DFROBOT
DRIVE THE FUTURE

micro:bit

A STARTER GUIDE OF BOSON KIT

Content

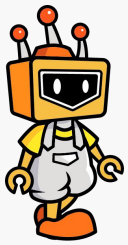
Chapter 1: Mind+ and micro: bit.....6
1. Introduction to Mind+.....7
1.1 Download and Install Mind+.....7
1.2 Mind+ Interface.....10
2. Micro: bit.....12
3. Get Started with Mind+ and micro: bit.....14
Chapter 2: What Makes a Machine "Come to Life".....18
Project 1: The Mysterious micro: bit.....18
1.Story of Micko and Dr.M.....18
2.Dr.M's Pocket.....19
3.Dream of Micko.....19
4.Dr.M's Blackboard.....19
5. Design of Micko.....22
Project2: Flash the LED Light.....23
1. Story of Micko and Dr.M.....23
2.Dr.M's Pocket.....24
3.Dream of Micko: Light up LEDs.....24
4.Dr.M's Blackboard.....24
5.Dream of Micko: Flash the LED Light.....28
6.Dr.M's Blackboard.....28
7.Design of Micko.....30
Project 3: Notification Light.....32
1.Story of Micko and Dr.M.....32
2.Dr.M's Pocket.....33
3.Dream of Micko: Control LED by Push Button.....34
4.Dr.M's Blackboard.....34
5.Dream of Micko: Control LED by Knob.....38
6.Dr.M's Blackboard.....38
7.Design of Micko.....40
Project 4: Electric Fan.....41

1.Story of Micko and Dr.M....41
2.Dr.M's Pocket....42
3.Dream of Micko: Push-button Fan....42
4.Dr.M's Blackboard....43
5.Dream of Micko: Set the fan running at different speed....46
6.Dr.M's Blackboard....46
7.Design of Micko....48
Chapter 3: Get hands on!....49
Project 1: Electronic Candle....49
1.Story of Micko and Dr.M....49
2.Dr.M's Pocket....50
3.Dream of Micko: Detect the sound strength....50
4.Dr.M's Blackboard....51
5.Dream of Micko: micro: bit Candle....53
6.Dr.M's Blackboard....53
7.Creation of Micko....54
Project 2: Automatic Door....55
1.Story of Micko and Dr.M....55
2.Dr.M's Pocket....56
3.Dream of Micko: Control Servo with Push Button....57
4.Dr.M's Blackboard....57
5.Dream of Micko: Control the Servo with Motion Sensor....60
6.Dr.M's Blackboard....60
7.Creation of Micko....62
Project 3: Music Box....63
1.Story of Micko and Dr.M....63
2.Dr.M's Pocket....63
3.Dream of Micko: Play Music....64
4.Dr.M's Blackboard....64
5.Dream of Micko: Smart Music Box....66
6.Dr.M's Blackboard....66
7.Creation of Micko....68

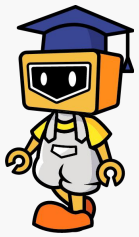
Project 4: Colorful LED Strip....69
1.Story of Micko and Dr.M....69
2.Dr.M's Pocket....70
3.Dream of Micko: Turn on the RGB LED strip....70
4.Dr.M's Blackboard....71
5.Dream of Micko: Turn on the rainbow LEDs....74
6.Dr.M's Blackboard....74
7.Dream of Micko: RGB LED strip controlled by a sound sensor....78
8. Dr.M's Blackboard....78
9.Creation of Micko....80
Chapter 4: Become an expert....81
Project 1: Electronic Stabilizer....81
1.Story of Micko and Dr.M....81
2.Dr.M's Pocket....82
3.Dream of Micko....82
4.Dr.M's Blackboard....84
5.Invention of Micko....87
Project 2: DJ panel....89
1.Story of Micko and Dr.M....89
2.Dr.M's Pocket....90
3.Dream of Micko....90
4.Dr.M's Blackboard....91
5.Invention of Micko....94
Project 3: Automatic Door....95
1.Story of Micko and Dr.M....95
2.Dr.M's Pocket....96
3.Dream of Micko....96
4.Dr.M's Blackboard....98
5.Invention of Micko....102
Project4: The Bomb Escape....103
1.Story of Micko and Dr.M....103
2.Dr.M's Pocket....104

3.Dream of Micko..... 104
4.Dr.M's Blackboard..... 105
5.Invention of Micko..... 113

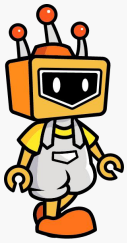
Chapter 1: Mind+ and micro: bit



Hello, Dr.M. More and more people are learning to code and their works are so funny.
And I also want to design my own works!

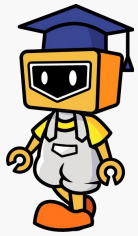


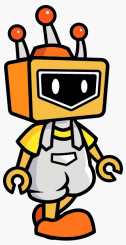
Well? Sweetie, what kind of works do you prefer?



I wanna make a light sensitive lamp.
I want to make a birthday gift for my best friend, like a colorful music box.
I want make a remote car to my classmates.
And I want to make a door that I can open it when sitting in a sofa.
I, I want to make all of these interesting DIYs!

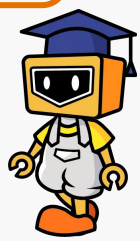
Adorable! Let me introduce a tool to make them all, a magic graphical coding software named Mind+. And many new things, such as electronic hardware with many functions and a microcontroller micro: bit.

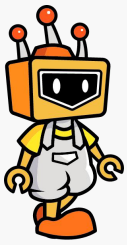




But Dr.M, I am afraid I cannot. I know nothing about programming. It seems too hard to learn programming.

Do not worry~ Even though we do not know complex codes, we can make many smart devices.





Really? Great! Let's begin our programming with Mind+ and micro: bit!

1. Introduction to Mind+

With so many different graphical Programming software products on the market, how can you get the right one? It may disappoint you when you found that your software only supports some certain hardware and robots or even cannot work with any hardware at all. Where to find a software with all you wanted?

Why not try our Mind+ graphical Programming software!

Mind+ is a graphical Programming platform that can support all kinds of open source hardware such as Arduino, micro: bit. It is not only suitable for primary and secondary school students, but also able to provide a great learning environment for makers who want to improve themselves by studying high-level Programming language such as Arduino, python, c, c++ etc. Drag and combine code blocks to make programs, easy to find the joy of creating.

From beginners to experienced makers, Mind+ can satisfy all your needs: learning Programming, running programs without downloading, experiencing IoT(Internet of Things), sharing ideas with Mind+ community and so on.

Come and play with Mind+, more surprises waiting for you!

1.1 Download and Install Mind+

STEP1: Download Mind+ Programming Software (The tutorial is based on Mind+ V1.1.2)

Open the webpage and download: http://mindplus.cc

Figure 1.1.1 Mind+ Download Interface

STEP2: Install Driver

After downloaded and installed the software, open and click "Learning" to learn how to install driver.

1. Click to open Learning ->Video Tutorials.

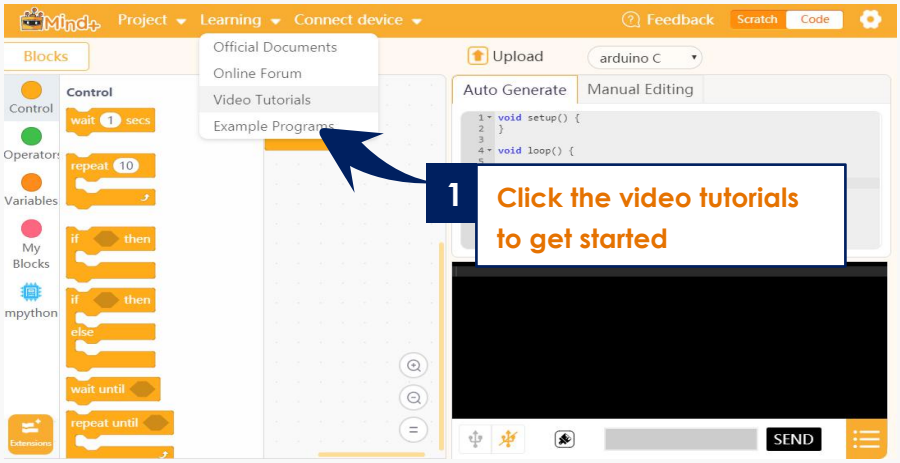


Figure1.1.2 Video Tutorials

2. Click the related tutorial, and play the video.

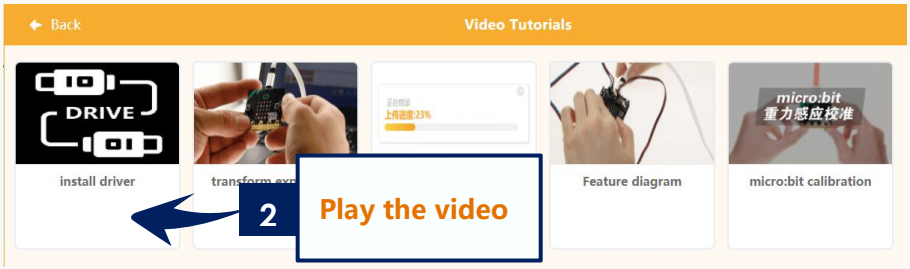
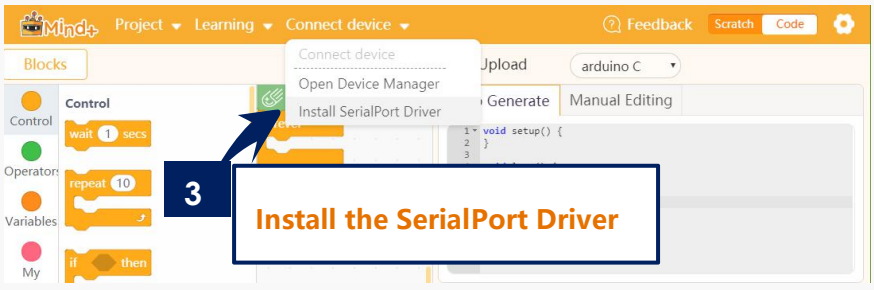


Figure 1.1.3 Click to play the video

3. Install the SerialPort driver according to the instruction.

   And you just need to install the driver for once.



1.1.4 Install SerialPort Driver

STEP3: Switch to "Code" mode (The tutorial is based on "Code" mode).

1. Click "Code" icon at the right corner to switch mode.

2. Switched to "Code" mode successfully.

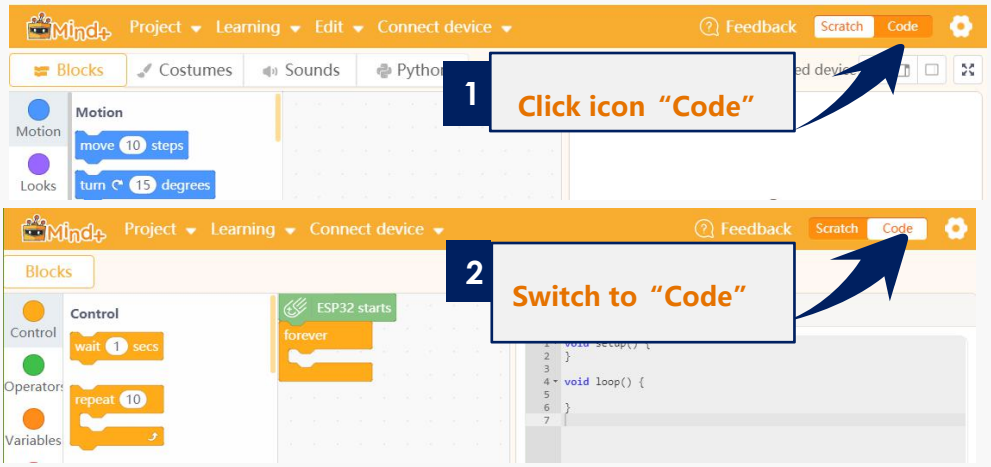
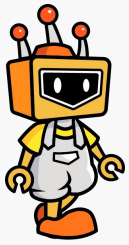
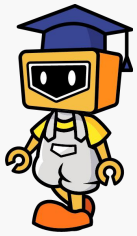


Figure 1.1.5 Switching mode



Dr.M, what's the difference between these 2 modes?

My little friend, the mode **Scratch** will execute script programs in hardware and Mind+ stage in real time. But the mode **Code** will upload programs to hardware devices first then execute.



1.2 Mind+ Interface

To convenient the operation, first we should be familiar with the basic functions of the software's interface. If we compare the Mind+ software as a stage, what are the functions of its different sections?

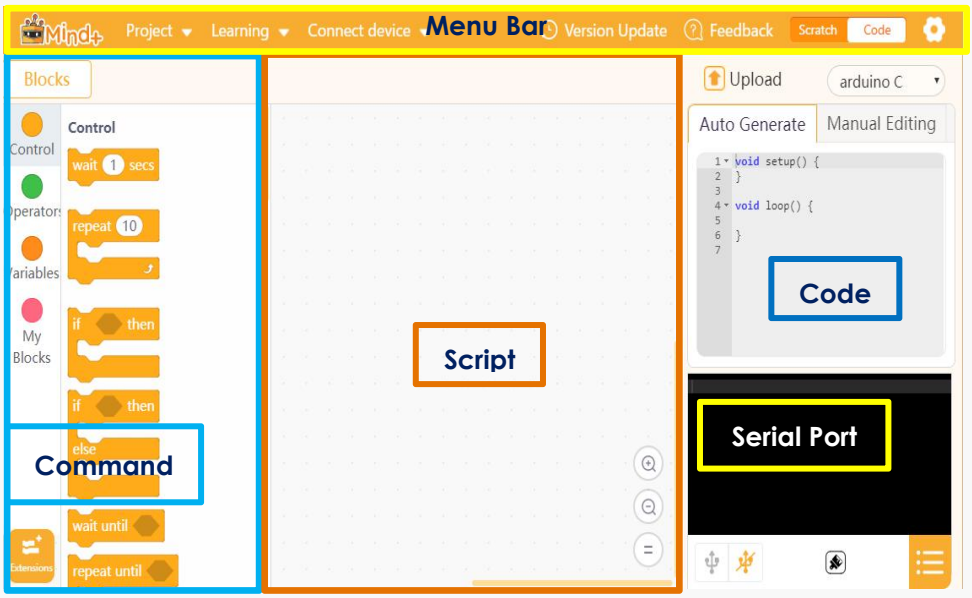


Figure 1.2.1 Mind+ Interface

Menu Bar: Software setting area.
This is the backstage of the whole "stage". Without the help of the menu bar, we can do nothing with this software. What functions are there behind the "stage"?

"Project": create a new project, load a project, save a project, save a project as.

"Learning": where you can find lots of useful tutorials and examples programs.

"Connect Device": detect the connected device, and you can select to connect or disconnect the device.

"Scratch/Code" icon: switch mode to run program.

" Setting" icon: set the theme and language of the software; share and learn on our community or E-mail us your feedback.

Command: This is the "tool section" of the stage. We need all kinds of tools to give an excellent performance on the stage, and we can get more extra tools in "Extensions" to support different hardware.

Script: Here is the core of the "stage performance". All performances will be acted under the instruction of "script". This is what we called graphical Programming, easy and simple for beginners to understand. Just drag the code blocks in the "Command" to here then you can program.

Code: If you want to check the codes of the blocks in the "script", this is the right place.

Serial Port: How do you know whether you have made a successful performance? -Interact with your audience. This is the place your audiences give you their "opinion" about the performance. The downloading situation will be displayed on there, so you can check whether your download is successful, or if not, what's the reason. Besides, you can see the programs running here. The serial port communication data will also be displayed. That is to say, if your micro: bit board is connected with an external sound sensor, you can find the related data here. This section also includes: serialport open/close, scroll display open/close, clear output, baud setting, serialport input and output format control.

2. Micro: bit

Micro: bit is an open source hardware ARM-based embedded system designed by the BBC. It has accelerometer, magnetometer, Bluetooth, three buttons and a display consisting of 25 LEDs.

With micro: bit, you can create many interesting and cool things, for example, use it as a mini-mobile phone; connect the board with phone through onboard Bluetooth, then send commands to control the board by the phone; realize telecommunication of multiple micro: bit boards via wireless connection to finish tasks requiring long-distance control; use magnetometer to make navigation device….

Micro: bit is equipped with light and temperature sensors and other common sensing devices, which means that it itself can produce a lot of common smart products used in our daily life.

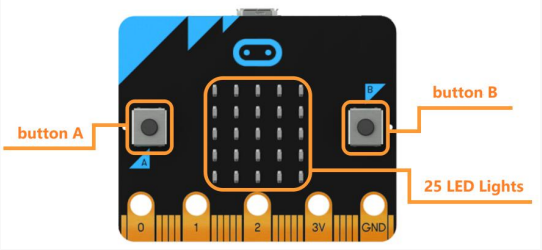


Figure1.2.1 Front of micro: bit

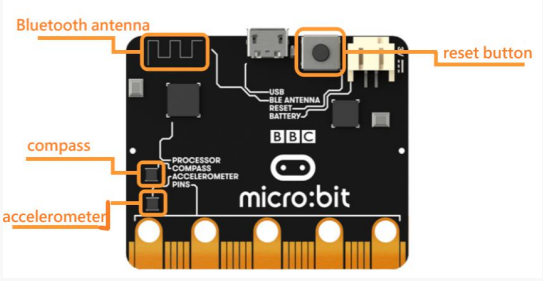


Figure 1.2.2 Back of micro: bit

micro: bit and micro: bit Expansion Board

Limitation of micro: bit

1. micro: bit cannot directly drive devices requiring large currents like, motor, servo etc. due to its poor drive ability, so edge connectors are necessary for connecting more external devices.

