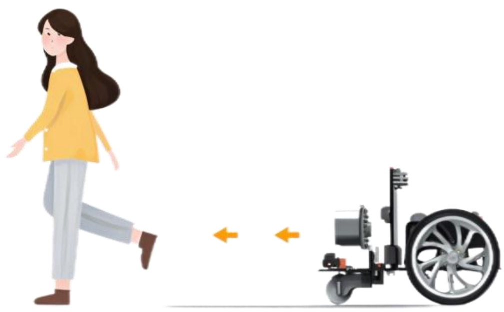
After passing through the rocky area with Detective Lumi, you were ready to quietly follow the suspicious figure ahead. Suddenly, you noticed that the suspicious figure was acting strangely, constantly stopping and starting as if checking if they were being followed. To avoid being detected by the person ahead, Detective Lumi decided to maintain a certain distance while following and said to you, "To prevent the person ahead from discovering us, let's have the TinkerBott robot follow him."

Next, you need to help Inspector Lumi TinkerBott give the car the ability to follow intelligently, so that it always keeps a certain distance from the target.

1. Knowledge Learning

1. Ultrasonic Distance Tracking

In order to achieve fixed-distance tracking, the ultrasonic sensor must continuously measure the distance to the target in front. The car compares this distance with the set target distance and adjusts the distance between itself and the target by controlling the speed of the motor. If the actual distance is less than the lower limit of the target distance, the car will slow down or retreat appropriately; if the actual distance is greater than the upper limit of the target distance, the car will accelerate forward to always keep a certain distance between the car and the moving target.



2. Application Scenarios of Fixed-Distance Tracking Technology

Fixed-distance tracking technology has a wide range of applications in real life. For example, in the field of intelligent transportation, the adaptive cruise control (ACC) system allows the car to automatically maintain a safe distance from the vehicle in front; in warehousing and logistics, AGV (automatic guided vehicle) can follow the movement of goods or staff to ensure a fixed distance; in the field of drones, drones use fixed-distance tracking technology to achieve stable following of moving targets (such as vehicles or people) and maintain a safe flight distance; in addition, service robots (such as restaurant delivery robots) can also use this technology to follow waiters or customers to avoid collisions. These applications all reflect the practicality and intelligent characteristics of fixed-distance tracking technology.

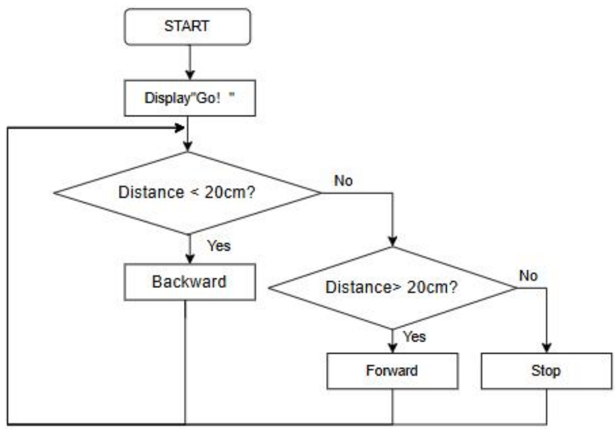
2. Mission Decryption

1. Task Description

Program the TinkerBott smart car to always keep a distance of 20 cm from the target object in front of it during driving.

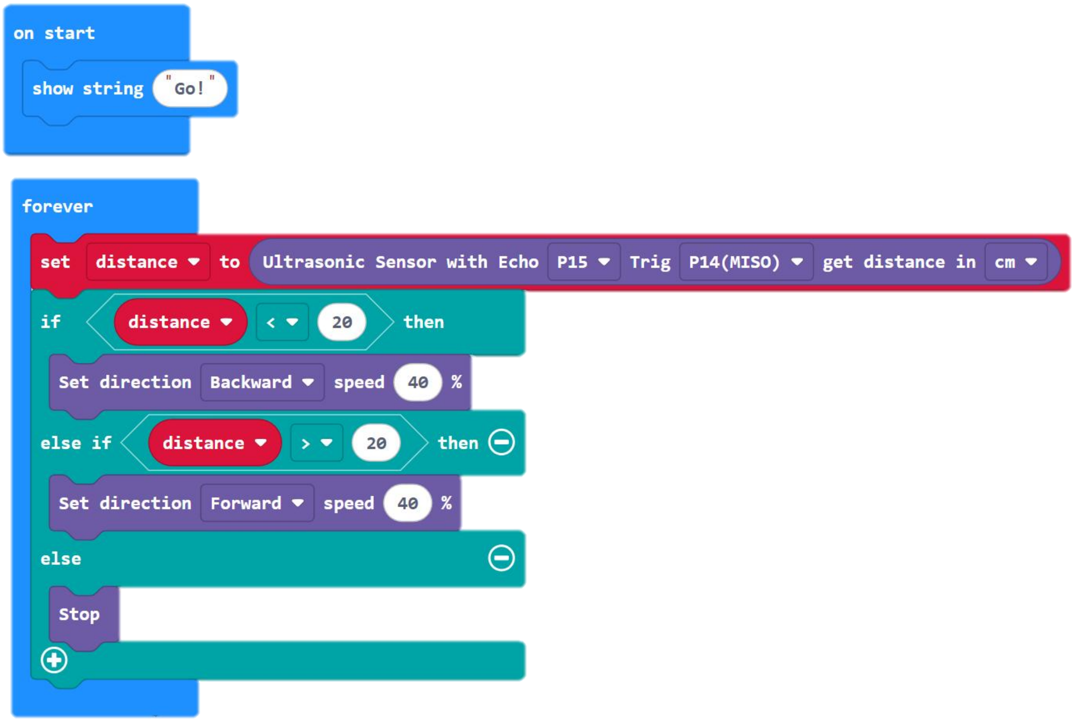
2. Programming

(1) Draw a Program Flow Chart



(2) Reference Program

Click here to view the program "Lesson11" for the car to follow the movement.



(3) Run the Program

After downloading the program to the main controller board, turn on the power of the robot. It is recommended that you use a flat object, such as a book, and place the book in front of the ultrasonic wave. By slowly translating the book, observe whether the robot can follow the book forward or backward.

Lumi Detective's car can now successfully keep up with the target vehicle and keep a certain distance. After a while, we entered a canyon with abysses on both sides of the road...

Lesson 12: Pull Back from the Brink

Walking on the cliff road is very dangerous. There are many sharp turns on the road. It is easy for the TinkerBott smart car to fall off the cliff. Inspector Lumi decided to take protective measures to prevent the car from falling off the cliff. After some thought, Inspector Lumi said: "If the car can recognize the black boundary of the cliff, it can avoid falling off. TinkerBott has a grayscale sensor that can be used to detect dark black lines and thus detect the cliff boundary."

Next, your task is to help TinkerBott identify the black boundary. Before completing the task, you need to understand the relevant knowledge of line-following sensors.

1. Knowledge Learning

1. Line Patrol Sensor

The line-following sensor is an electronic component used to detect color differences. It can judge the depth of color by sensing the intensity of light reflected from the surface of an object. The line-following sensor module of the Micro:bit car contains two probes (one on each side), and the hardware installation position is shown in the figure below.

2. Working Principle of Line Patrol Sensor

The line patrol sensor transmits infrared light to the ground through the infrared transmitting tube, and the ground will reflect some of the light. The line patrol sensor converts the received light intensity into grayscale value. Black absorbs more infrared light and reflects less, so the return value is high; white absorbs less infrared light and reflects most of it, so the return value is low. The line patrol sensor of the car is connected to P0 and P1 of the Micro:bit main control board. You can get it by reading the analog values of P0 and P1.

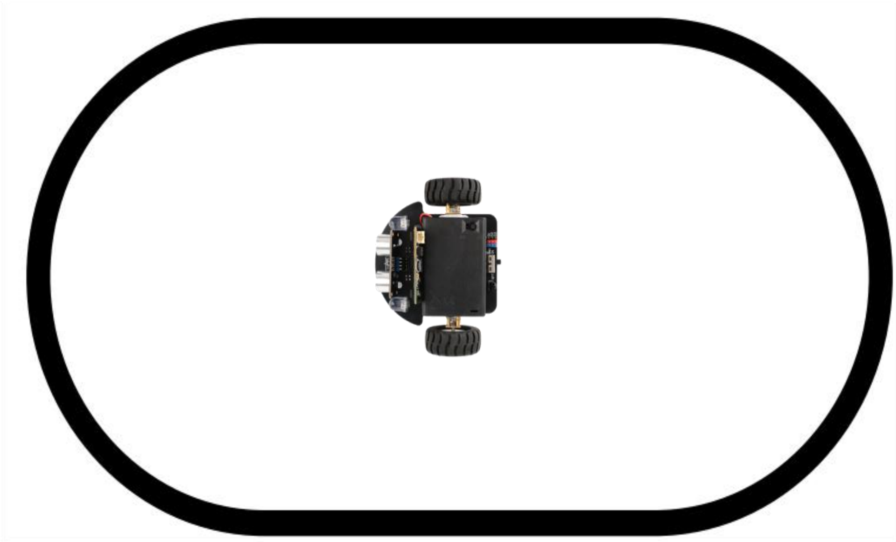
3. Parameters of Line Patrol Sensor

Parameter	Description
Output Type	Analog signal
Detecting Color	Black (high reflectivity), white (low reflectivity)
Operating voltage	3.3V
Output value range	0 ~ 1023 (higher values mean less reflected light, lower values mean more reflected light)
Pin Description	P0—right grayscale sensor; P1—left grayscale sensor

2. Mission Decryption

1. Task Description

Let the TinkerBott smart car move in the black circle. To complete this task, you need to first let the car automatically recognize the black boundary line, and then control the car to avoid the black boundary line. As shown in the figure below:



2. Programming

(1) Task 1: Test the grayscale value range of the black line.