2. The power supply and signal voltage of micro: bit is 3.3V, which cannot work together with the common 5V electronic components.

The advantages of micro: bit expansion board

1. micro: bit expansion board can easily lead out the extra pins of micro: bit, plug and play, easy to use;

2. The expansion interfaces of micro: bit improve the safety and stability of connection.

3. Micro- USB external power port VIN can be used to provide extra power for external electronic devices such as LED strip, servo etc. (USB wire still needs to be connected to micro: bit when running the programs.)

4. Micro: bit expansion board comes with 3.5mm headphone jack and volume knob, supporting direct connection of headphone.

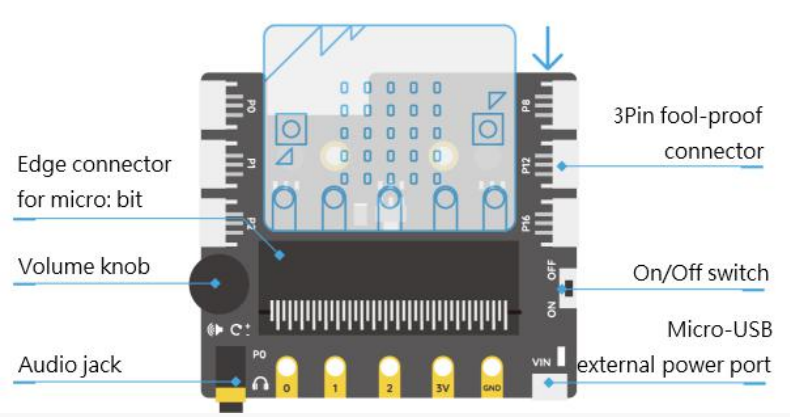


Figure 1.2.3 micro: bit expansion board

The maximum rated current of micro: bit expansion board is 500mA. Please confirm the max power of your component. Do not use the expansion board together with high-power servo, otherwise on-board units damage may occur.

3. Get Started with Mind+ and micro: bit

Please prepare a Windows computer and components as below before we get started:

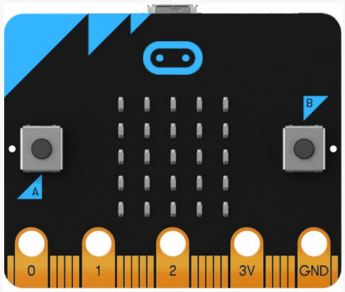
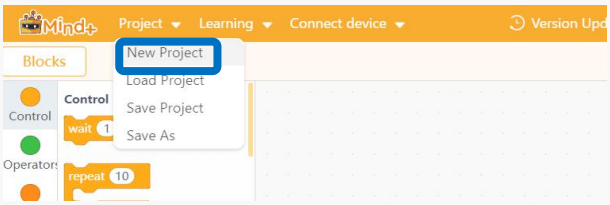


Figure 1.3.1 micro: bit

Figure1.3.2 micro USB connector

STEP1: Select micro: bit

1. Click "Project"-> "New Project" at the upper-left corner.



1.3.3 Create a new project

2. (1) Click "Extensions" at the lower-left corner
   (2) Click "Board"
   (3) Click "Micro: bit"

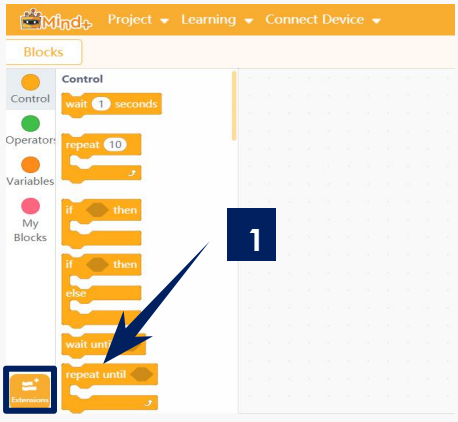


Figure 1.3.4 Extensions

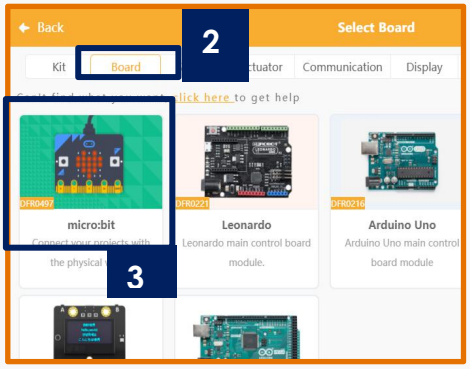


Figure1.3.5 "micro: bit"

STEP2: Edit a simple program

Edit a program to display a pattern on micro: bit LED panel:

1. Drag the blocks we do not need currently to “command area” to remove it.

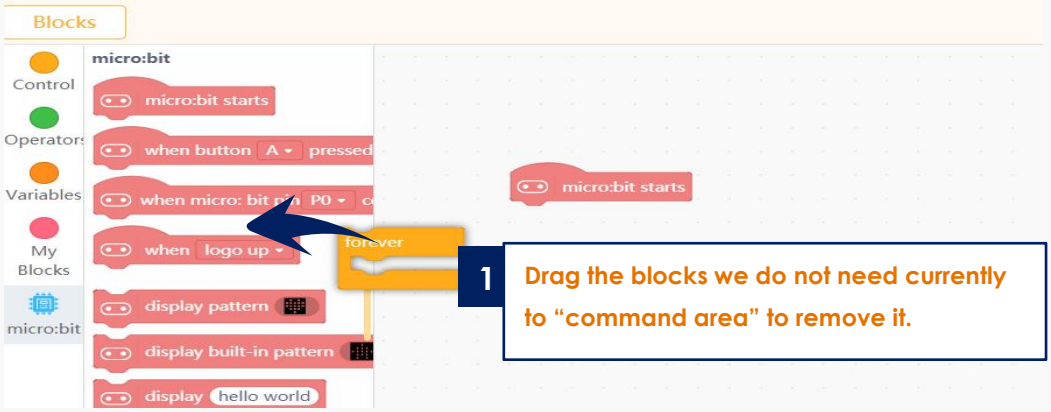


Figure1.3.6 Remove Blocks

2. Click “micro: bit” module

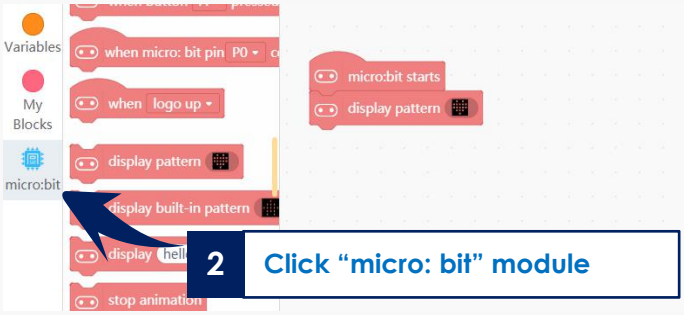


Figure 1.3.7 Click “micro: bit” module

3. Drag “display pattern” block onto “Script” area.
4. Place “display pattern” block under the “micro: bit starts” block

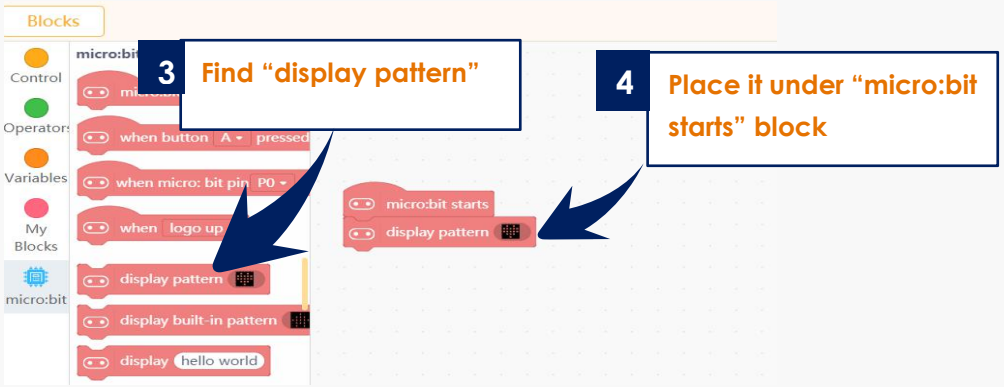


Figure1.3.8 Programming

STEP3: Upload to device

1. Connect micro: bit board to your computer via USB, then the power indicator on the board will light up.

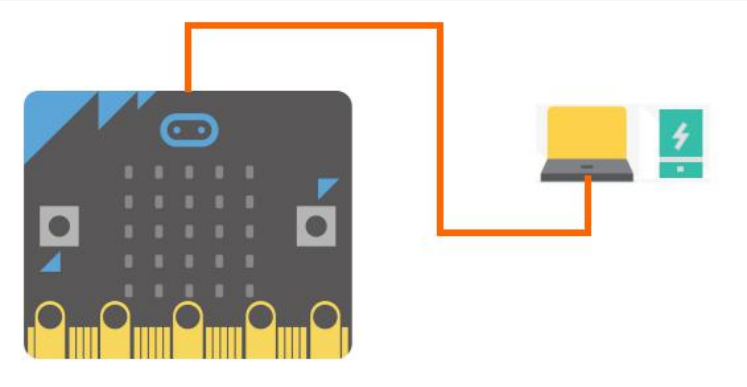


Figure 1.3.9 Connect micro: bit to computer

2. Click "Connect Device" in the menu bar, then click "COM-micro: bit" in the drop-down menu to connect micro: bit board to the computer.
3. When connected, edit your programs and click "Upload" to upload the programs to your board (if the connection fails, then it will prompt "No Device Connected". There will be a prompt box to display the uploading progress. When the program is updated, the prompt box will disappear.

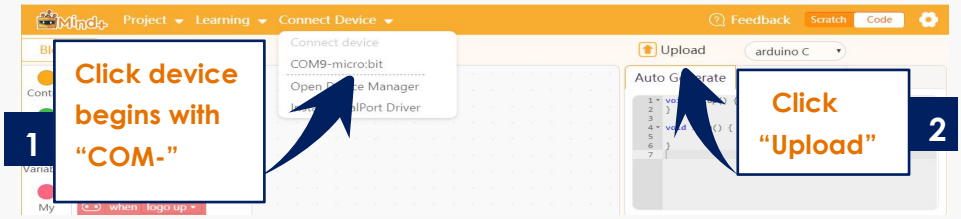


Figure 1.3.10 Run the program

micro: bit power indicator flashes only when uploading the program, and stops flashing and keeps ON after the upload is completed.

Try uploading a program to your micro: bit board, and make the board display an "love heart" for you!

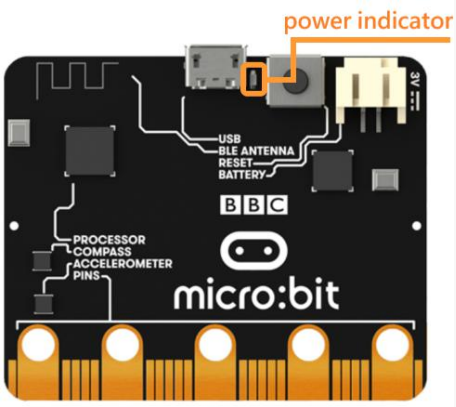
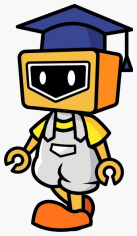


Figure 1.3.11 Power Indicator

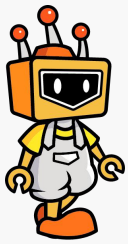
Little Tips

The basic usages of micro: bit was shown above. If you have any quires, please feel free to contact us on our forum at **https://www.dfrobot.com/forum/**

Welcome to our DFRobot Community to find more tutorials and fantastic projects, and of course you can also share your own projects or ideas on our community. Looking forward to your log in!



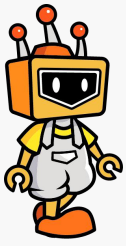
How do you feel about these learning? Have you improved yourself?



Of course! Mr. DFRobot, now I have a basic knowledge about Mind+ and micro: bit.

And I can make a red heart!

You are so smart! This is just a begin, we will get to know more about Mind+ and micro: bit in future. And you can use them to do what you want.



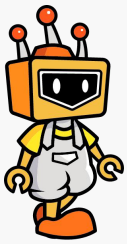
Awesome! I will try my best.
Dr.M, please teach me.

Chapter 2: What Makes a Machine "Come to Life"

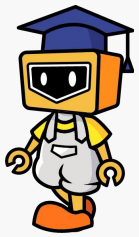
After reading the previous introduction, are you itch to have a try? Now, let's begin our journey with micro: bit!

Project 1: The Mysterious micro: bit

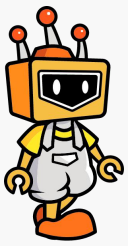
1.Story of Micko and Dr.M



Dr.M, I have found many emojis in computer and cellphone. Like these, how adorable they are!

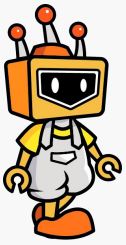


My dear, if you like it you can make your own emojis.



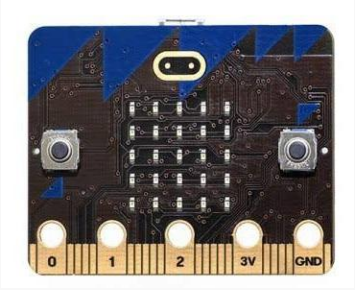
Wow! Dr.M, please help me make it. Then I can share it with my friends.

Do you remember the new friend micro: bit we met in the last chapter? It will help you make it.

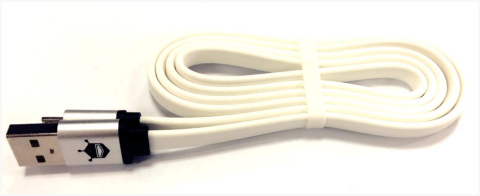


Yes, I remember. I cannot wait to make it.

2.Dr.M's Pocket



1 × micro: bit

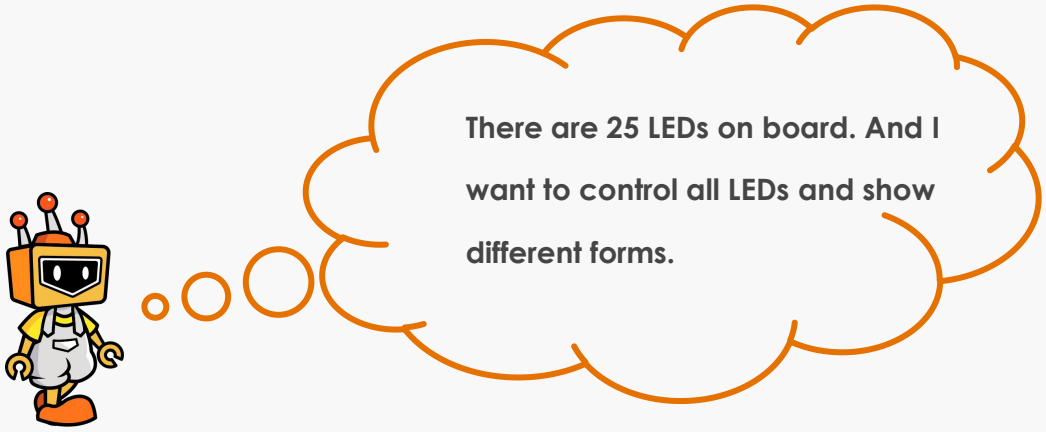


1 × micro USB Cable

Little Tips

All necessary components are in the pocket of Dr.M.

3.Dream of Micko



4.Dr.M's Blackboard

STEP1 : Connect micro: bit to your computer via USB cable.

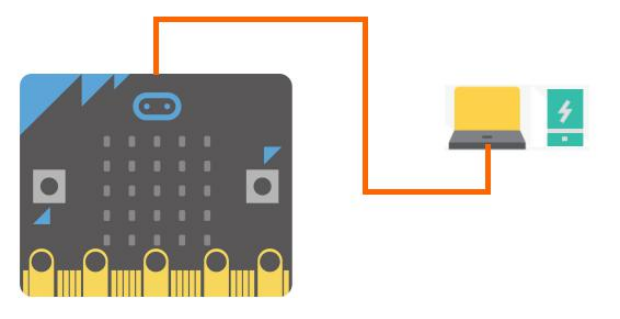


Figure 2.1.1Connect micro: bit to Computer

STEP2: Programming

1) Open Mind+ software, click Project->New Project, and then select "micro: bit" board from "Extensions"

Let's review the steps to select "micro: bit" board:
(1) Click "Extensions" at the lower-left corner
(2) Click "Board"
(3) Click "micro: bit"

2) Click "micro: bit" at the left side of command area, and find "micro: bit starts" block, which is very necessary for editing your program.

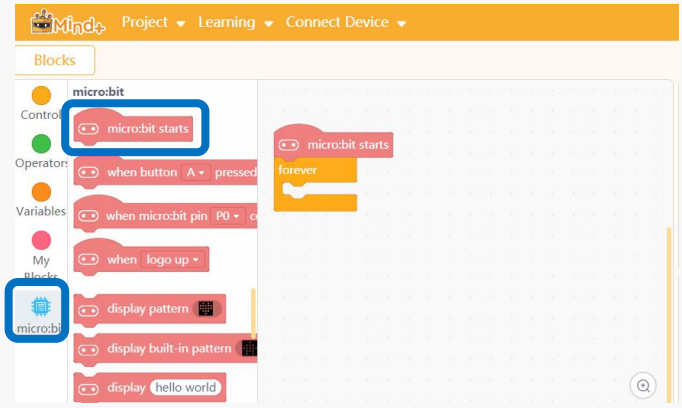


Figure 2.1.2 micro: bit starts

3) Drag "forever" block to the left to delete it.

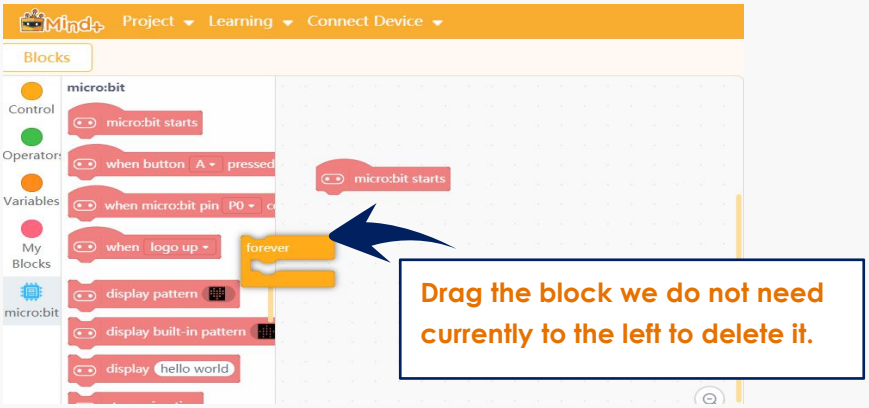
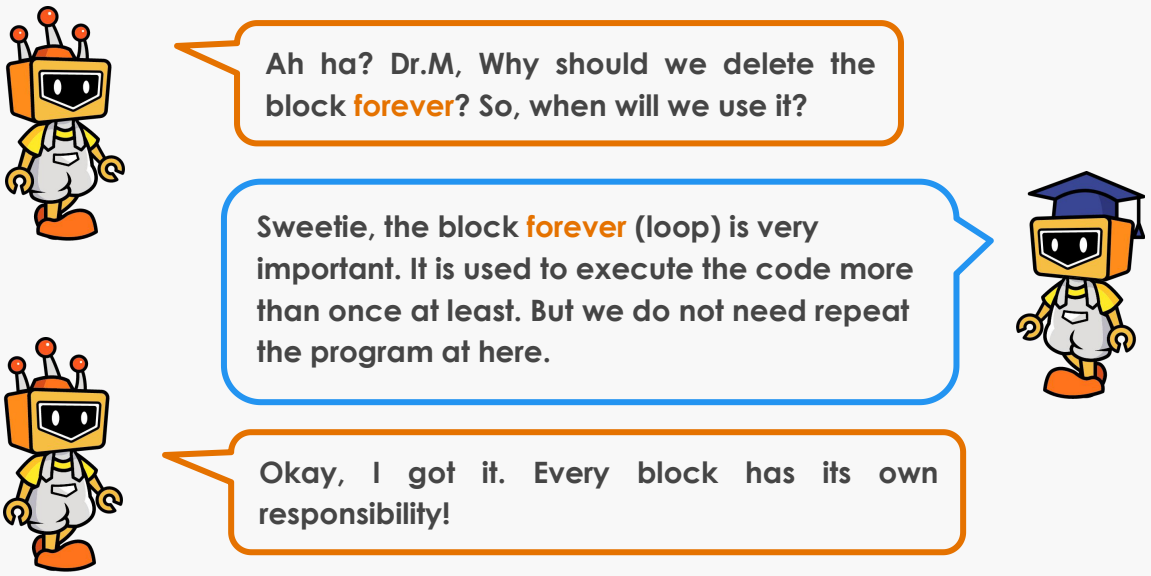


Figure 2.1.2 Delete Blocks



4) Click micro: bit module, and find "display pattern" block. Drag the block to script area and place it under "micro: bit starts" block.

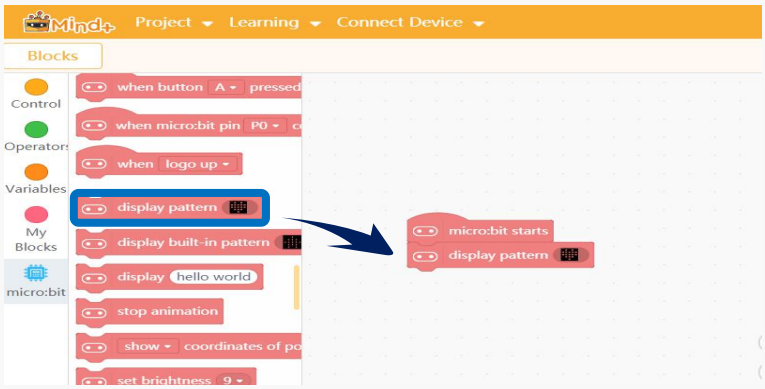


Figure 2.1.3 Drag Blocks

5) Click the black square in the "display pattern" block we dragged to script area, then we can start to design our own pattern. First, let's make a super cute "love heart".

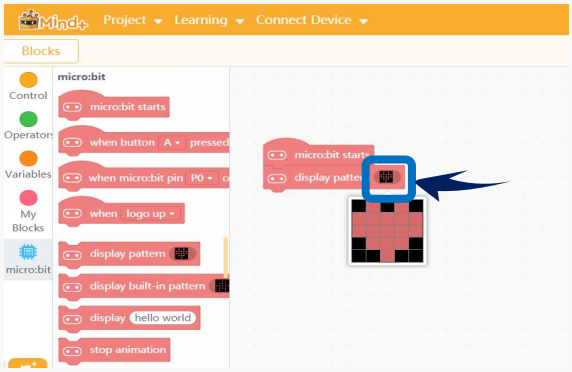


Figure 2.1.4 Design a heart pattern

6) Click "Upload" to upload the program to micro: bit board. Then, you will find a "heart" on micro: bit LED panel.

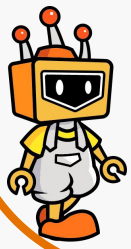
Remember to connect the device before you "Upload": Click "Connect Device" in the menu bar, then click "COM-Micro bit" in the drop-down menu to connect micro: bit board to the computer.

5. Design of Micko

Try making one virtual pet, imaging what it looks like?

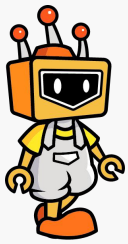
Look Design

Program Design



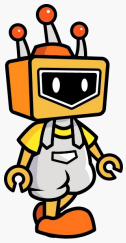
Project2: Flash the LED Light

1. Story of Micko and Dr.M



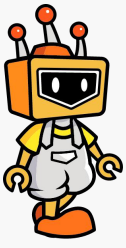
Dr.M, the lovely little heart made last week is very interesting, my classmates love it!

My dear, not just emojis, playing mind+ and micro: bit can make as much as you can imagine.



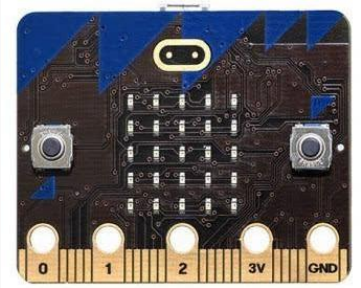
Really?! Mr DFRobot, colorful lights are my favorite things. Could I bring them home?

It's my pleasure to teach a student who is fond of learning. Okay, let's welcome a new friend-LED lights.

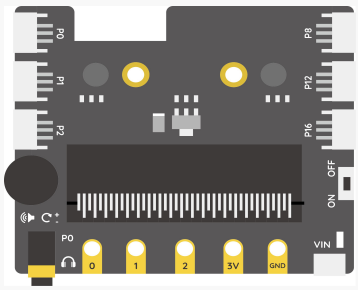


Fantastic! Let's do it now!

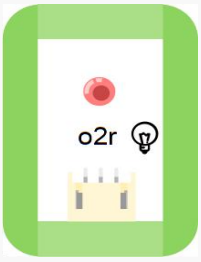
2.Dr.M's Pocket



1 × micro: bit



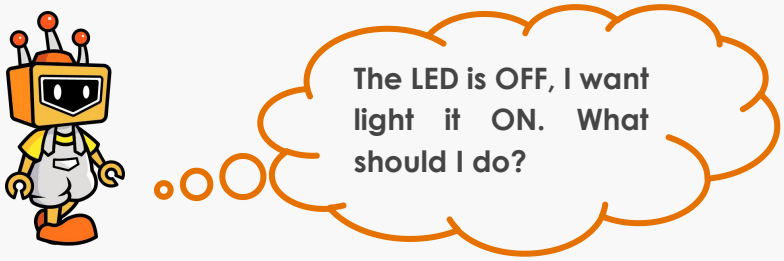
1 × micro: bit Expansion Board



1 × LED Module

1× micro USB Cable

3.Dream of Micko: Light up LEDs



4.Dr.M's Blackboard

STEP1: Connect micro: bit board to micro: bit expansion board.

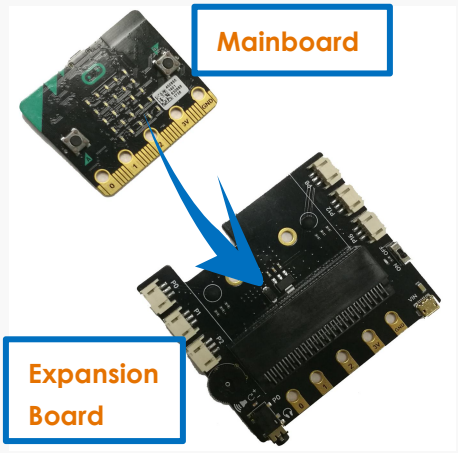


Figure 2.2.1Connection Method

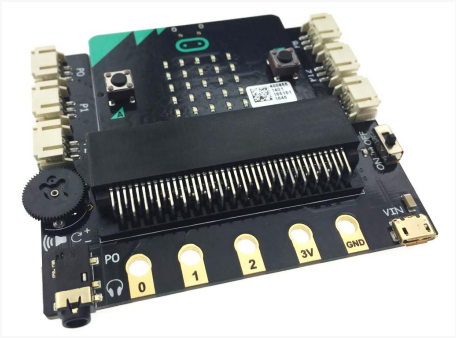
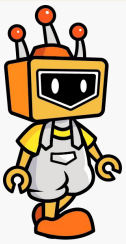


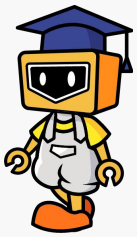
Figure 2.2.2 Connection Diagram

Note: the following projects we will learn are based on the combined use of micro: bit board and micro: bit expansion board.



Dr.M, Why should we use these two boards together? Could we only use the main control board micro: bit?

Well, the reason why should we use the expansion board has already explained in the introduction of micro: bit.



STEP2: Connect the LED module to pin P1 of the expansion board.

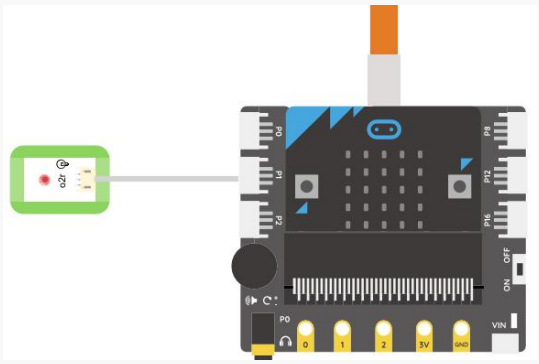


Figure 2.2.3 Connect LED

STEP3: Programming

1. Open Mind+ interface, create a new project and select micro: bit board in extensions. Scroll down the "micro: bit" blocks to find "set digital pin P0 output LOW" block, drag it onto script panel. Then we need to change "P0" to "P1", "LOW" to "HIGH".

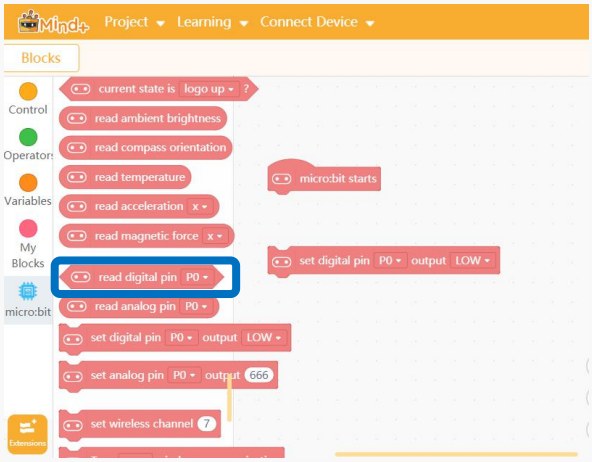
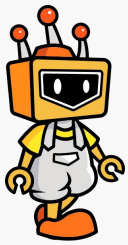
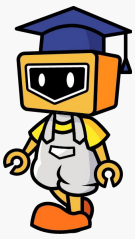


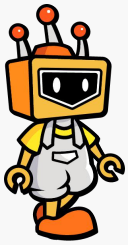
Figure 2.2.4 Set Digital Pin



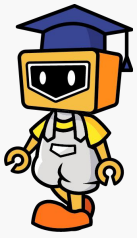
Dr.M, I am confused. May I ask you some questions?



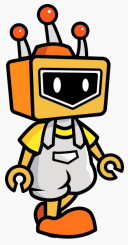
Okay, take it easy. If you have any questions, feel free to ask me.



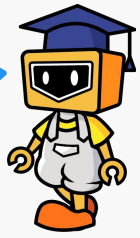
The 1st question: Why should we change P0 to P1? Will we change the pin to P1 every time?



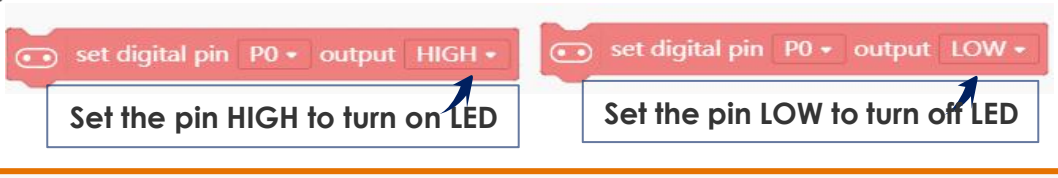
Little friend, pins are the way that module connected to main board. Then we can control the module by change pin value. In this project, the reason why we change it to P1 is that the LED is connected to P1.

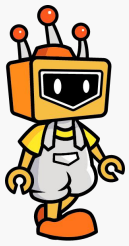


I see, so if module is connected to P2 then we need to change the pin number to P2.
The 2nd question: What is high level and low level?

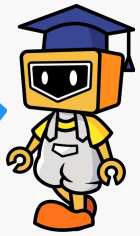


At first, we need to know when the digital pin output low level, which means no voltage or low voltage, and cannot reach the minimum voltage value to light ON the LED. But when the digital pin output high level, which means it can reach the minimum voltage value to light it ON. So, the LED is ON.





Well, the high level is corresponding to light ON LED, and the low level is corresponding to light OFF LED.
The last question: Why should we choose digital pin? But not analog pin?



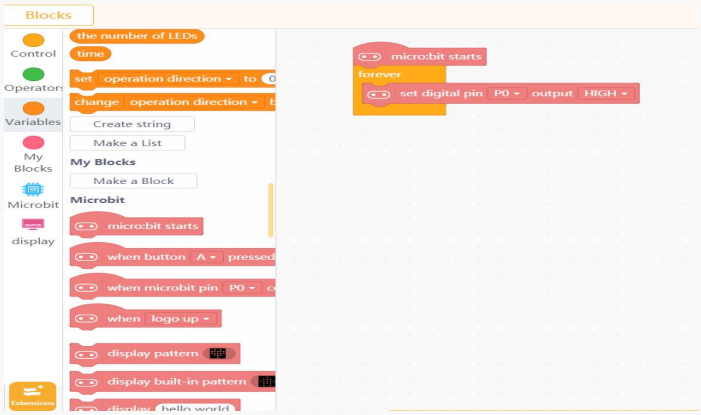
How observant you are!

The digital pin is used to control digital signal. But what is the digital signal?

Take LED as an example, it only has 2 modes: ON and OFF, which correspond to two modes of digital signals, high level and low level. But the eye-protecting lamp, its brightness is adjustable. We can change the brightness as we want. Here the way we control the lamp brightness used the analog pins.