Use micro:bit to control the TinkerBott smart car to read the data of the line patrol sensor in the black line and white paper areas respectively, and record the grayscale value range to prepare for the subsequent identification of the black line boundary.

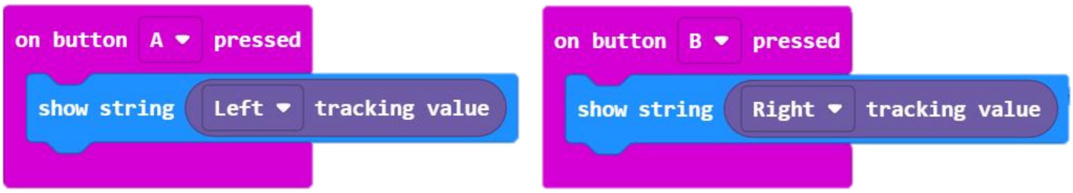
① Instruction Learning

In the Micro:bit programming platform, find the Acebott car expansion package, and in the "Executive" command, find the building block for obtaining the grayscale value of the line patrol sensor.

Building Blocks	Description
	Read the grayscale value measured by the line patrol sensor. This block can be used to select and read the grayscale value from either the left or right probe of the line patrol sensor.

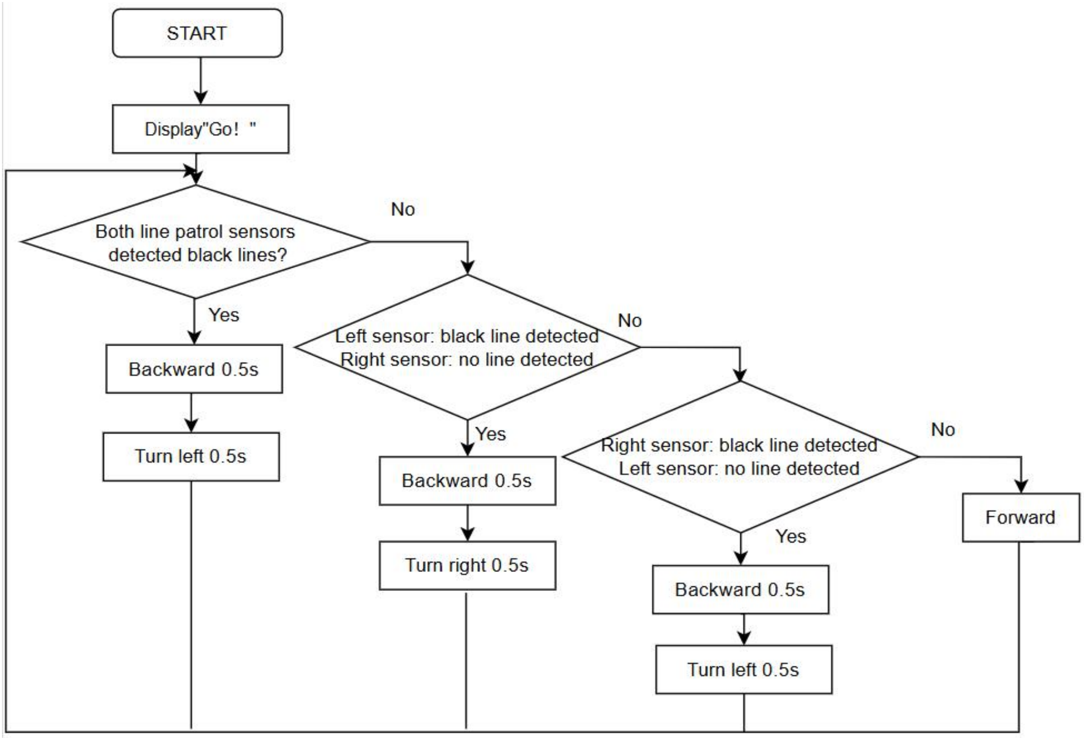
② Measure the grayscale value of the sensor probe on the black line

You need to measure the grayscale values of the left and right probes of the TinkerBott smart car in the black line area respectively, and use the buttons and LED dot matrix screen to obtain the values multiple times. According to the test results, determine a grayscale threshold that just detects the black line (for example: threshold = 800). You can refer to the following program to obtain the values. Remember to read the grayscale value multiple times and record it on paper!



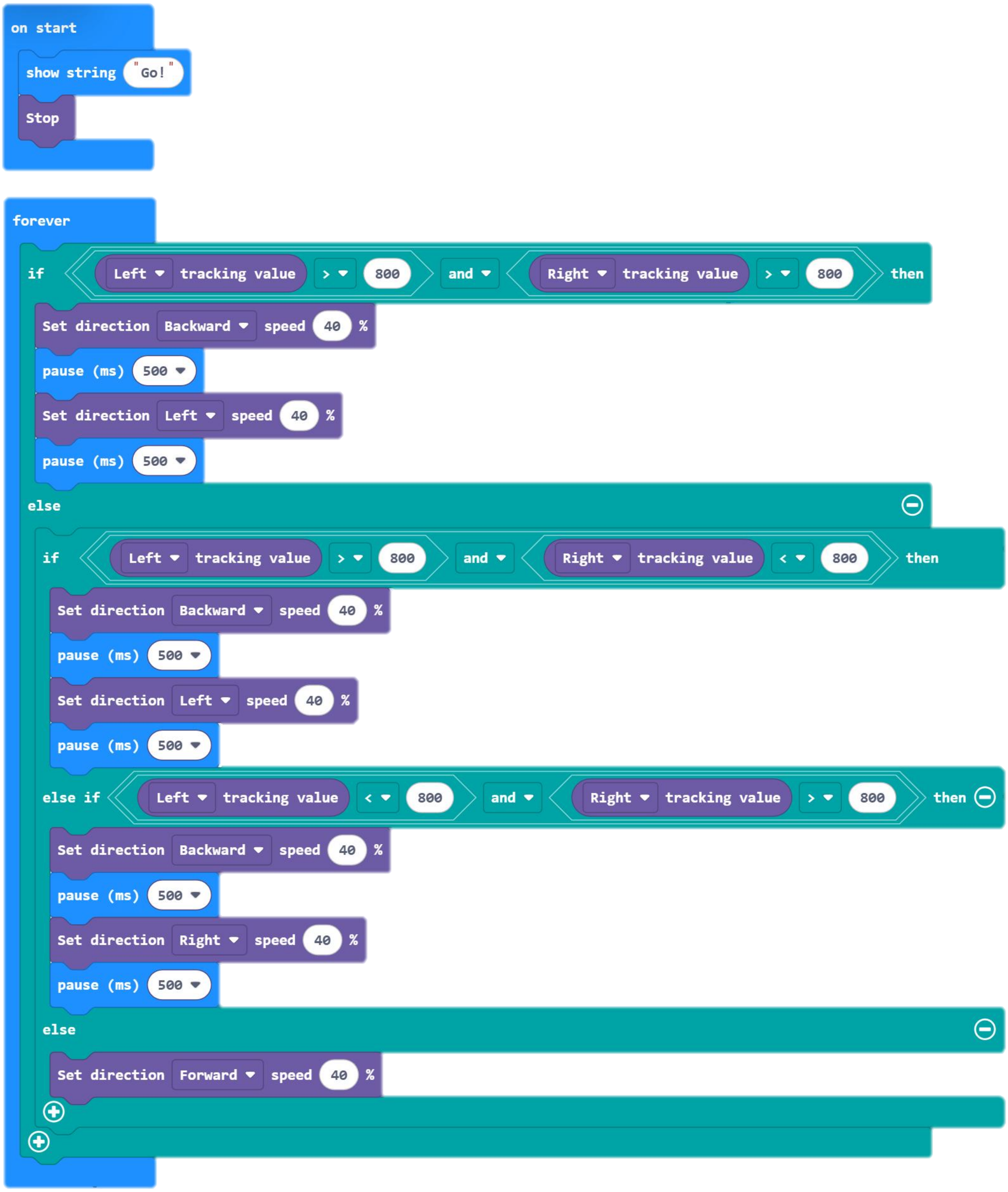
(2) Task 2: Make the Car Move Only within the Black line

① Draw a Program Flow Chart



② Reference Program

[Click here to view the program "Lesson 12" for the robot to detect the black line boundary.]



③ Run the Program

After downloading the program to the main control, turn on the power of the car, place the car in the black line map, and observe whether the car retreats and turns in time after reaching the black line boundary.

Inspector Lumi watched the car safely pass the steep cliff section, but the road ahead seemed unusually quiet. The target did not appear, and there were no clues. The only thing that attracted attention was a long black line that stretched from their feet to the unknown distance.

Lesson 13: The Stalker

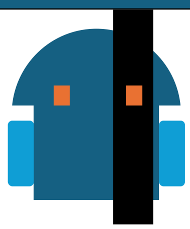
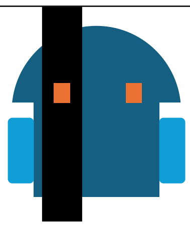
TinkerBott has lost track of the target person, and only a winding black line is left in front of it. Inspector Lumi frowned and analyzed calmly: "This track seems to lead us to somewhere. It seems that we need to let the TinkerBott smart car drive along this black line to continue tracking the target. Don't forget that the line patrol sensor can see the black line."

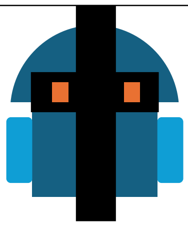
You need to make the TinkerBott smart car run accurately along the black line and maintain stable and smooth movement. Before completing the task, you need to understand the relevant knowledge of line patrol.