Thank you! I will think it over.

2. Drag the "forever" block out from "control" module, nest "set digital pin1 output HIGH" block in forever(loop) statement, and place them under the main program. Then "upload" the program to your device.



2.2.5 Execution Interface

3. Result: the LED will be on when the code is uploaded successfully.

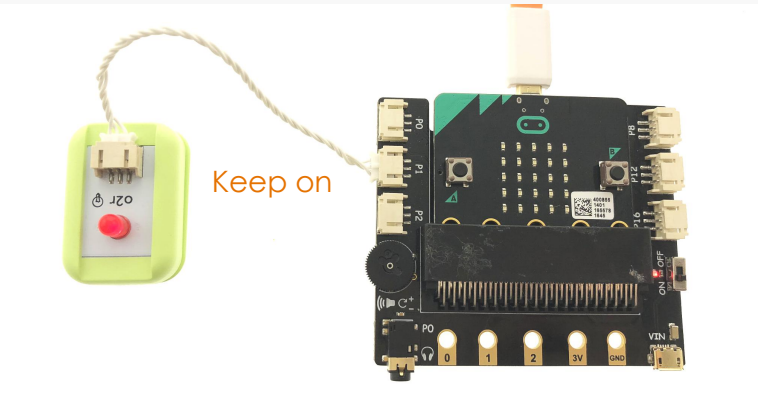
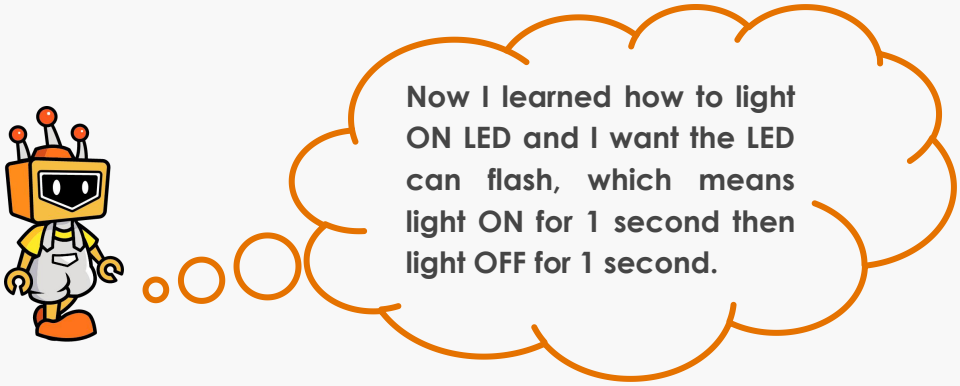


Figure 2.2.6 Result

5.Dream of Micko: Flash the LED Light

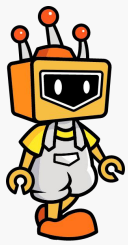


6.Dr.M's Blackboard

1) The connection is same as above. Create a new project to make the LED flash.

2) How to flash the LED? Just let the LED repeatedly keep on for 1 second and keep off for 1 second. To achieve that effect, we need to use "wait 1 sec" block in "control" module.

Figure 2.2.7 "wait 1 sec"



Dr.M, I remember! Here we can use "forever" (loop) to keep execute the program.

Good memory! Yes, the loop block 'if…then' is used to execute program again and again.

3) Place "wait 1 sec" block under the "set digital pin P1 output HIGH" block so as to let the LED keep on for 1 second then do the next action.

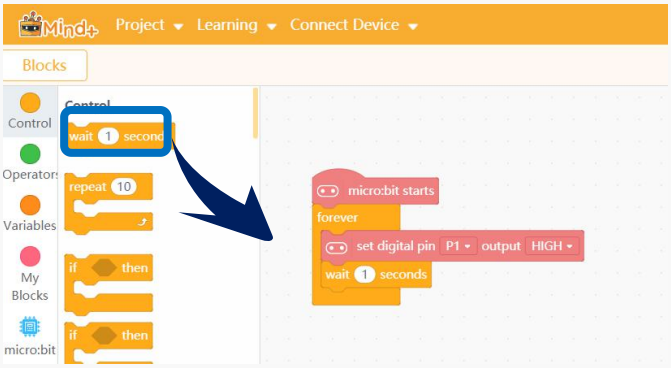


Figure 2.2.8 Drag "wait 1 sec" Block

4) Next, we need to let the LED keep off for 1 second. Right-click "set digital pin P1 output HIGH", then click "Duplicate".

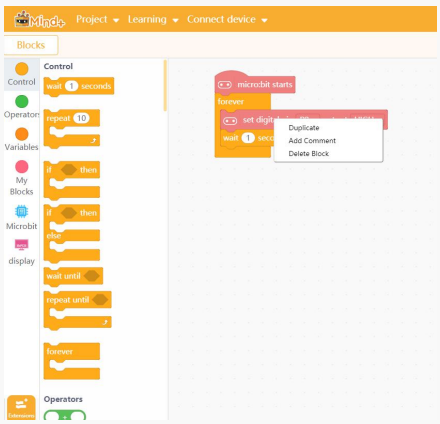
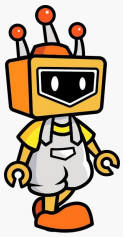


Figure 2.2.9 "Duplicate" Blocks



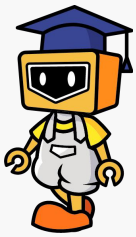
Cool! Click the right mouse can duplicate the blocks. It is very convenient!

5) Place the duplicated blocks under the block "wait 1 sec", change "HIGH" to "LOW" to let the LED keep off for 1 second. Since we used forever statement here, the program of LED flashing will be running all the time.

Figure 2.2.10 Run the program

Result: the LED keeps flashing when the code is uploaded successfully.

7.Design of Micko



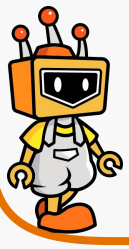
Sweetie, I heard that you will go camping with your friends. I am worrying you may get lost outside.

How about make one emergency device SOS?

Reminder: How to send emergency signal SOS via LEDs?
To send out SOS for help, we just need to flash LEDs according to the rule 3 shorts, 3 longs and 3 shorts again.
3 longs are presented by 'ON---OFF---ON---OFF---ON---OFF---' and 3 shorts are presented by 'ON-OFF-ON-OFF-ON-OFF- '. Notice it? The time that 3longs costs are 3 times of 3shorts. Let's do it!

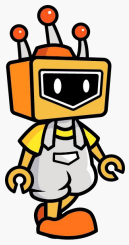
My Design

My Program



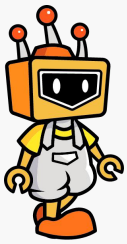
Project 3: Notification Light

1.Story of Micko and Dr.M



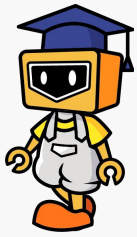
Dr.M, I will flying to DF Starship with my friends this week, but I am a little afraid of dangers.

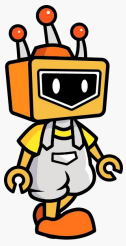
Eh, it will cost several hours at plane. You guys should prepare some emergency equipment.



Yes, Dr.M, if the plane goes off course and we got lost in the vast cosmos, what should we do?

My dear, do not be frightened. Once you have be noticed by other spaceships, you will be safe.

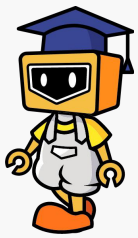




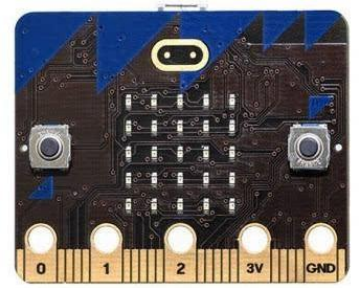
Oh! We can make a notification light. Its brightness can be adjusted from the maximum to the minimum. If any dangers, we can adjust the light to the maximum brightness. And others can easily find us!

Dr.M, could you please help me make it?

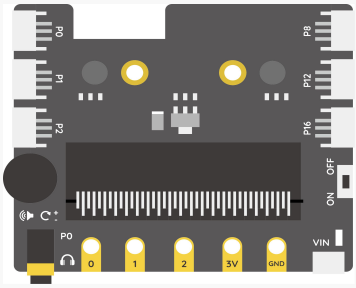
No problem! Let's make one notification light.



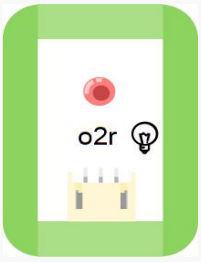
2.Dr.M's Pocket



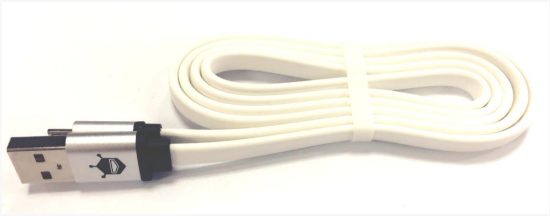
1 × micro: bit



1 × micro: bit Expansion Board

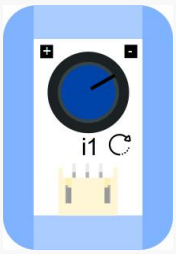


1 × LED Module



1× micro USB Cable

1 × Push Button



1 × Knob

Know something about Push-button and Switch Knob:

Push-button is the common button we can find on lamp. By programming, we can realize to turn on/off the LED through the Push-button.

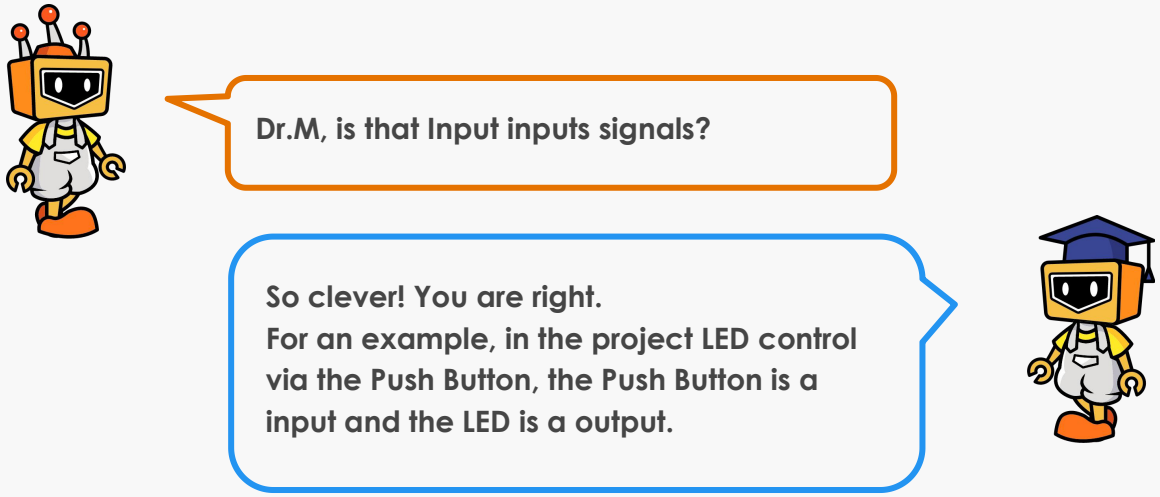
Switch the knob to different angles, we can control the brightness of the LED.

What's the differences between Push-button and Knob?

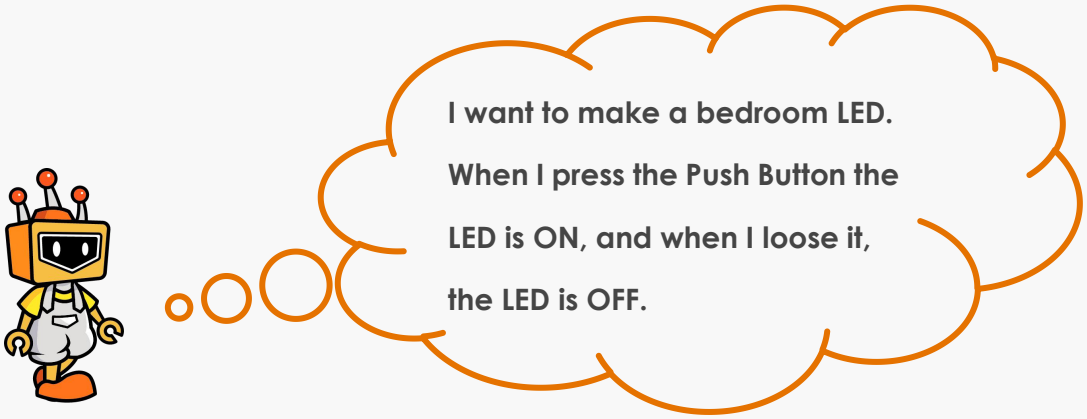
Both are input modules.

The electric circuit can only be connected when the Push Button is pressed.

And the electric circuit can be connected when the Knob rotates within one range.



3.Dream of Micko: Control LED by Push Button



4.Dr.M's Blackboard

STEP1: Connect Push-button module to pin P0 of the expansion board; connect the LED to pin P1.

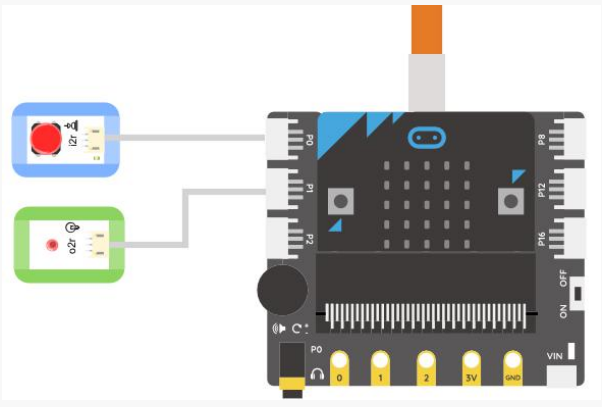


Figure 2.3.1 Connection Diagram

STEP2: Programming

1) Click "micro: bit" module, find "read digital pin P0" block and drag it to script area.

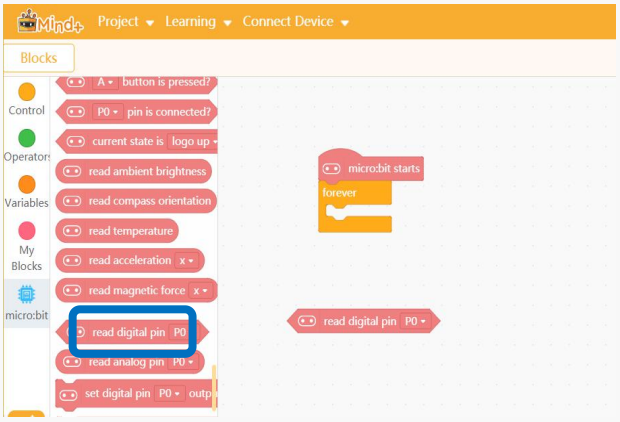
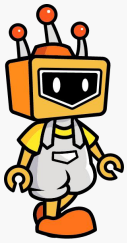


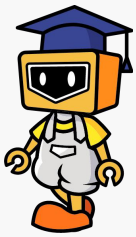
Figure 2.3.2 read digital pin P0 block



Dr.M, why not use high level and low level at here?

Sweetie, because we need to check the P0 value to judge whether the Push Button is pressed.

When the Push Button is pressed, the digital value read by Mind+ is 1, which means high level and the LED is ON.
Well, when the Push Button is loosen, the digital value read by mind+ is o, which means low level then the LED is OFF. So it is certain to be used in the future, but not now.



2) The "=" in the "Operators" module can be used to judge if the push-button is being pressed. If the value of "read digital pin P0" is "1", the button is pressed, otherwise, it is unpressed. Nest the block "read digital pin P0" into the equation.

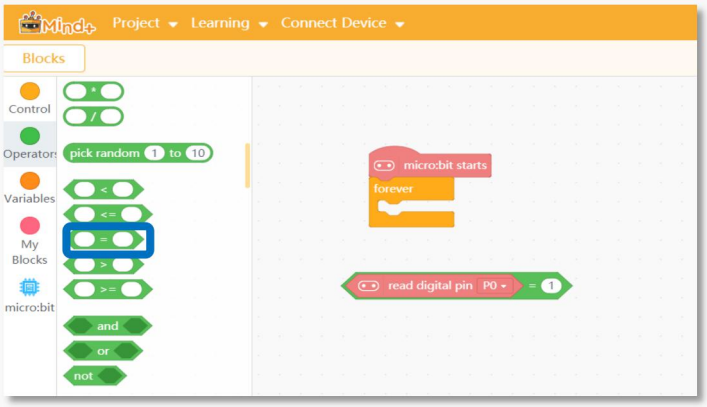
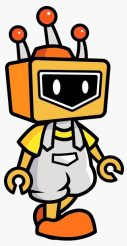


Figure 2.3.3 "=" Block



Dr.M, how to judge whether the Push Button is pressed?

Good question! We can judge it by the condition judgement command "if…then…else". But how? Go on reading and you will see.

3) Find **"if then…else"** block in the **"Control"** module, and drag it into the "forever" block.

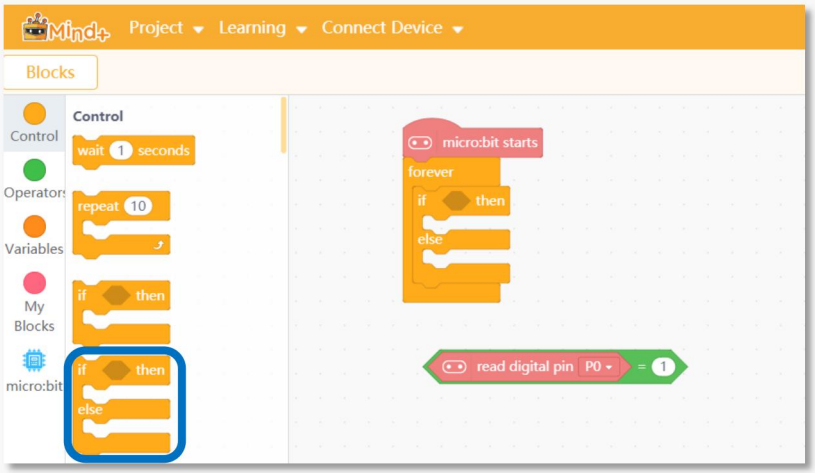


Figure 2.3.4 "If then…else" Block

4) Nest the conditions of equation into the "if then…else" block as shown below:

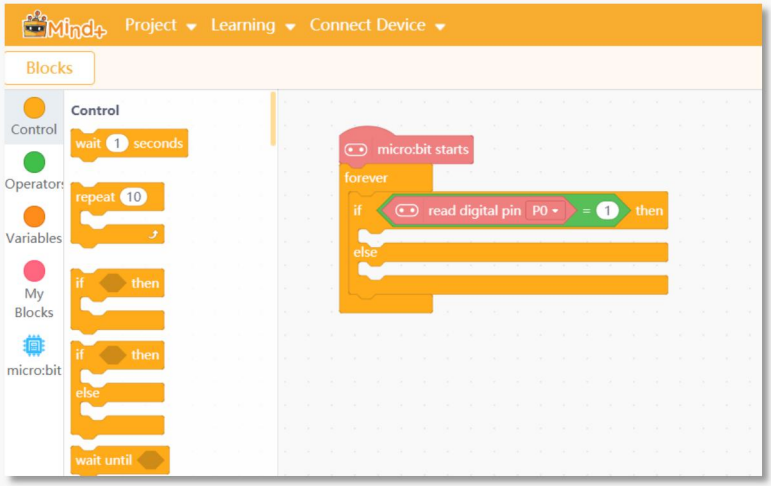


Figure 2.3.5 Nest Blocks

5) Find the block "set digital pin P0 output LOW" in the "micro: bit" module. Revise the block into "set digital pin P1 output HIGH" and place it under "if then" block. When the button is pressed, this command will be executed and the LED will be on. Then place "set digital pin P1 output LOW" block under the "…else" block. When the button is released, the LED will be off.

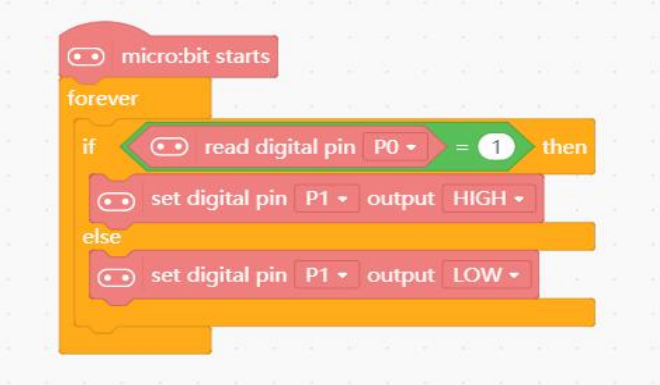
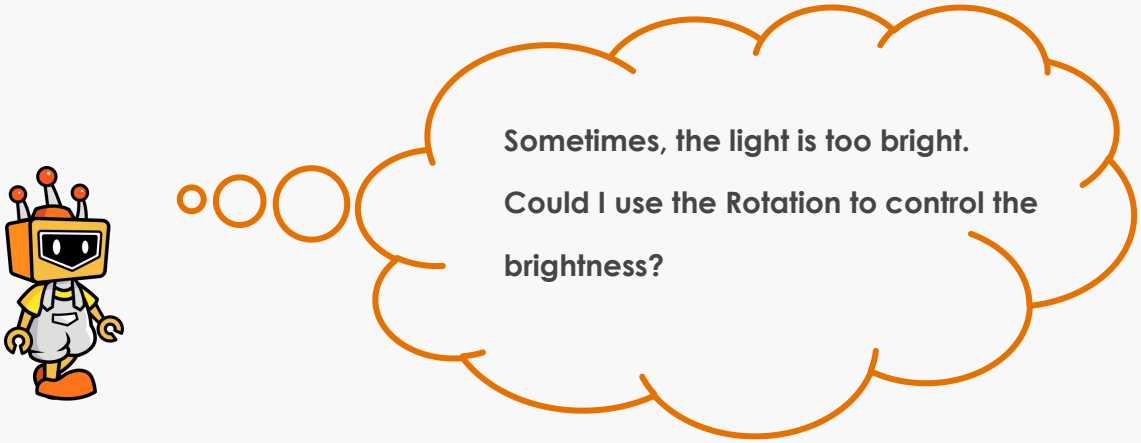


Figure 2.3.6 Run the Program

5.Dream of Micko: Control LED by Knob



6.Dr.M's Blackboard

STEP1: Replace the push-button module in the last project with switch knob module.

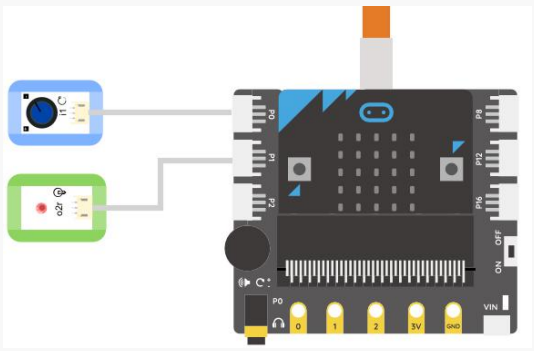


Figure 2.3.7 Connection Diagram

STEP2: Programming

1. Connect the switch knob to pin P0. Click "micro: bit" module and select "read analog pin P0" block.

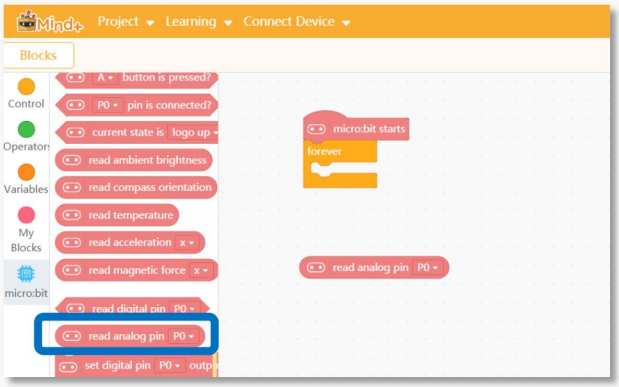


Figure 2.3.8 "read analog pin P0" Block

The brightness of LED is decided by the value of the analog pin (knob's angle). Switch the knob to control the LED's brightness. Micro: bit "read analog pin" (knob) ranges from 0 to 1023, so does the output analog pin. So, it's ok to read the analog pin P0 directly!

2. Drag "read analog pin P0" to the script area.

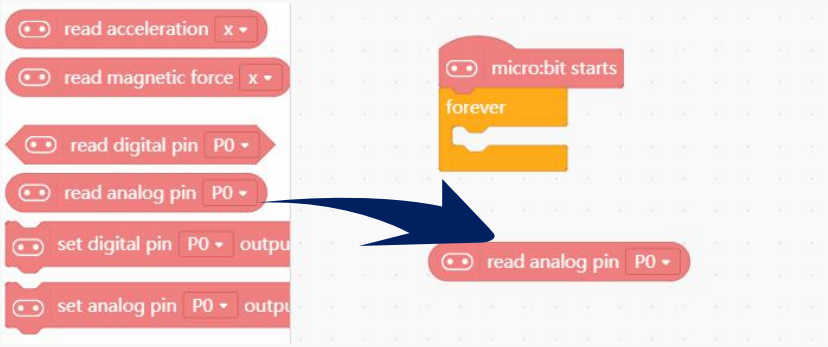


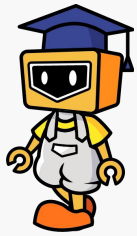
Figure 2.3.9 "read analog pin P0" Block

3. Drag "set analog pin P0 output" block of "micro: bit" module into "forever" block in script area, and change P0 to P1. Nest the "set analog pin P1 output" block with the block "read analog pin P0".

Figure 2.3.10 Run the Programs

We can understand the codes in this way: the value of the position the knob switched is the brightness of the LED.

7.Design of Micko



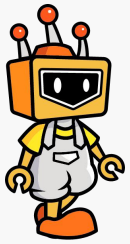
My dear, can you design a beautiful and practical rotation light now?

My Design

My Program

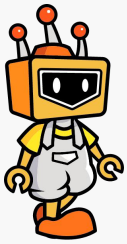
Project 4: Electric Fan

1.Story of Micko and Dr.M



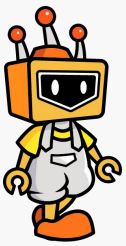
Dr.M, DF Starship is so interesting! With the inertia, we are floating in the outer space. I have seen lights and shades from the portholes.

Little friend, are you get used to the spaceship environment?



It's all right, except the high temperature in the cabin. We all felt very hot and wanted to have a fan.

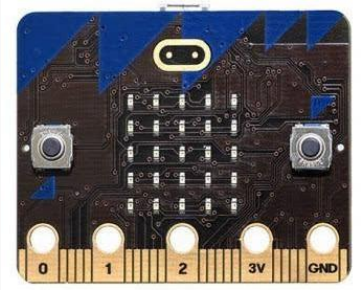
Poor you guys. Do you want to make one fan with adjustable wind?



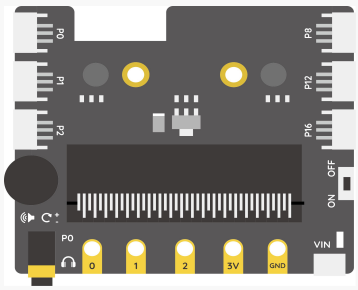
Yes, yes! I will not worry about hot weather if I have one. Dr.M, please teach me!

Sure, let's begin the class.

2.Dr.M's Pocket

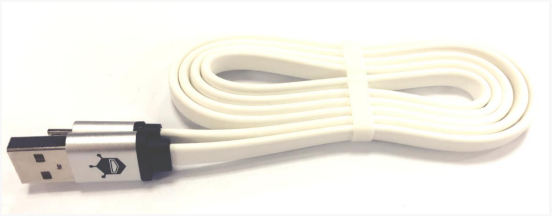


1 × micro: bit



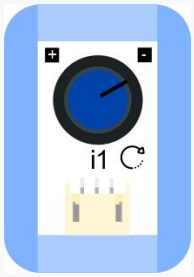
1 × micro: bit Expansion Board

1 × Fan Module



1× micro USB Cable

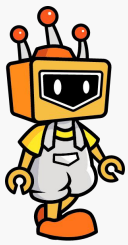
1 × Push Button



1 × Knob

The fan is similar to household fan, inside of which has motors and blades. The speed of the fan is adjustable.

3.Dream of Micko: Push-button Fan



I want a remote-control fan so that I can sit at sofa and conveniently turn the fan ON and OFF.

4.Dr.M's Blackboard

STEP1: Connect the push-button module to pin P0 of the expansion board; connect the fan module to pin P1.

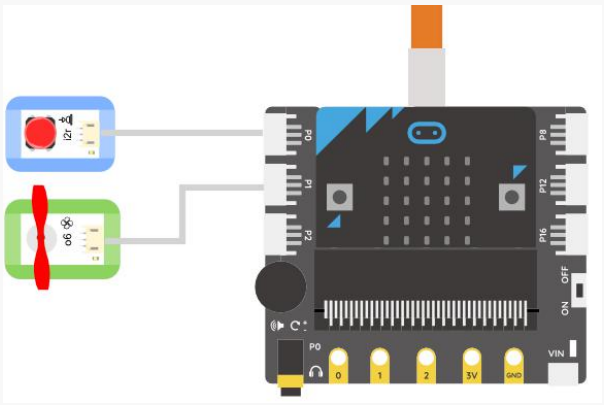
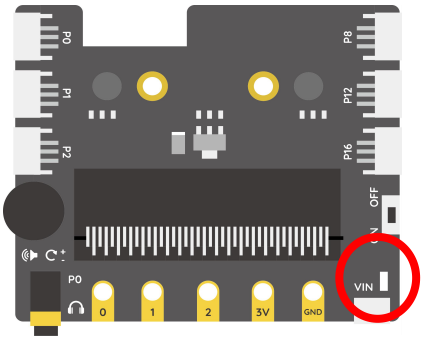


Figure 2.4.1 Connection Diagram

Let's see how to use the fan module:

Connect micro: bit with computer, debug and download your programs.

In the experiment the power consumption of fan and servo is large, so independent power supply (USB power supply) is necessary. Connect the external power to the VIN port of the extension board.



Detect the pin value (0 or 1) to judge if the button is pressed. When it is pressed (pin value=1), the pin connected to the fan is HIGH, and the fan starts to work. When the button is released (pin value=0), the pin connected to the fan is LOW, and the fan stops rotating.

STEP2: Programming

1. Click "micro: bit" module, select "read digital pin P0", and drag it to script section.

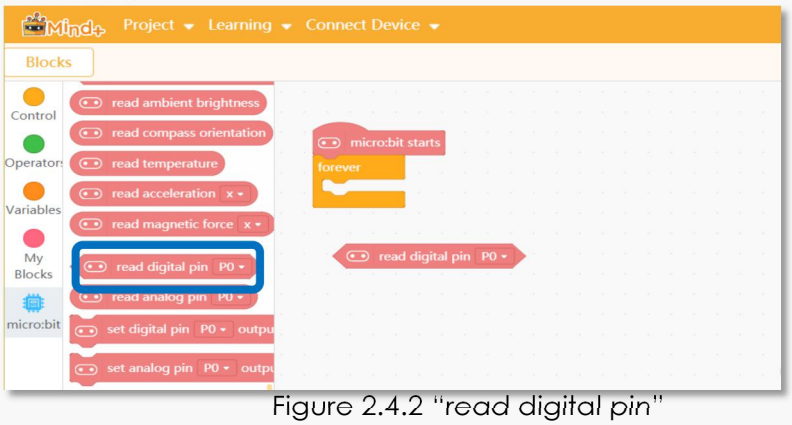


Figure 2.4.2 "read digital pin"

2. The "=" in the "Operators" module can be used to judge whether the button is pressed. If the value of "read digital pin P0" is "1", which suggests the button is pressed, otherwise, it is released. Nest the block "read digital pin P0" into the equation.

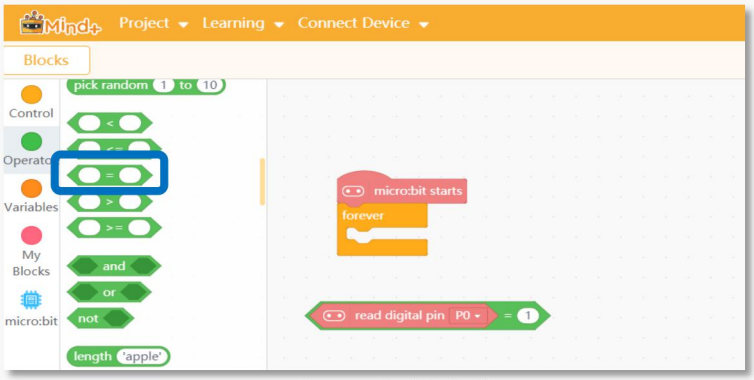


Figure 2.4.3 "="

3. Find "if then…else" block in the "Control" module, and drag it into the "forever" block.

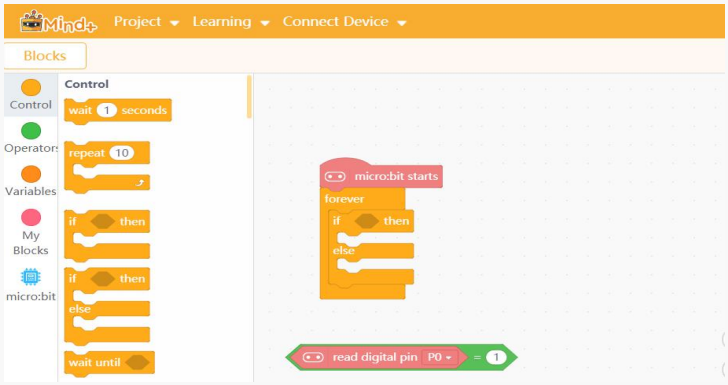


Figure 2.4.4 "If then…else"

4. Nest the conditions of equation into the "if then…else" block as shown below.

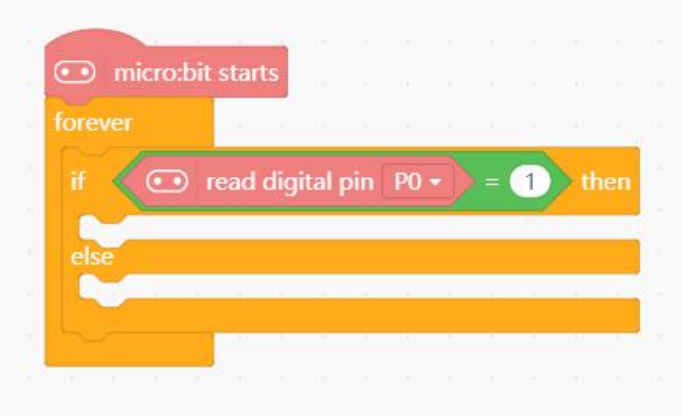


Figure 2.4.5 Nest Blocks

5. Find the block "set digital pin P0 output LOW" in the "micro: bit" module. Revise the block into "set digital pin P1 output HIGH" and place it under "if then" block. When the button is pressed, this command will be executed and the fan starts to rotate. Otherwise, the fan stops rotating.