1. Knowledge Learning

1. Car Patrol Logic

Since the left and right grayscale probes of the line patrol sensor can detect the intensity of the reflected light from the ground in real time, the current position of the car can be determined based on the returned value, that is, whether the car deviates from the black line, and the driving direction can be corrected by adjusting the speed of the left and right motors. The specific algorithm is shown in the table below:

Sensor Status	Description	Smart Car Adjustment Action
	The left sensor detects the white area and the right sensor detects the black line.	Turn right
	The right sensor detects the white area and the left sensor detects the black line.	Turn left

	Both sensors do not detect the black line.	Straight
	Both sensors detect the black line.	Depending on the specific situation

2. Relationship between the Car Position and the Line Patrol Sensor Value

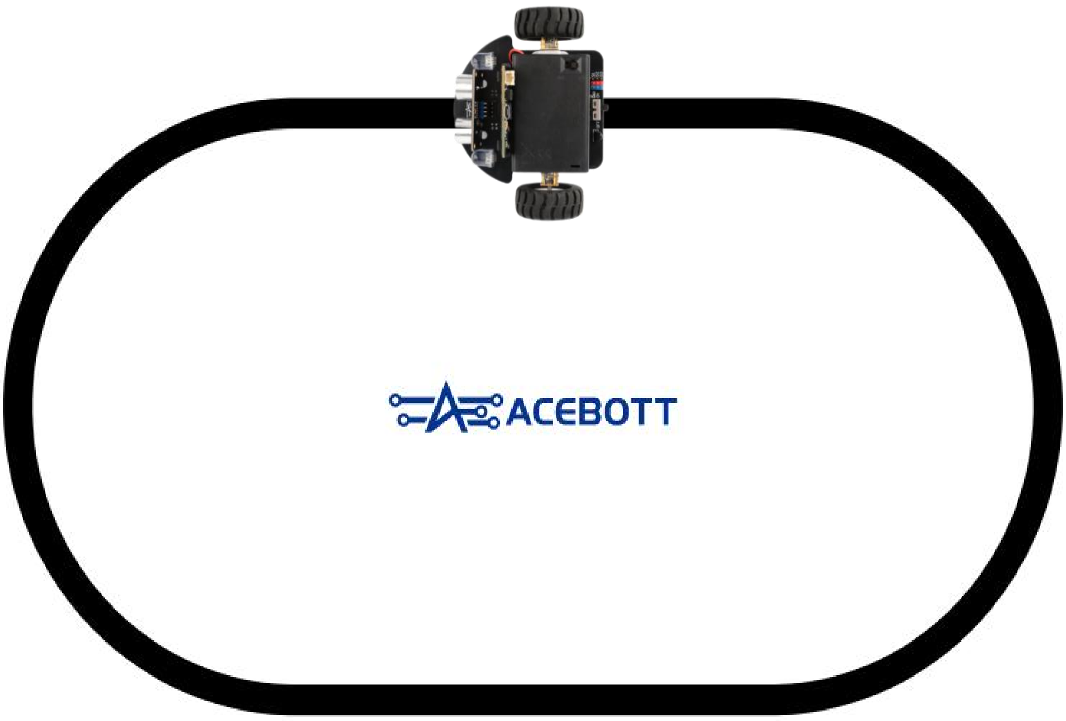
According to the above situations when the car is driving on the black line, you need to detect the corresponding values of the line patrol sensor. Just like the last lesson, you can use the buttons and LED dot matrix screen to obtain the values returned by the left and right probes in the current environment and record them.

The line patrol sensor completely detects the black line, and the grayscale value is high (such as greater than 900). The grayscale value of the white area detected is low (such as less than 300, which depends on the value of the on-site test.)

2. Mission Decryption

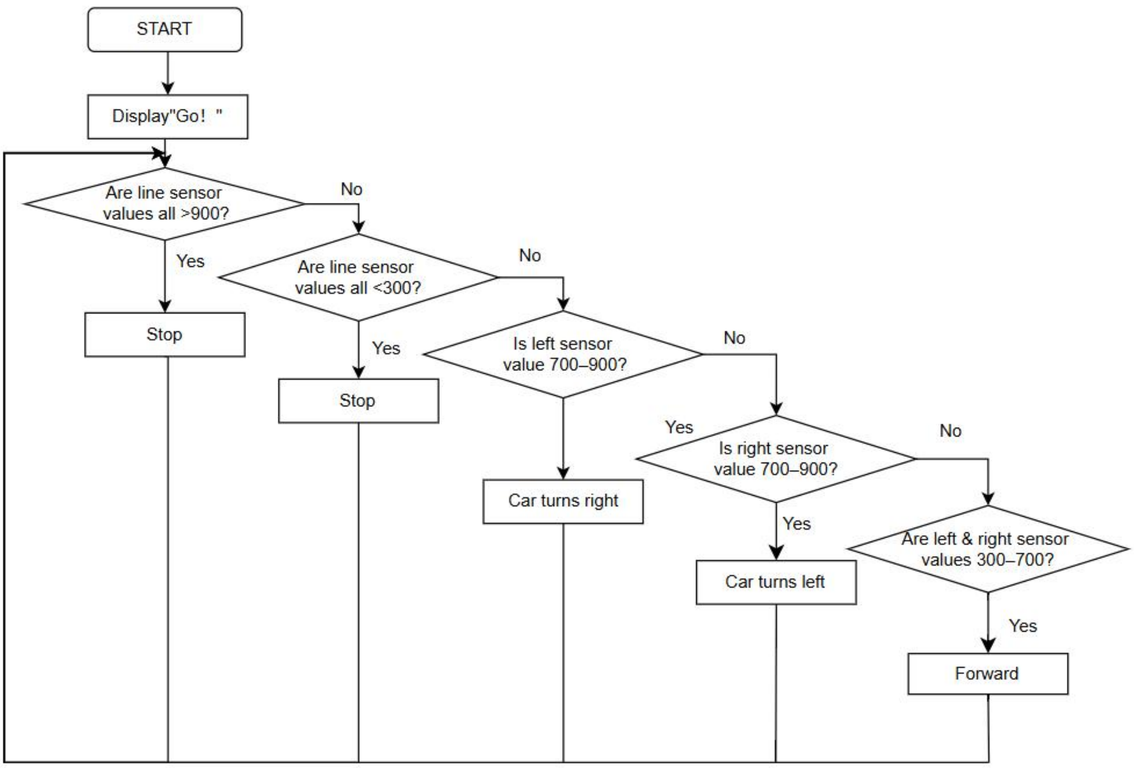
1. Task Description

Program TinkerBott to move along the black line below.



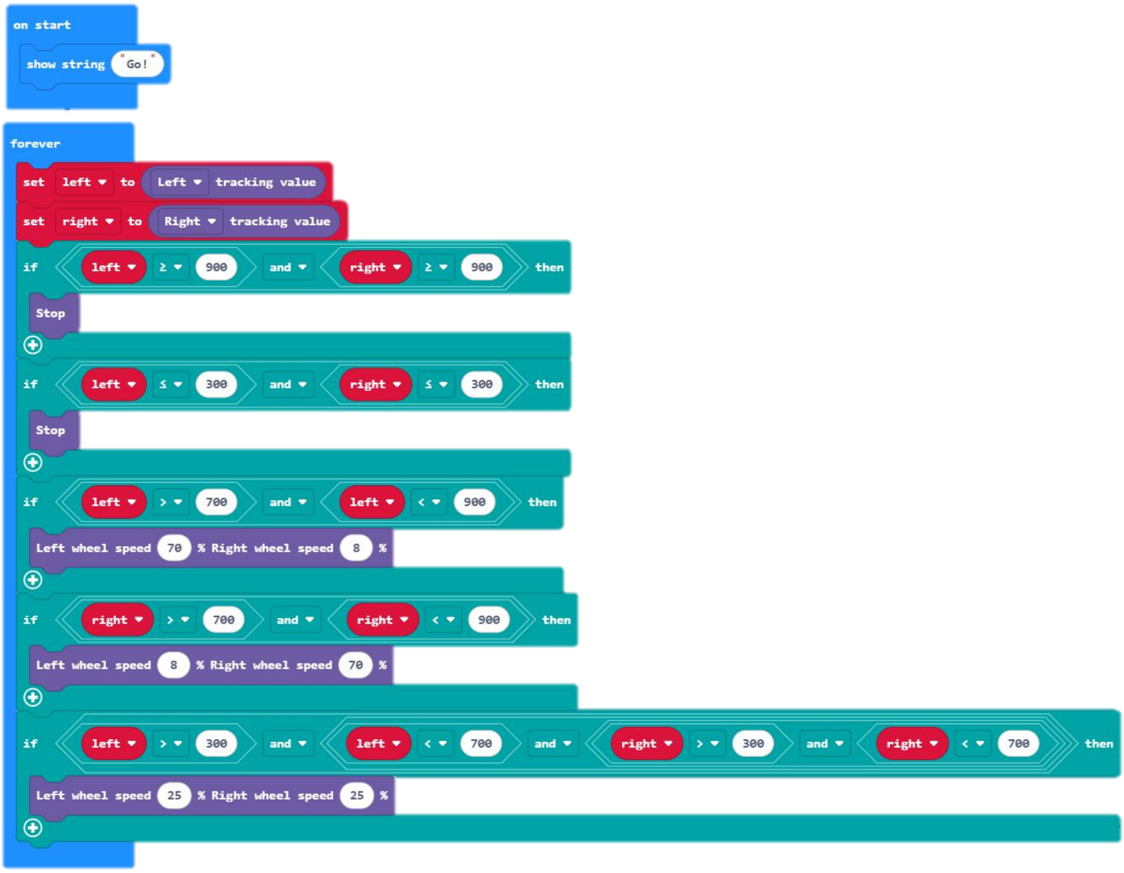
2. Programming

(1) Draw a Program Flow Chart



(2) Reference Program

[Click here to view the program "Lesson13" for the car's line-following motion.]



If the car cannot move normally during line inspection, it may be due to the following reasons:

1. The car is moving too slowly.

To solve the problem, you can increase the motor PWM value in the program to give the car enough speed.

2. Due to interference from ambient light factors, TinkerBott may mistakenly identify white areas as black lines.

To solve this problem, you can first test the specific values of the grayscale sensor on the white area and the black line, and adjust the threshold of the line-following sensor in the program accordingly.

(3) Run the Program

After downloading the program to the main control, place the car on the black line on the map, turn on the power switch of the car, and observe whether the car can move along the black line.

Inspector Lumi's car sped along the black line and found that the target vehicle had entered a busy residential area. Suddenly, the car braked and stopped. What happened?

Lesson 14: Intelligent Driving

As TinkerBott continued to drive along the black line, small animals suddenly appeared in front of it. In order to protect the safety of these small animals, TinkerBott had to stop in time when it found small animals. Faced with this situation, Inspector Lumi said: "We need to enable the car to handle multiple tasks at the same time: it must be able to patrol the line and avoid obstacles in front of it, so as to achieve more intelligent driving."

You wondered: "Should TinkerBott avoid obstacles first or patrol the line first when performing a task?"

Detective Lumi said: "Good question! When the car encounters multiple tasks at the same time, it needs to determine the priority of each task. Next, it is up to you to implement the multi-tasking functions of obstacle avoidance and line patrol for TinkerBott smart car!"

1. Knowledge Learning

1. Intelligent driving Decision-Making

It is a core link in autonomous driving technology, which means that after the intelligent car perceives the environment, it automatically makes safe and reasonable driving behavior choices based on factors such as traffic rules, road conditions, and driving goals, just like human drivers constantly make "decisions" when driving. Safety is the most important thing in the driving process of the car, so obstacle avoidance is the top priority for the car, followed by line patrol.

2. Functions

In programming, a function is a block of code with a specific function that can be called repeatedly, thereby improving the readability, maintainability, and reusability of the code. By encapsulating the ultrasonic obstacle avoidance and black line driving functions into two functions, we can easily call them in the main program to achieve comprehensive control of the car. The use of functions makes the code structure clearer and facilitates subsequent modification and expansion.

Functions usually have the following structure:

●**Define function:** Define the function to be completed through programming blocks or codes.

●**Call function:** Call the defined function at the required location to execute the function.

2. Mission Decryption

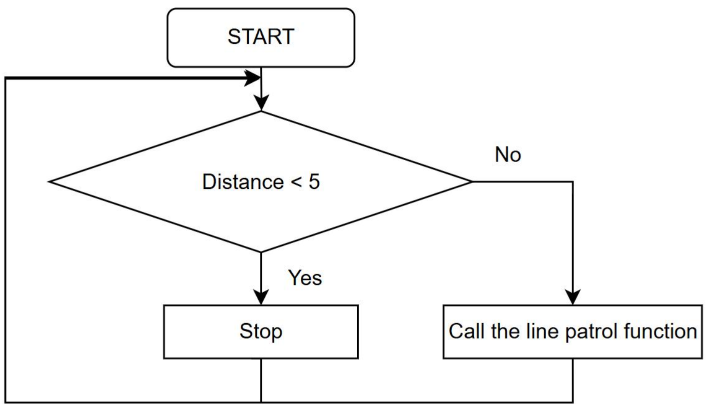
1. Task Description

Programming makes the TinkerBott smart car move steadily along the black line, and stop when encountering an obstacle, waiting for the obstacle to be removed before continuing to move along the black line.

2. Programming

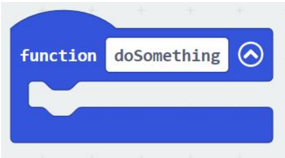
(1) Draw a Program Flow Chart.

According to the task requirements, the TinkerBott smart car should prioritize avoiding obstacles, and only execute the line patrol function when no obstacles are detected.



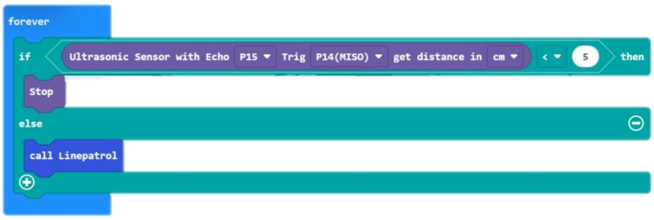
(2) Block Learning

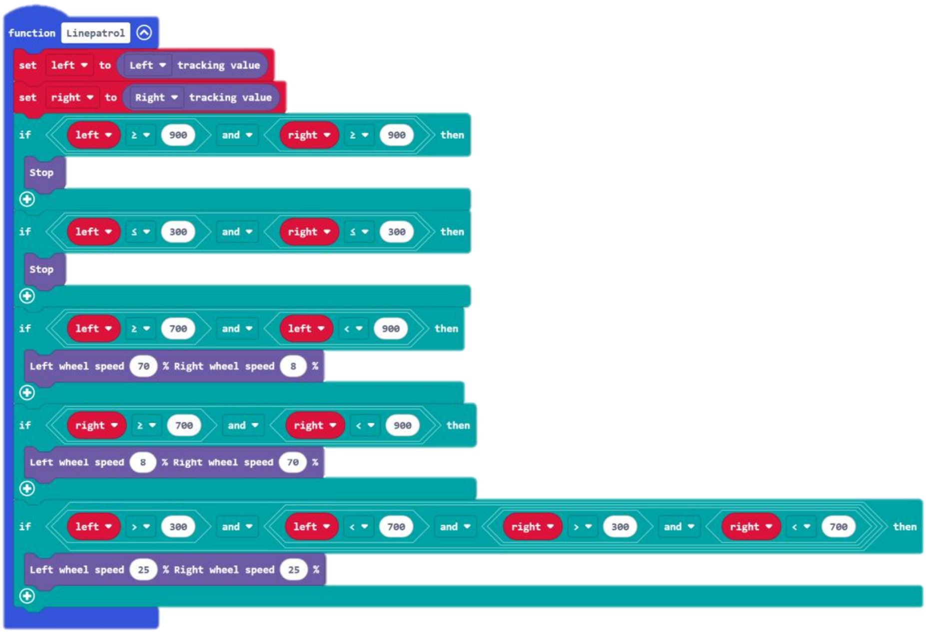
In order to realize the functionality of the encapsulated function, you need to find the "Functions" command in the programming platform. Then you can create and call the function according to the following methods and statement blocks.

Building Blocks	Description
	Click “Make a Function” to create one or more functions.
	Declare a function named "doSomething". You can define your own function name. Add code to the function body to achieve the required functionality.
	Return statement, used to return a value from a function to the caller.
	Used to call defined functions and execute corresponding functions.

(3) Reference Program

Click here to view the program "Lesson14" of the smart driving car.





(3) Run the Program

After downloading the program to the main control, place the car on the black line on the map, turn on the power switch of the car, let the car move along the black line, and observe whether the car can achieve the obstacle avoidance function when patrolling the line.

Lumi's car successfully combined the line patrol and obstacle avoidance functions and finally drove along this black line section. Suddenly, the target vehicle made a sharp turn and disappeared without a trace, leaving only some tire marks on the ground. The line patrol mode also did not work, and the other party seemed to have discovered that the car was following along the black line.

Lesson 15: Infrared Remote Control Pursuit

Looking at the scattered tire tracks at the scene, Inspector Lumi thought for a while and said, "It seems that the target has discovered us. The movement path of the TinkerBott smart car in patrol mode is too fixed and we can't continue to track it." Then, Inspector Lumi took out a remote control from his pocket and said, "The next tracking requires more flexible operation! We must manually control the car to quickly find the target's trace!"

Next, you need to realize the function of remote control of TinkerBott smart car. Before completing the task, you need to understand the knowledge related to infrared remote control.

1. Knowledge Learning

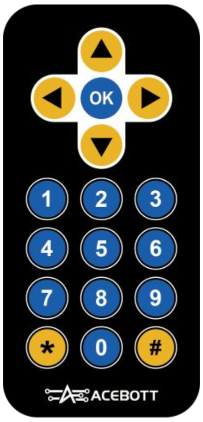
1. Infrared Communication

Infrared communication is a wireless communication technology that uses infrared light waves for data transmission. Infrared light is located in the electromagnetic spectrum, between microwaves and visible light, and its wavelength ranges from approximately 700 nanometers to 1 millimeter.

The realization of infrared communication mainly depends on two parts: infrared transmitter and infrared receiver. The infrared transmitter is responsible for transmitting infrared signals, and the infrared receiver is responsible for receiving infrared signals transmitted by the infrared transmitter.

2. Infrared Remote Controller

The commonly used infrared transmitter is the infrared remote controller, which consists of multiple buttons and infrared light-emitting diodes. Each button corresponds to a different infrared code value. After pressing the button, the infrared light-emitting diode will emit infrared light with the corresponding code. There are many coding methods for infrared remote controls, and the most common coding method is NEC coding.



Button	Value	Button	Value	Button	Value
▲	B946FF00	▼	EA15FF00	OK	BF40FF00
◀	BB44FF00	▶	BC43FF00	1	E916FF00
2	E619FF00	3	F20DFF00	4	F30CFF00
5	E718FF00	6	A15EFF00	7	F708FF00
8	E31CFF00	9	A55AFF00	0	AD52FF00
*	BD42FF00	#	B54AFF00		

3. Infrared Receiver

(1) Infrared Receiver

The function of the infrared receiving module is to receive the infrared light emitted by the infrared remote control and convert the infrared light into a corresponding level signal which is input into the controller. The controller then converts the level signal into a corresponding key code value through a program algorithm.