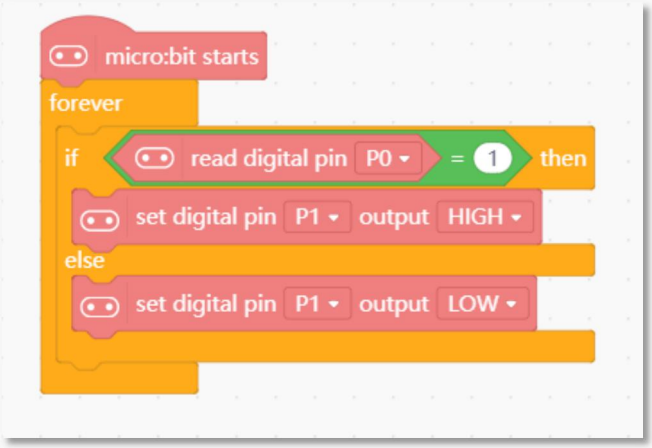
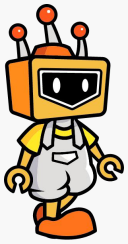
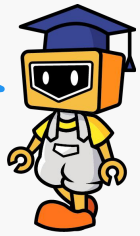


Figure 2.4.6 Run the program

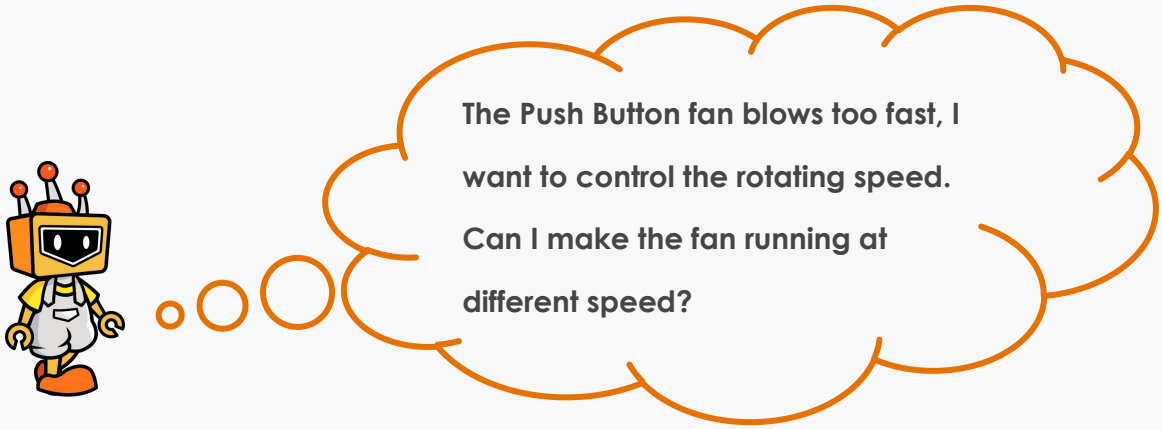


Dr.M, this program is similar to the former Notification Light.

How observant you are! Yes, the principle to control a fan is as soma as to control a LED.
It is so great that you can think and consider.
Keep this good habit!



5.Dream of Micko: Set the fan running at different speed



6.Dr.M's Blackboard

STEP1: Replace the push-button module in the button-fan project with switch knob module.

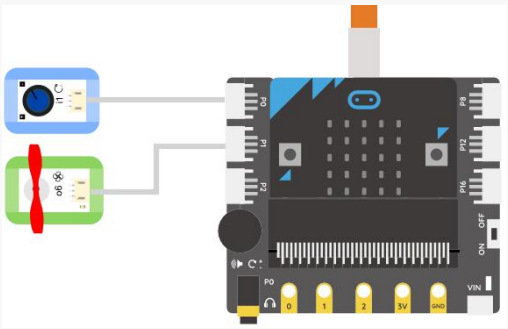


Figure 2.4.7 Connection Diagram

STEP2: Programming

1. Click "micro: bit", drag "read analog pin P0" block to script section

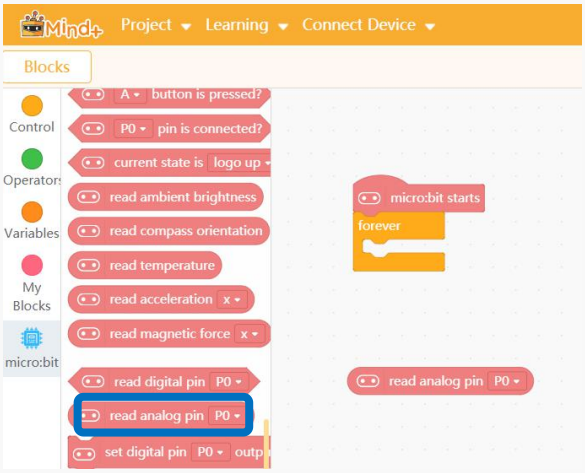
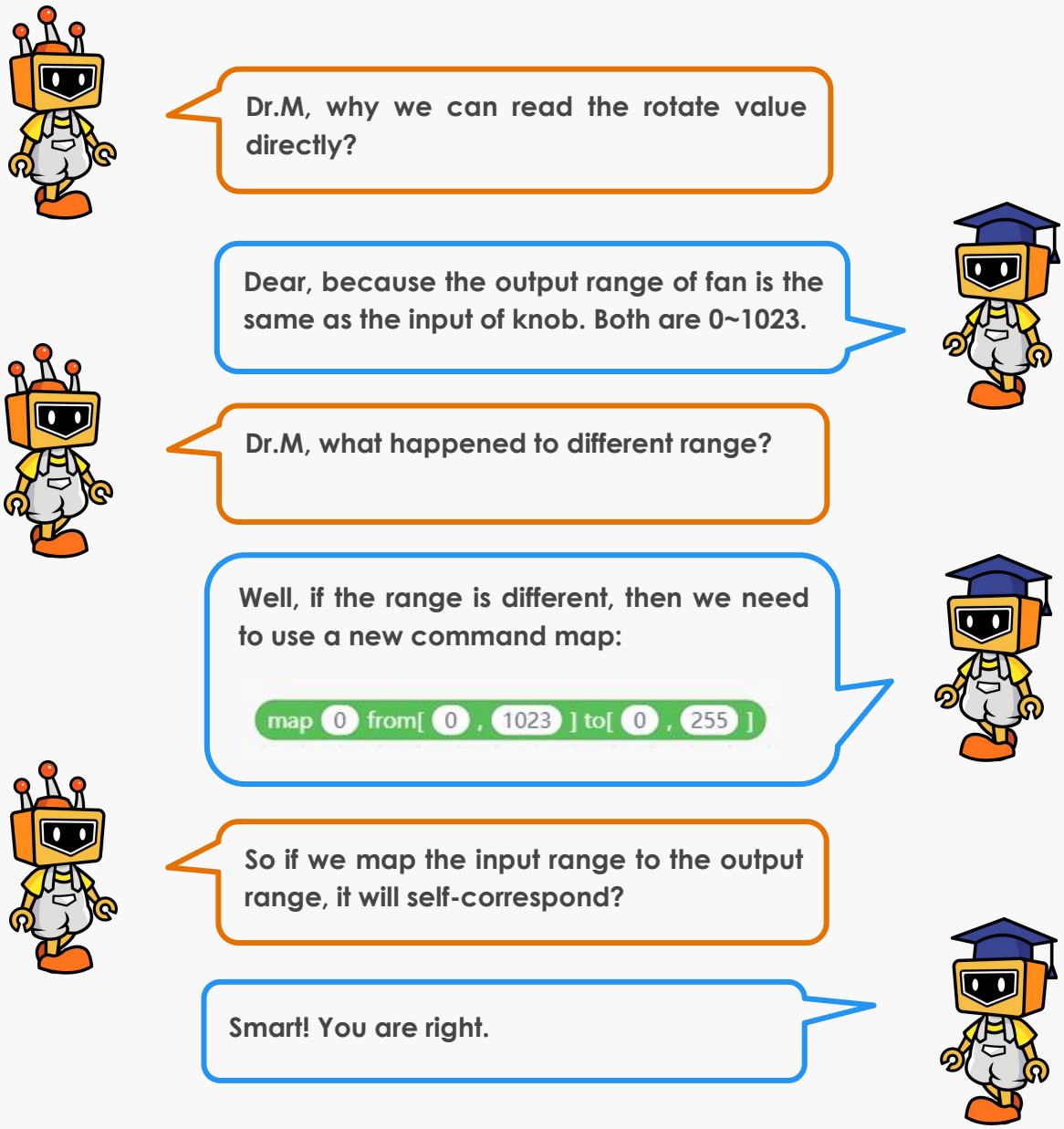


Figure 2.4.8 Read analog pin P0

2. Drag the "set analog pin P0 output" of the micro: bit module to the "forever" in the script area, revise P0 to P1, and nest the two blocks as shown below.

Figure 2.4.10 Run the Program

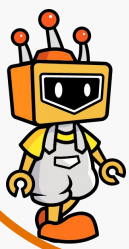


7.Design of Micko

The summer is coming, do you want to make a cool fan lamp? Then you can enjoy the cold wind more comfortable.

Tip: How to use 1 micro: bit to control LED and fan at the same time?

My Design	My Program

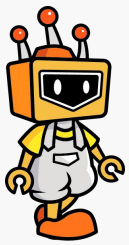


Chapter 3: Get hands on!

Except the push-button and switch knob module we learned before, there are more commonly-used sensors. Use them to make more smarter, funnier and useful projects. Come and learn!

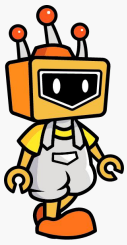
Project 1: Electronic Candle

1.Story of Micko and Dr.M



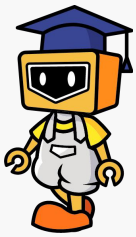
Dr.M, it's been a long time. We are back from the DF spaceship.

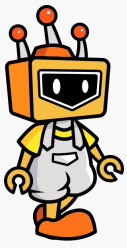
How's the traveling? Have you enjoyed yourself?



Yes, a nice trip. Here is a TV in the DF spaceship and I watched a video about greenhouse effect, and the main cause is greenhouse gas. So seriously. From the news I know any human activity may emits greenhouse gas, such as making a fire, driving car, air-conditioner.

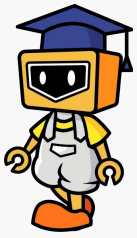
You are right, we should try our best to protect nature, and reduce the greenhouse gas emission.



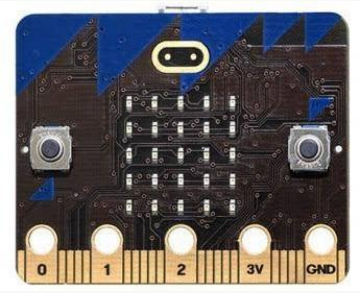


Dr.M, I will attend a birthday party of my best friend. What kind of gift should I give her?

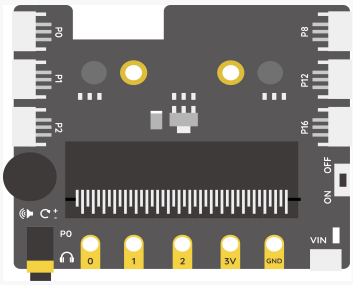
How about an electronic candle? An electronic candle will not pollute the environment.



2.Dr.M's Pocket



1 × micro: bit



1 × micro: bit Expansion Board

1 × LED Module

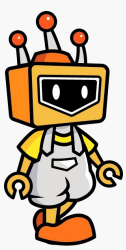
1 × Sound Sensor

1 × micro USB Cable

Let's get to know about sound sensor:

Sound sensor can sense the intensity of sound. After programming, it can be used to turn on/off the LED through the sound intensity.

3.Dream of Micko: Detect the sound strength



I want to know my voice power. That is to say, if the micro: bit shows 1 on board, means low voice; if it shows 2 means high voice.

How to design the project?

When the analog value of sound intensity is less than 80, we set the on-board LED display the figure "1", which means that the sound is low and intensity is grade 1.

Elsewise, when the analog value is over 80, the on-board LED display figure "2", which means the sound intensity is grade 2.

4.Dr.M's Blackboard

STEP1: Connect the sound sensor module to the pin P0 of the extension board; LED module to P1.

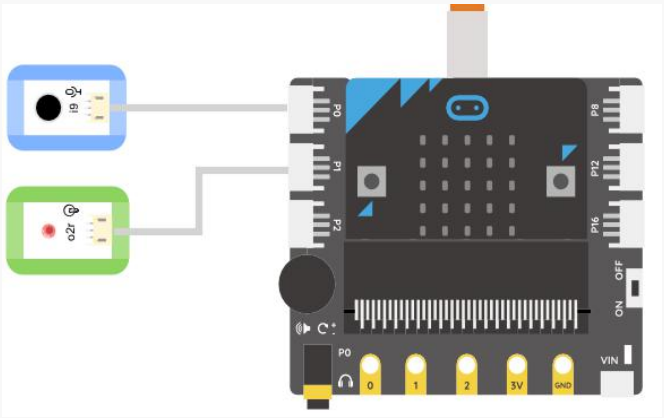


Figure 3.1.1 Connection Diagram

STEP2: Programming

1. Let's begin to design our project according to the tips: drag "read analog pin P0" and "display" blocks from the "Micro: bit" module; drag out "if… then…else" block from the "Control" module; drag "<" in the "Operation" module to script area.

Let' s review "if…then…else" statement:

The function of "if…then…else" statement:
If condition is true, the statements following "then" are executed. If condition is false, the statements following "else" are executed.

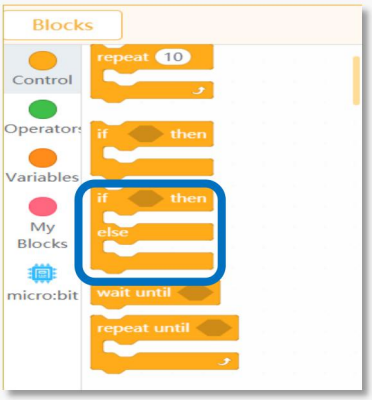


Figure 3.1.2 "if…then…else"

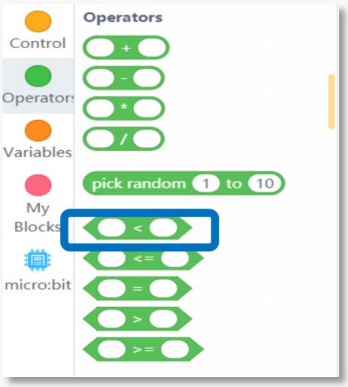


Figure 3.1.3 "<"

2. Read the analog value(0~1023) of "read analog pin P0" that connected with sound sensor to detect the sound of intensity.
3. Edit the programs to realize "sound intensity detecting". "<" can be used to judge if the analog value of sound intensity meets the set standard.

AS shown in figure 3.1.4, if the condition "'read analog pin P0'<'80'" is true, also means the sound intensity grade is 1, the on-board LED displays figure "1".

If the condition is false, the sound intensity grades 2 and the LED displays figure"2". Add a "wait 1 sec" block under the "...then" to prevent the LED figure from flashing.