(2) Infrared Receiving Module Parameters

Parameter	Description
Operating voltage	3.3-5V
Signal pin	P11
Modulation frequency	38kHz (standard infrared signal carrier

	frequency)
Effective receiving distance	0.5 - 5 meters (requires the remote control and receiver to be in a straight line without any obstruction)

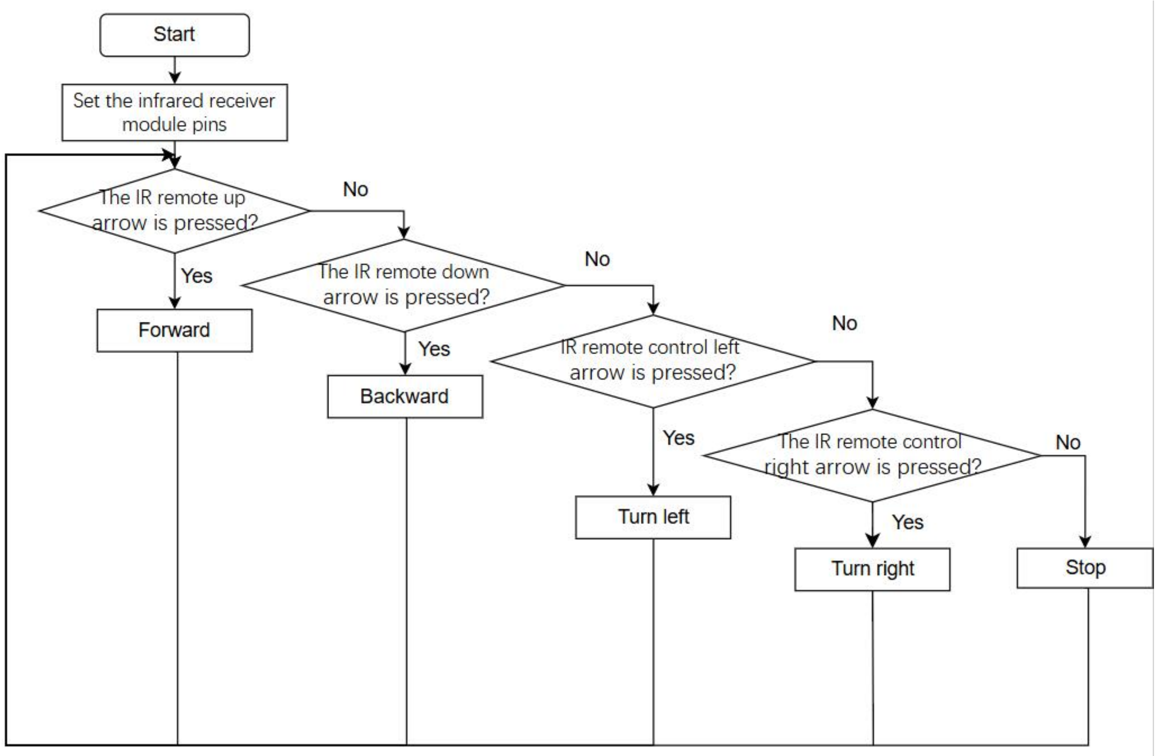
2. Mission Decryption

1. Task Description

Programming to use infrared remote control to control the TinkerBott smart car to move forward, backward, left and right.

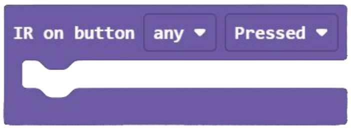
2. Programming

(1) Draw a Program Flow Chart



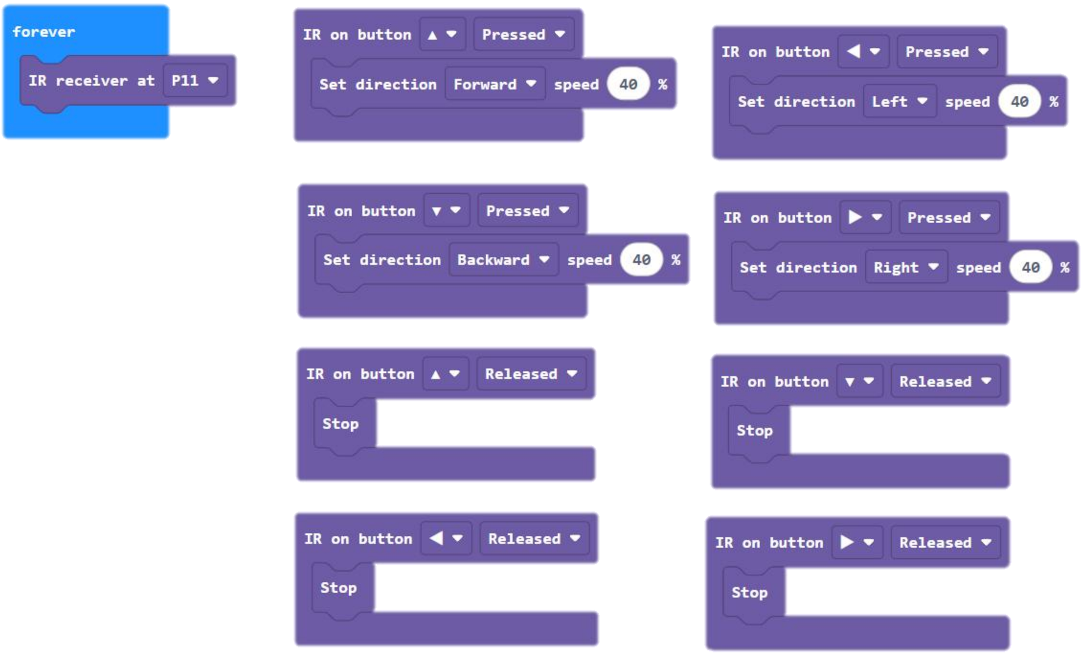
(2) Block Learning

You can find the Acebott car expansion package in the Micro:bit programming platform, and find the infrared receiving building block statement in the "Sensor" instruction. The description of these building block blocks can be referred to in the table below.

Building Blocks	Description
	Initialize the infrared receiving function of the car, here set the pin to P11
	Used to detect whether an infrared signal is received. The infrared receiver will return True when it receives a valid signal.
	Used to decode the received infrared key value. The parameter can be any, front, back, left, right, numbers 1-9, etc. The parameter "any" means that it can match any infrared remote control key value.
	Used to create infrared key event response. The first parameter is the key being pressed, and the second parameter can be "Pressed" or "Released", which represent whether the key is pressed.

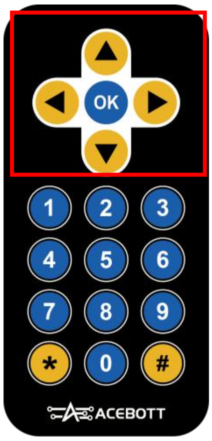
(3) Reference Program

[Click here to view the program "Lesson15" for infrared remote control of the car.]



(4) Run the Program

After downloading the program to the main control, turn on the power of the car and use the front, back, left, and right buttons of the infrared remote control to control the movement of the car.



Inspector Lumi uses an infrared remote control to control the TinkerBott smart car. However, during the process, he found that the remote control sometimes failed to work. Why is that?

Lesson 16: Bluetooth Communication

When using the infrared remote control to control the TinkerBott smart car, the infrared remote control sometimes fails. After observation, Inspector Lumi found that if there is an obstacle between the infrared remote control and the infrared receiver, the TinkerBott smart car cannot receive the remote control signal. This is because the traditional infrared light is very weak and cannot penetrate obstacles. Faced with this situation, Inspector Lumi thought for a moment and said: "In order to control TinkerBott more conveniently, we can choose wireless communication technology with stronger penetration and wider transmission range, such as Bluetooth communication."

You asked in confusion: "If we use Bluetooth communication, what should we use to control the TinkerBott smart car?"

Inspector Lumi replied, "We can install the ACEBOTT APP on our mobile phones and use the mobile phones to control it."

At this moment, the mysterious shadow's car suddenly accelerated and rushed to an abandoned factory on the outskirts of the town. Detective Lumi looked determined: "The truth is right in front of us. Next, you have to learn how to use the Bluetooth remote control to control the car to catch up quickly!"

1. Knowledge Learning

1. Bluetooth Communication Basics

Bluetooth is a short-range wireless communication technology that enables data transmission between devices. It is widely used in wireless headphones, smart homes, and connections between mobile devices. Bluetooth technology has the characteristics of low power consumption, high speed and convenient connection, making it very suitable for controlling small smart devices.

2. How Bluetooth Works

Bluetooth communicates via radio frequency signals in the 2.4GHz band. The following steps need to be completed between devices:

① **Pairing and connection**: The device searches and connects to the target device through the Bluetooth module.

② **Data transmission**: Devices send and receive data via Bluetooth protocol.