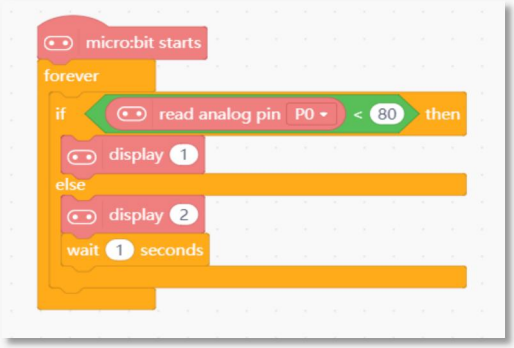
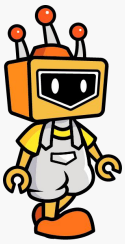


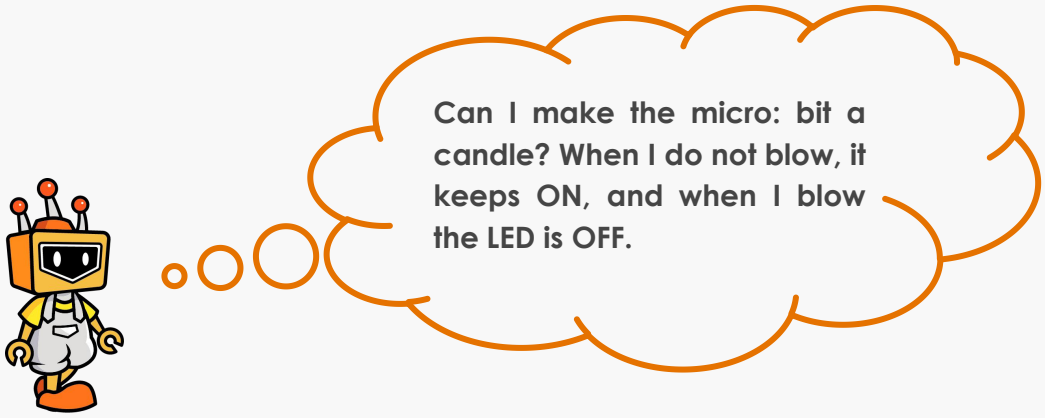
Figure3.1.4 Run the Program



Dr.M, this program really can test my voice power!

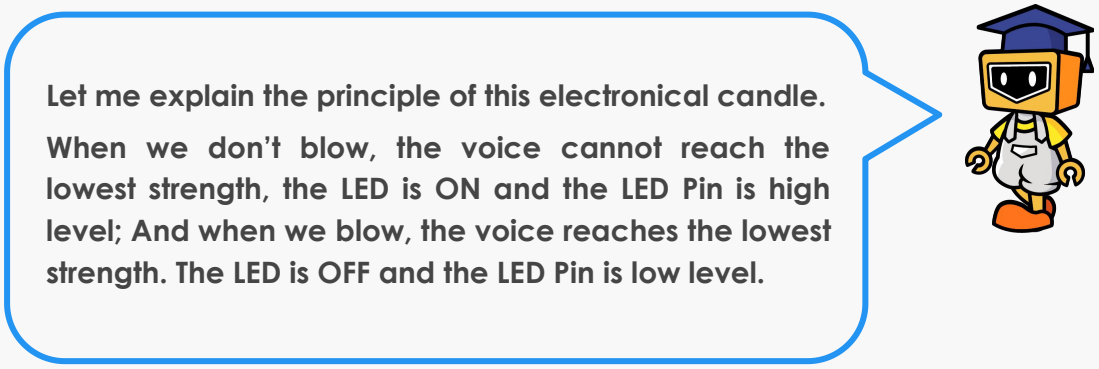
Well done! If it always shows 2, suggesting the environment is a little noisy. And the value 80 is too low. So, you should try setting a larger number.

5.Dream of Micko: micro: bit Candle



6.Dr.M's Blackboard

Control the LED to turn on/off according to the value of the pin connected to the sound sensor. “



1. Drag “set digital pin P0 output” block from “micro: bit” module to script area, and change P0 to P1.

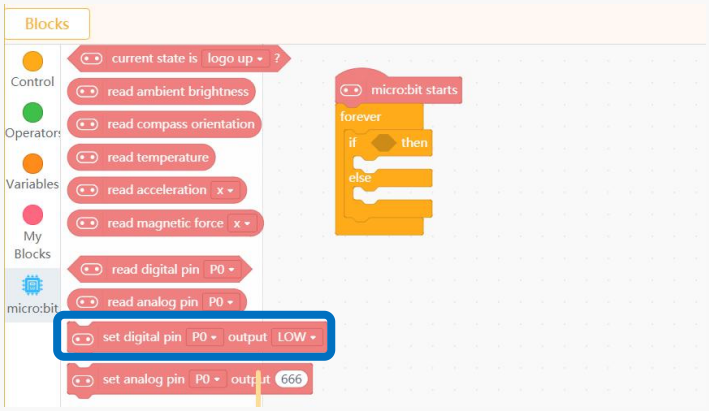


Figure 3.1.5 “set digital pin output”

2. Programming to make a "electronic candle"

When the analog value of sound sensor is less than the preset value 80, the LED keeps on, otherwise, blow it like a candle, when the sound intensity is over 80, the LED keeps off for 2 seconds.

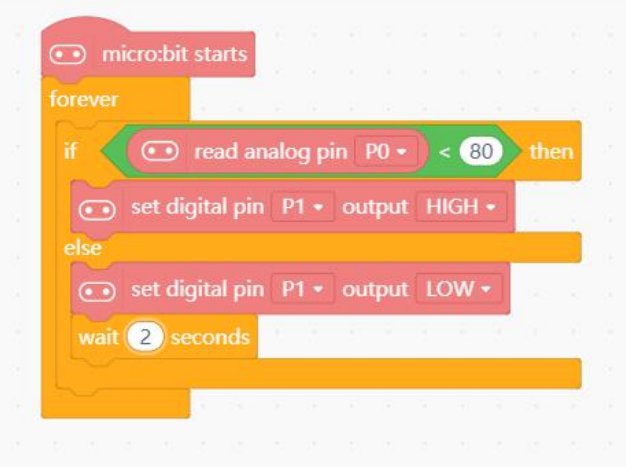
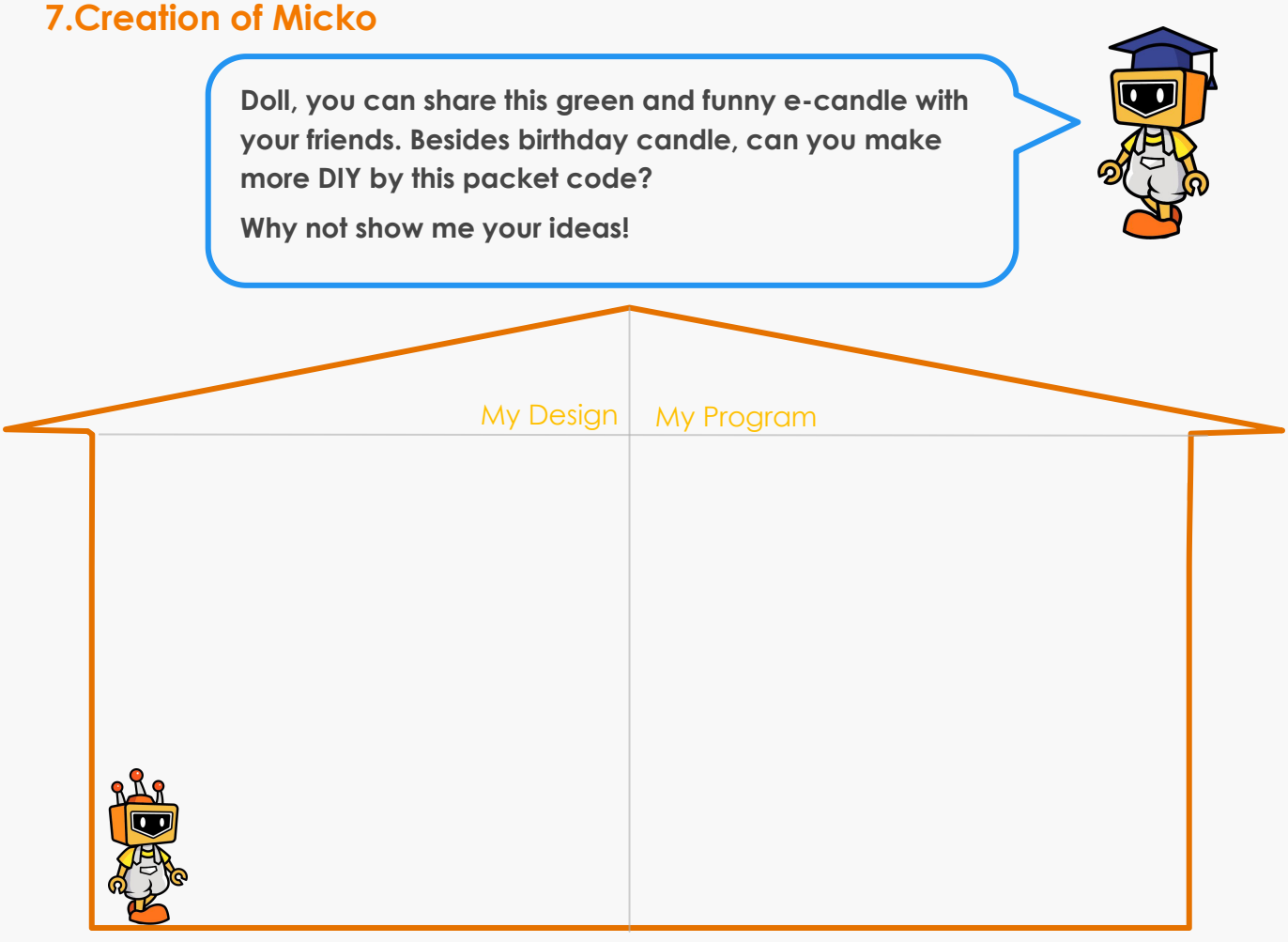


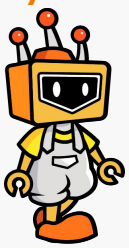
Figure 3.1.6 Run the Program

7.Creation of Micko



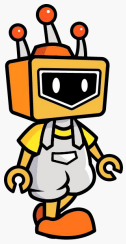
Project 2: Automatic Door

1.Story of Micko and Dr.M



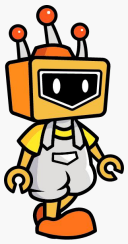
Dr.M, it is so funny to make that e-candle last week. There are so many interesting gismos in our life.

Yes, my dear, it's not the lack of joys in life, but the lack of the eyes to find joys. Have you find any joys?



Well, Dr.M, I went to library borrow robot-making book last week and found that I need not open and close the door. It seems like the door can feel my actions.

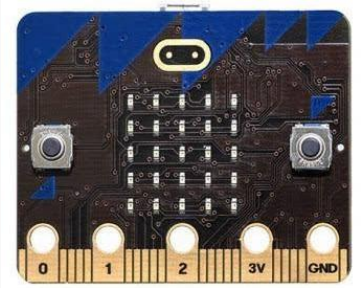
Sweetie, I am so pride of you! You are a little observer now. Guess why you need not to open and close the door?



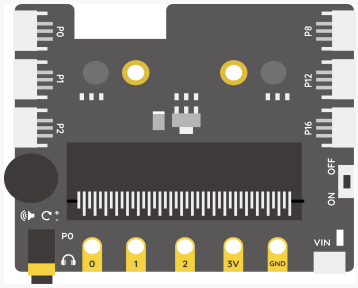
Em, maybe someone find we will go and open the door. But when I came in, I found no one.

Ha-ha, my dear, there is a magic device in the door. The device can control the door and make the door open and close.

2.Dr.M's Pocket

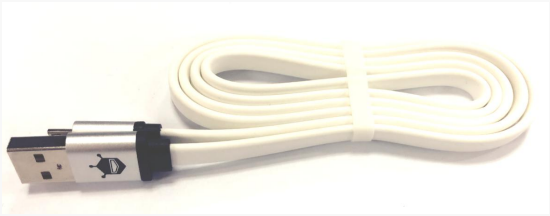


1 × micro: bit



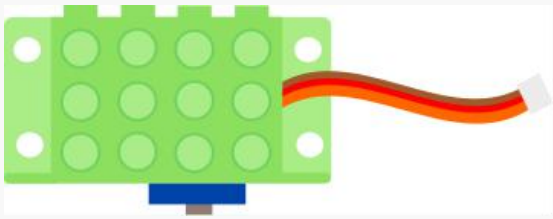
1 × micro: bit Expansion Board

1 × Push Button



1× micro USB Cable

1 × Motion Sensor



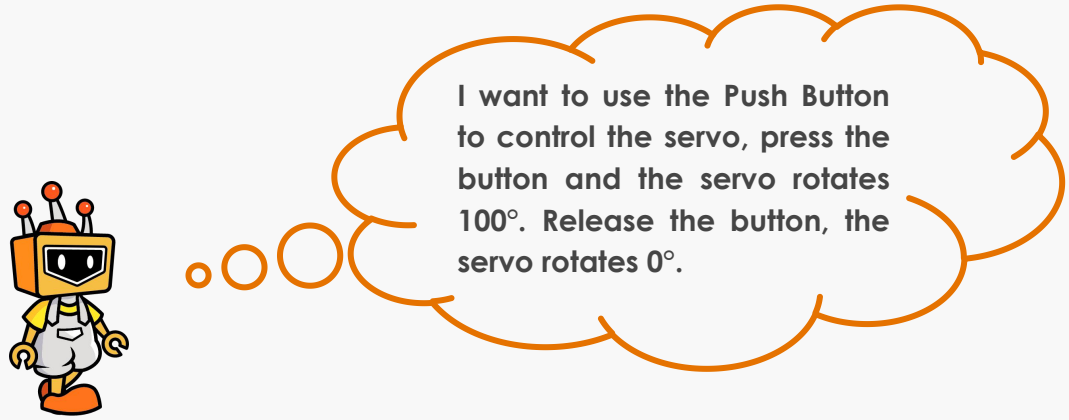
1 × Servo

First time to see Motion sensor and servo module, let's take a look:

Motion sensor, also called PIR sensor, can detect the infrared signals from human body or other animals and triggers with movement.

A servo come with a shaft that can be positioned to specific angular (0-180°/0-360°). Compared with motor, it has smaller rotation speed but larger torque. They are extremely useful in robotics.

3.Dream of Micko: Control Servo with Push Button



4.Dr.M's Blackboard

STEP1: Connect push-button module to pin P0 of the expansion board; connect servo to P1.

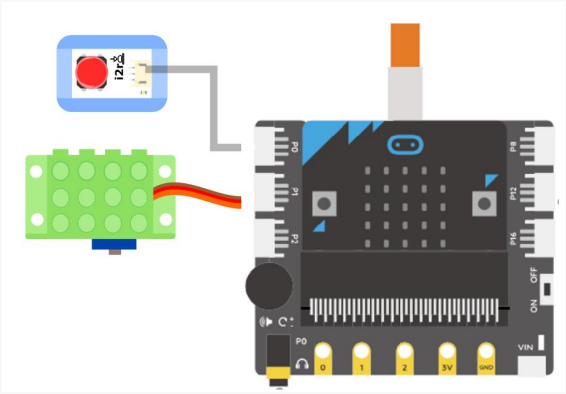


Figure 3.2.1 Connection Diagram

Add a command module before programming:

If we want to control the servo, we have to add the servo mand module. How to do that?

Click Extensions->Actuator->Micro Servo, and then back to the former interface.

Figure 3.2.2 Extensions

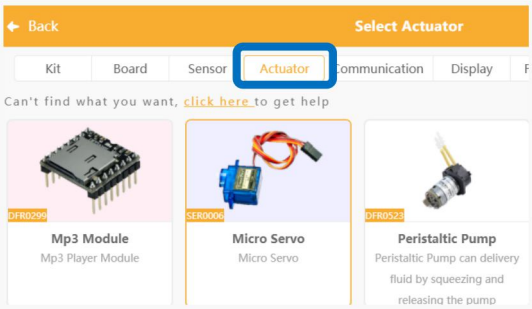


Figure 3.2.3 Actuator->Micro Servo

STEP2: Programming

1. Drag "set pin P9 servo to…degree" block in "actuator" module. Change P9 to P0. Set pin P0 to control servo' s value.

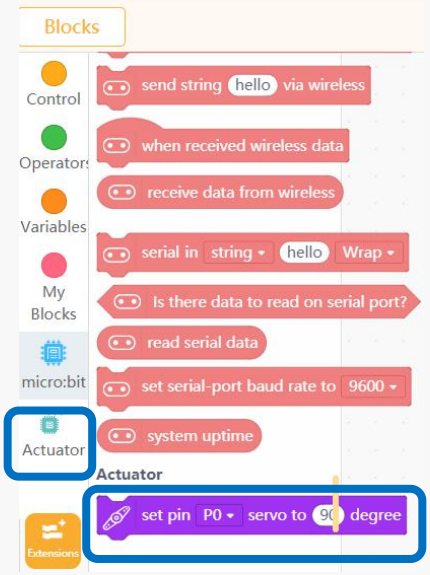
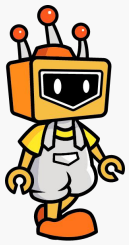
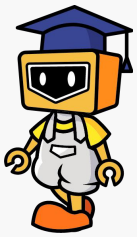


Figure 3.2.4 "set pin P0 servo to…degree"



Dr.M, is that control the servo value is to control the degree?

Yes, you can control the rotate degree by input the servo degree.



2. Set: if the button is pressed, the servo rotates for 100 degree and the LED display "o". Otherwise, the servo rotates for 0 degree and the LED display "×". Here we need to use "if…then…else" statement.