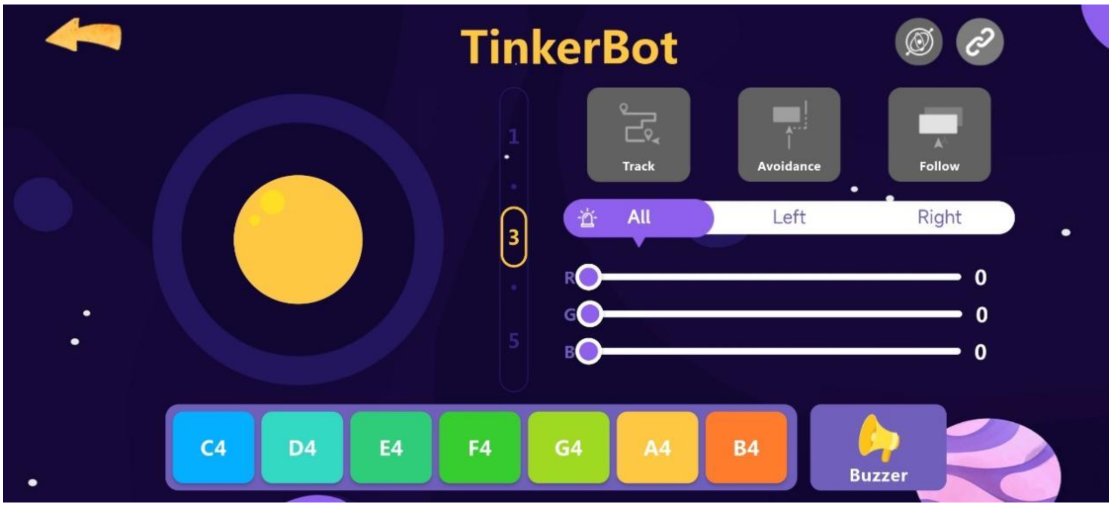
③ **Decoding and execution**: The receiving end decodes the transmitted data and performs corresponding operations, such as controlling the movement of the car.

3. ACEBOTT APP

To control TinkerBott through a mobile phone, we also need to install the ACEBOTT APP on the mobile phone and use the ACEBOTT APP to control the movement of the smart car. The specific steps are as follows:

If it is an IOS phone, you need to search the keyword: ACEBOTT in the APP Store and then download it; if it is an Android phone, you need to search the keyword: ACEBOTT in the Google Play Store and then download it. The icon is shown on the right.

After opening the APP, find the control interface of the TinkerBott smart car, click the [Control] module, and the control interface of the car will appear, as shown in the figure below.



There are many function buttons on the Micro:bit car APP interface. When these buttons are triggered, they can send different protocol instructions to the car's Bluetooth module. After receiving these protocol instructions, the Bluetooth module can parse and process the instructions. For example, if you push the joystick up, the mobile phone APP sends the "0501" instruction through Bluetooth. After the car's Bluetooth module receives the instruction, the car can move forward according to the received instruction. The following are the protocol instructions for joystick control:

Device Button	Protocol instructions	Car instructions
Joystick up	0501	Forward
Joystick down	0502	Backward
Joystick left	0503	Turn left
Joystick right	0504	Turn right
Joystick stop	0500	Stop

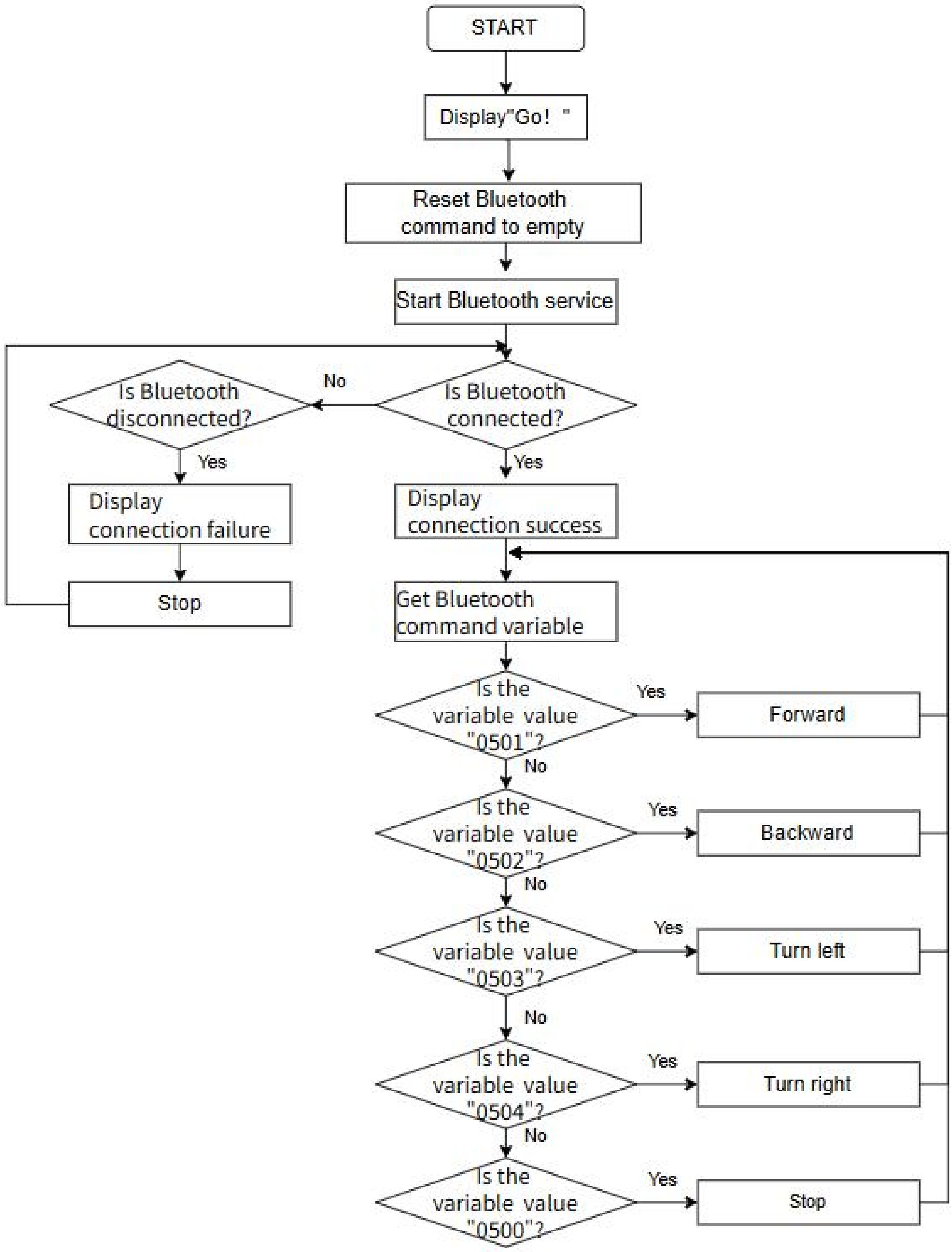
2. Mission Decryption

1. Task Description

Programming to use ACEBOTT APP to control the movement functions of TinkerBott smart car.

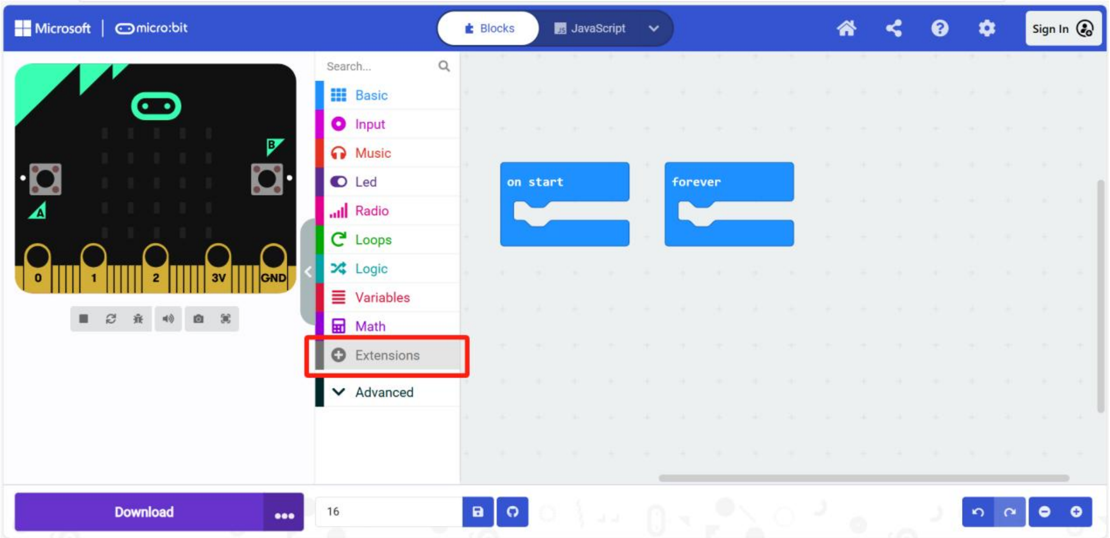
2. Programming

(1) Draw a Program Flow Chart

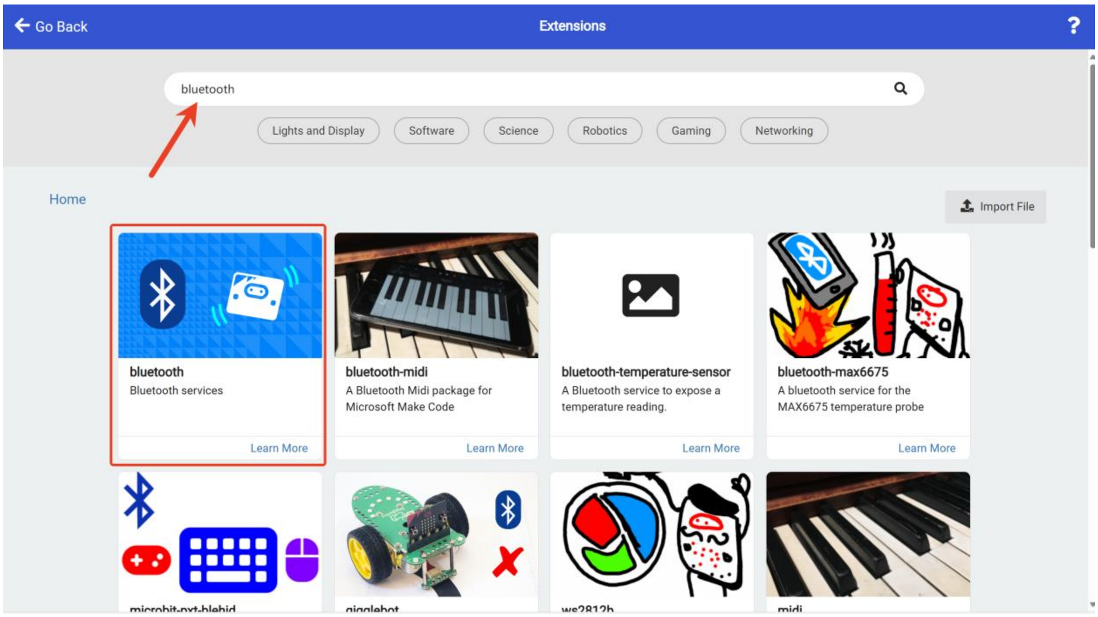


(2) Bluetooth Command Learning

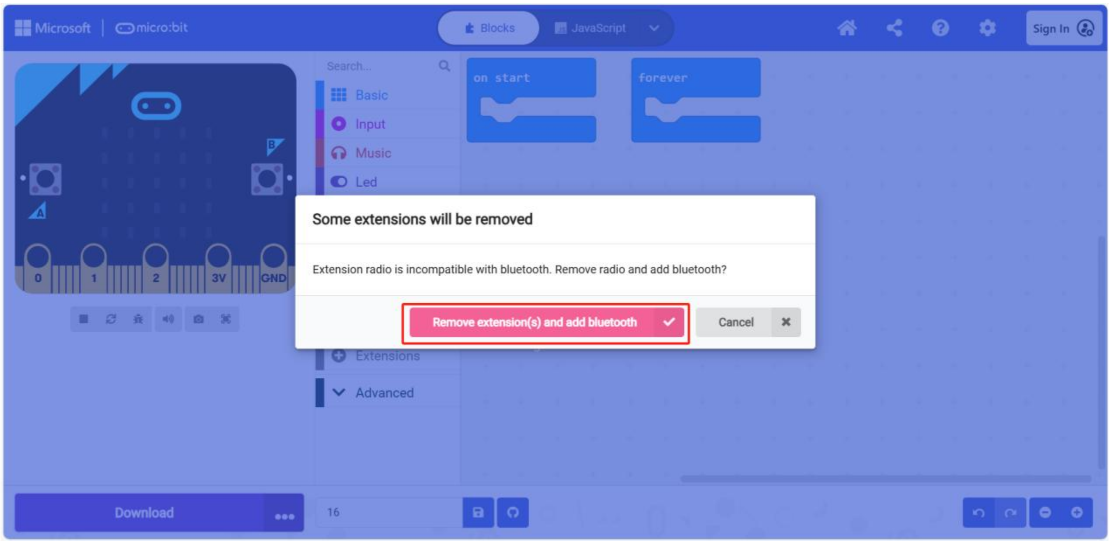
① To control the TinkerBott smart car via Bluetooth, you need to create a new project for this lesson and click "Extensions".



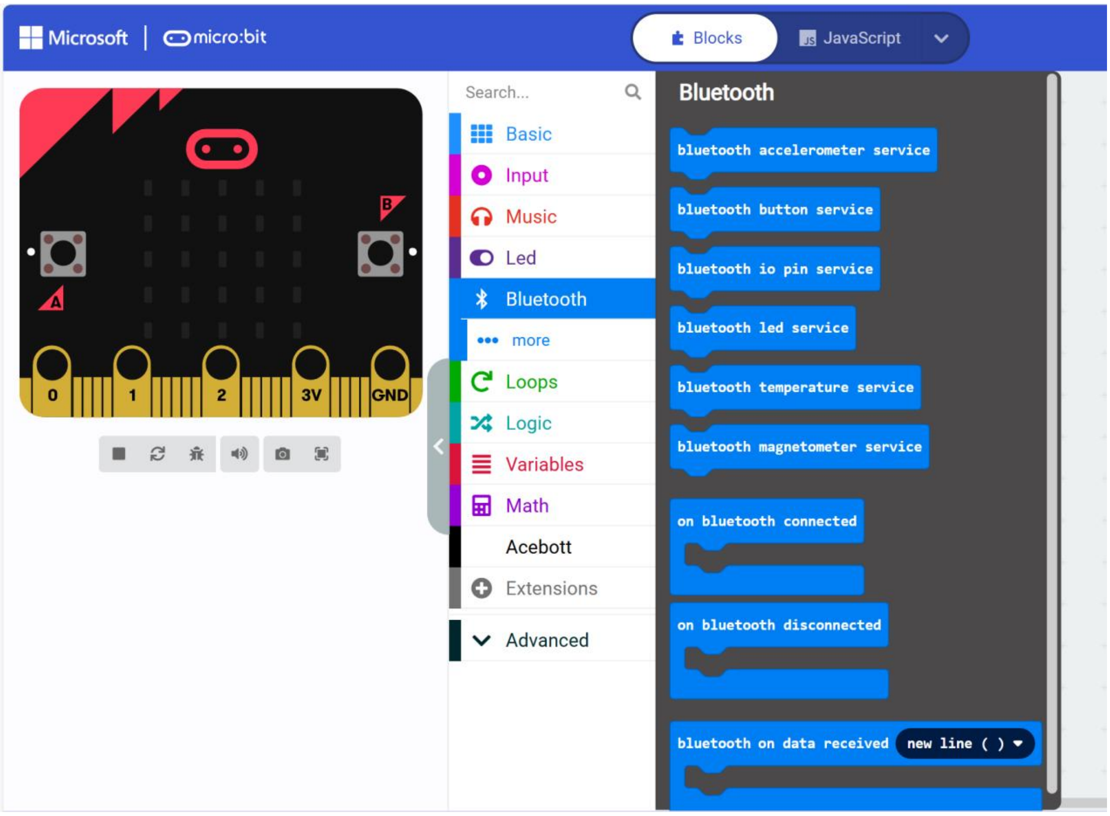
② In the pop-up search interface, enter “bluetooth” and select the corresponding Bluetooth expansion board package.



③ Then a prompt will pop up: incompatible with the radio module, click "Remove extension(s) and add bluetooth", then the instructions of the Bluetooth module will be loaded into the programming interface.

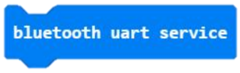


④ After adding the Bluetooth expansion package, click "Bluetooth" and "more", and the relevant instructions of the Bluetooth programming module will appear.



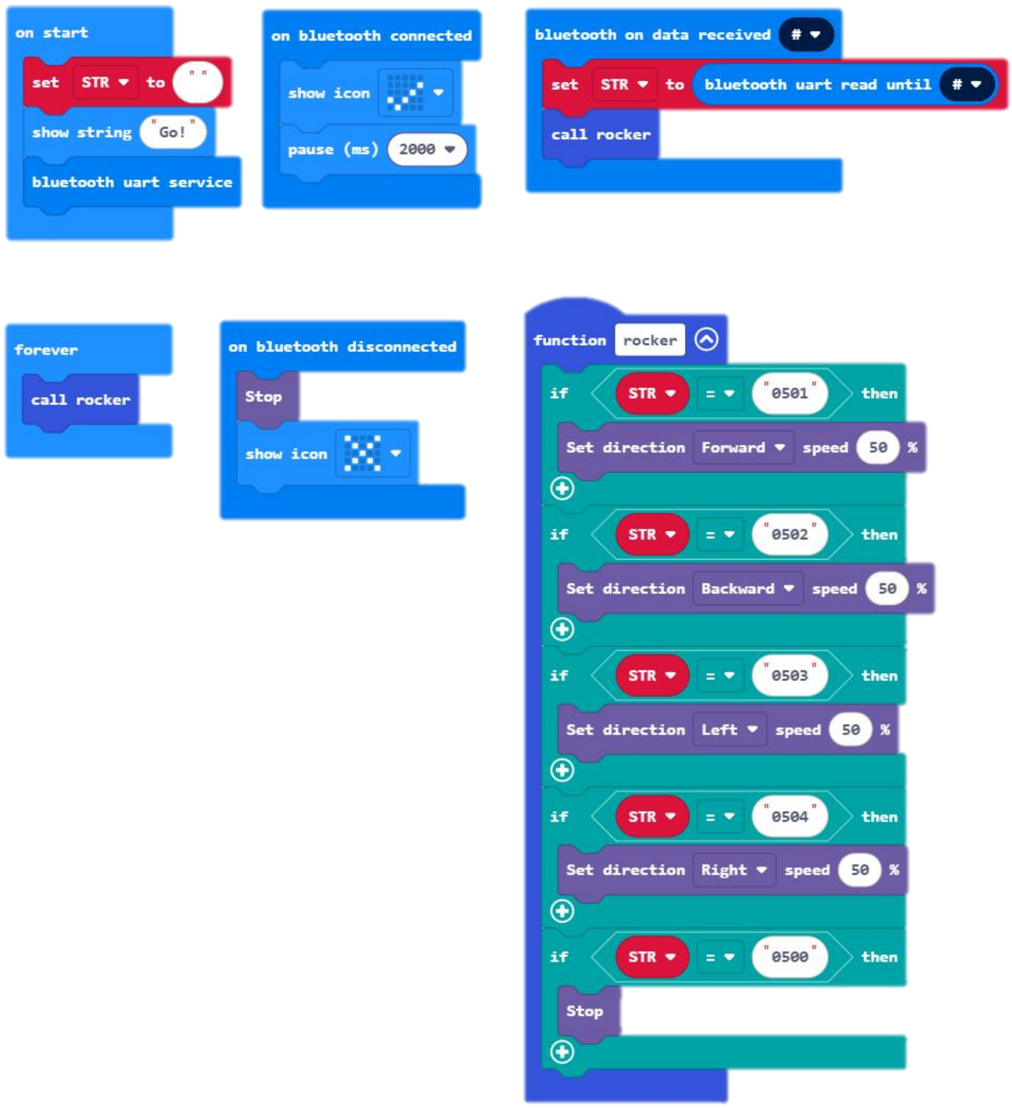
In this class, you will learn how to use the "Bluetooth" and "...more" Bluetooth programming statements in the programming platform to complete the task of controlling the movement of the Bluetooth car.

Building Blocks	Description

	Used to initialize or configure Bluetooth serial port service
	Read data from the Bluetooth serial port until encountering a specified delimiter (configurable via dropdown menu) to trigger an event. When sending Bluetooth remote control commands to the car, "#" serves as an end-of-command marker. Upon detecting this character, the system recognizes that a complete command has been received and proceeds to parse and execute it.
	Used to handle the event of successful connection of Bluetooth device. Write code in this module to perform specific operations
	Process data received from the Bluetooth serial port until a specified delimiter (configurable via dropdown menu) is encountered. Similar to block #2, but includes additional logic for processing received data, such as parsing commands.
	Used to handle the event of Bluetooth device disconnection.

(3) Reference Program

Click here to view the APP joystick controlled car program "Lesson16_1".



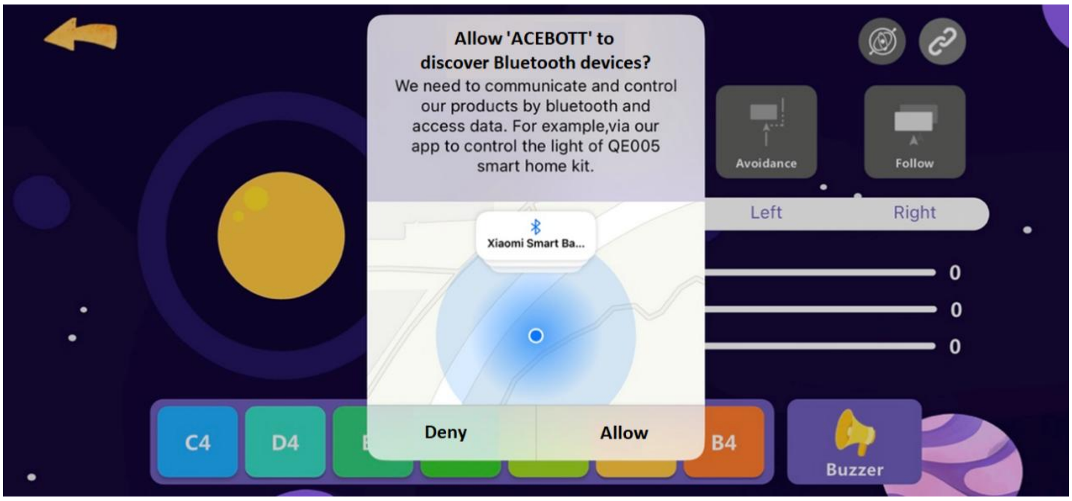
(4) Bluetooth Remote Control Operation Steps

① After burning the program to the car's main controller board, unplug the data cable and turn on the car's power;

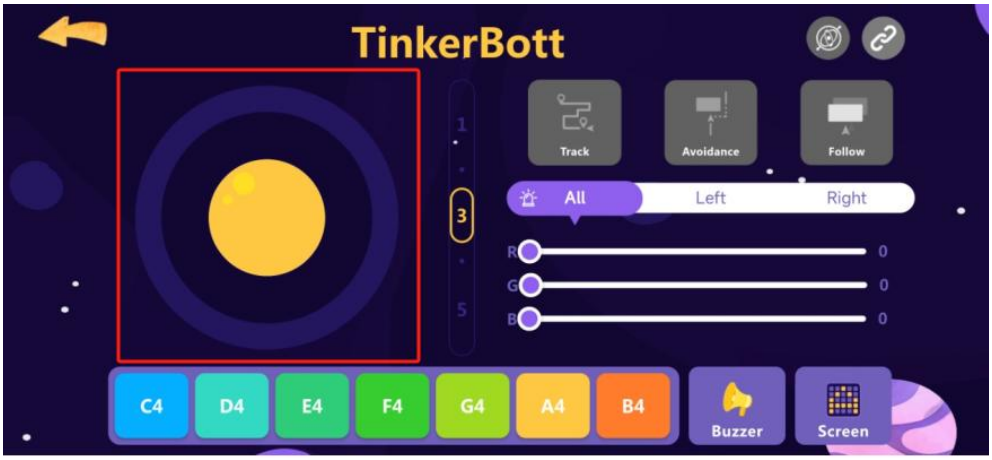
② Turn on the Bluetooth function of your mobile phone; open the ACEBOTT application, click the "Control" button at the top, and find the "TinkerBott" car.

③ Next, select the operation mode of the car and click "Control" to enter the remote control panel interface of the car.

Note: When loading for the first time, a prompt may appear asking "Allow ‘ACEBOTT’ to discover Bluetooth devices?" Please select "Allow" to connect and control the function normally.



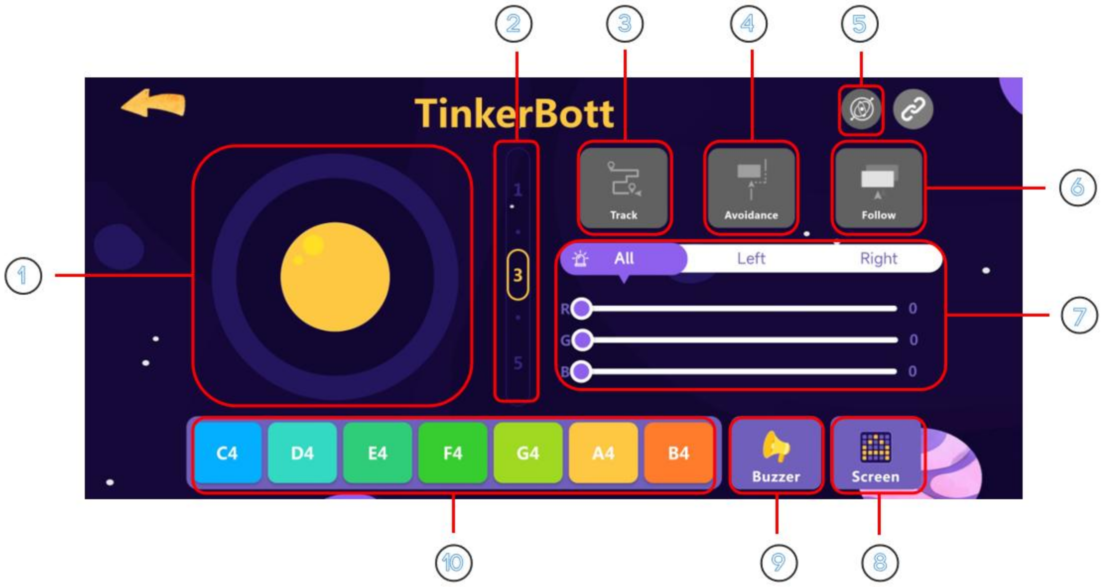
④ Click the Bluetooth icon in the upper right corner of the TinkerBott smart car remote control panel. When the Bluetooth icon changes from gray  to purple  , it means the Bluetooth connection is successful. At this time, you can operate the left joystick to control the car's forward, backward, left, and right movements!



If you want to experience the functions of other buttons on the APP, such as line patrol, obstacle avoidance, etc., you can click on the reference program below and download it to the car's main control, making full use of all the functions of the car to help Lumi detective solve the case quickly!

Click here to view the complete program "Lesson16_2" for controlling the car with the APP.

The functions of the APP remote control panel are as follows:



① **Joystick button:** Control the movement of the car forward, backward, left and right by manually operating the joystick button in the circular area.

② **Motor speed levels:** adjust the motor speed, divided into 5 levels (0-5), the larger the number, the faster the speed.

③ **Line tracking:** Press the "Track" button and the car switches to automatic line tracking mode.

④ **Avoidance:** Press the "Avoidance" button and the car will switch to the mode of automatically avoiding obstacles.

⑤ **Gyroscope:** Press this button to use the phone's gyroscope function to control the car's movement by tilting the phone.

⑥ **Following:** Press the "Follow" button and TinkerBott will follow the specific target.

⑦ **RGB headlight control:** By clicking the All/Left/Right button, you can switch to control the on and off of different lights. The sliding button below can adjust the brightness of the headlights (0-255), and supports RGB color adjustment.

⑧ **Screen:** Screen pattern drawing. By drawing the pattern of the dot matrix screen in real time on the mobile phone, the pattern on the screen of the car can be controlled.

⑨ **Play music:** Press the "Buzzer" button to play the corresponding music. There are four songs in total.

⑩ **Note playback:** Note playback buttons. The 7 buttons correspond to the notes **C4, D4, E4, F4, G4, A4, B4.** Press to play the corresponding note.

End

With your help, the car activated the Bluetooth mode, and the Bluetooth signal penetrated the factory gate. Detective Lumi used the APP to smoothly control the car to rush into the factory. In the center of the dim warehouse, they finally discovered the true face of the "mysterious shadows" - it turned out that they were a group of abandoned experimental robots! They wandered around because of program confusion. In order to repair their own hardware, they once acted quietly at night and dismantled the car in front of the library, just to get the parts. They used sound, light and hidden tracks to transmit information, trying to ask for help from humans.

"It turns out that these robots were once the assistants of the town's scientists, but they were forgotten here after the experiment was interrupted." Detective Lumi sighed. He pressed the Stop button on the control console in the center of the warehouse, and all the robots lit up blue lights at the same time and then slowly turned off the lights. They finally stopped their disorderly actions.

The town has returned to peace. The residents repaired the programs for the robots, and they became the patrollers of ACE Town. And you, the young detective, also won the Medal of Honor of ACE Town because of this adventure!