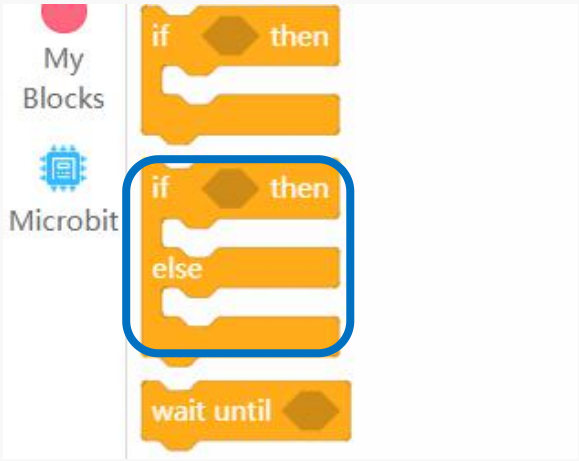


Figure 3.2.5 "if…then…else"

Note: an external power supply is necessary since the power consumption of the servo is large.

If you forgot how to connect the external power, review it at the "Control the fan with Push Button"

3. Run the program

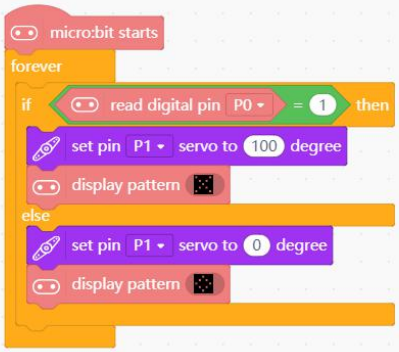
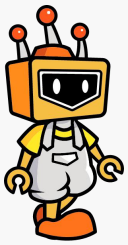


Figure 3.2.6 Run the Program



Dr.M, this is exactly what I want! But I forgot to connect the power supply and worried that the servo maybe broken.

That's Ok, we need to be patient and careful during testing the program.

Result

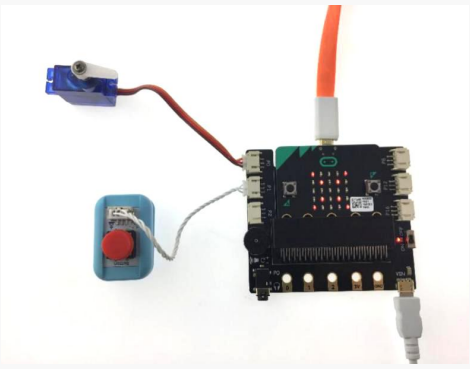


Figure 3.2.7 Result-Loosen

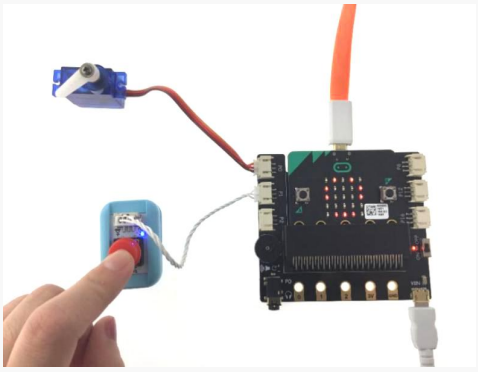
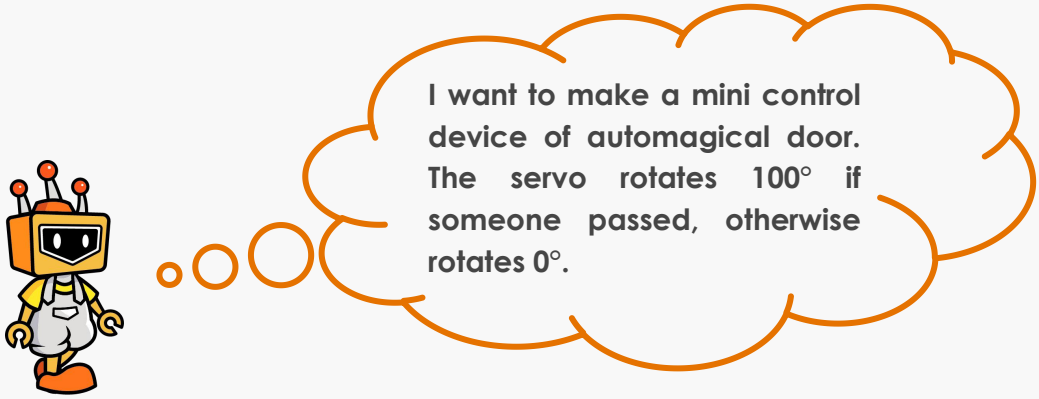


Figure 3.2.8 Result-pressed

5.Dream of Micko: Control the Servo with Motion Sensor



6.Dr.M's Blackboard

STEP1: Connect the servo to pin P1; connect the motion sensor to pin P0.

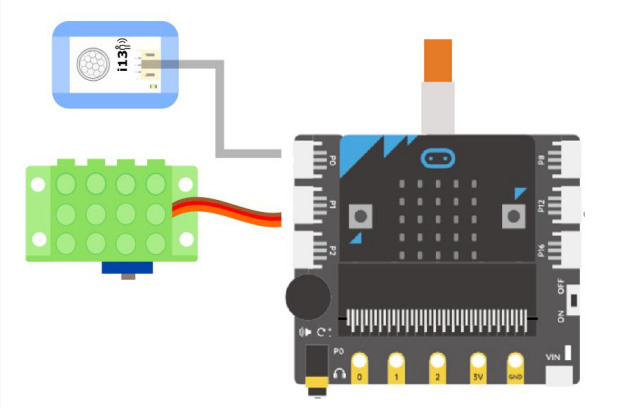


Figure 3.2.9 Connection Diagram

STEP2: Programming

1. Read the digital pin P0 to get the digital signal data of the motion sensor.

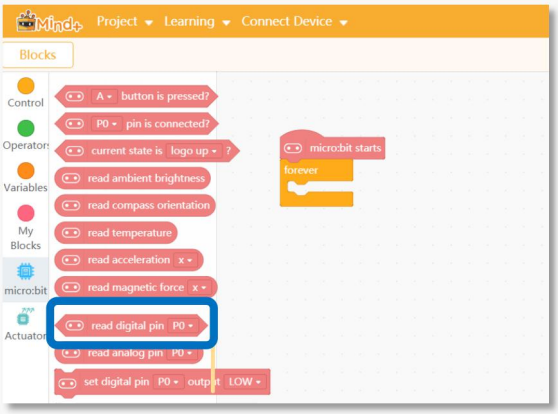


Figure 3.2.10 "read digital pin"

2. When the motion sensor detected that there is people passing by, the servo rotates for 100 degrees; otherwise, it rotates 0 degree. We need "if…then…else" statement here.

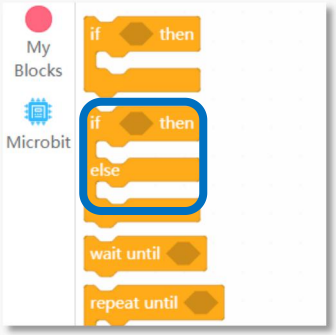


Figure 3.2.11 "if…then…else"

3. Run the program:

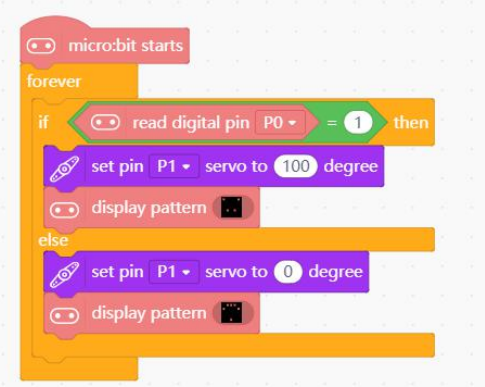


Figure 3.2.12 Run the Program

Result

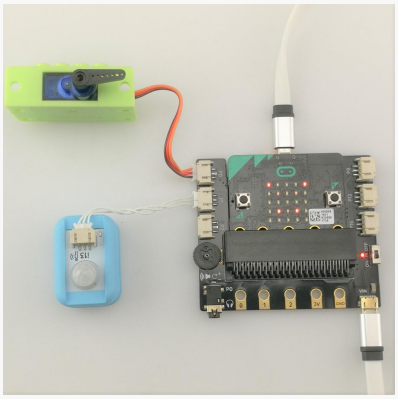


Figure 3.2.13 No one passed by

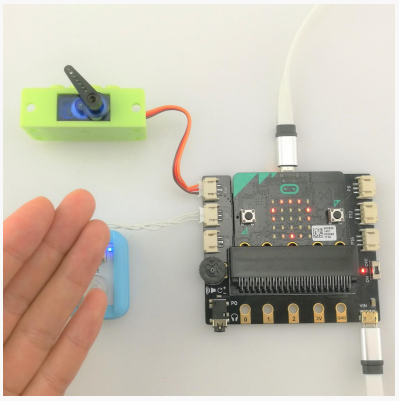
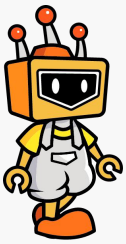


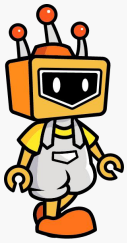
Figure 3.2.14 When people passed by

7.Creation of Micko

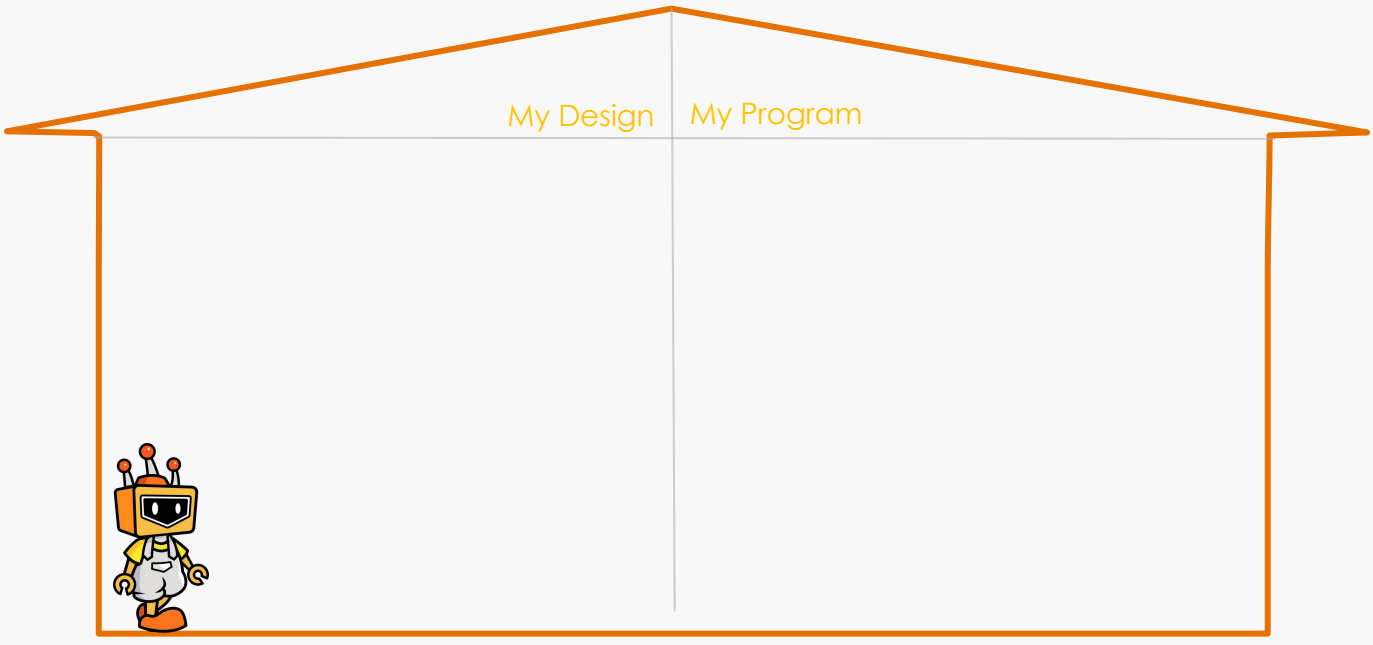


What a magic device! I know the way to control a door now.

Yes, my little friend, do you remember what I mentioned above? With micro: bit and mind+, you can make, enjoy and learn.

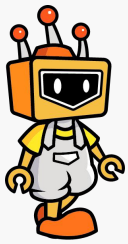


Dr.M, besides automagical door control, can we make other gizmos?



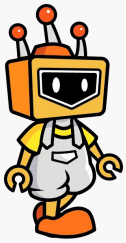
Project 3: Music Box

1.Story of Micko and Dr.M



Dr.M, except robot class, do you know which class is my favorite?

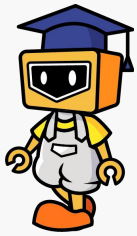
Let me guess, probably music? The beautiful and lovely sounds are always attractive.



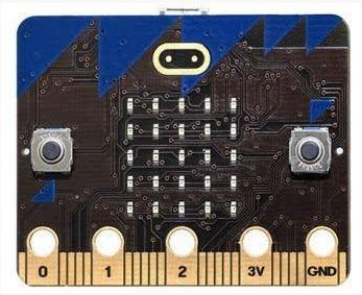
Bingo! Music is one of my favorite classes.

Last week, our music teacher showed us a beautiful box and that box can play music. My goodness, I want to have one.

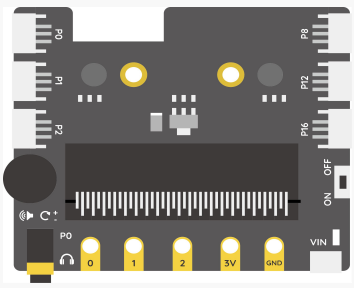
Why not take the micro: bit and make one?



2.Dr.M's Pocket

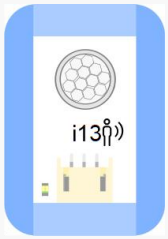


1 × micro: bit



1 × micro: bit Expansion Board

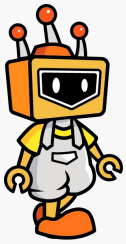
1 × Earphone

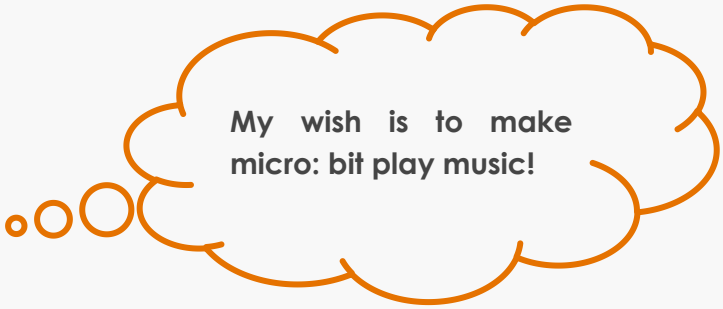


1 × Motion Sensor

1 × micro USB Cable

3.Dream of Micko: Play Music





4.Dr.M's Blackboard

STEP1: connect micro: bit board to computer with USB cable, plug the earphone in the expansion board; connect the motion sensor to pin P1 of the expansion board.

Note: pin P0 has been occupied by the earphone jack data communication, so it cannot be used when playing sounds.

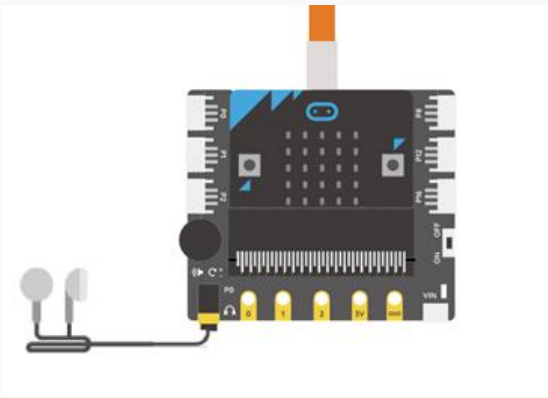


Figure 3.3.1 Connection Diagram

STEP2: Programming

1. Drag **"pin P0 play sound…until done"** block from "micro: bit" module to script area. There are many built-in sounds to choose. We select the sound **"dadadadum"** here.

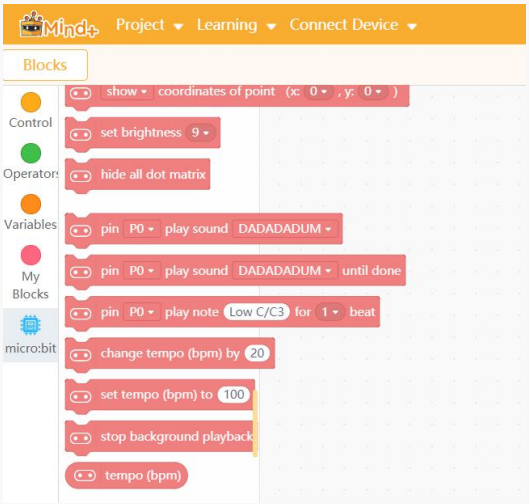


Figure 3.3.2 Command "pin P0 play sound…until done"

2. Run the program

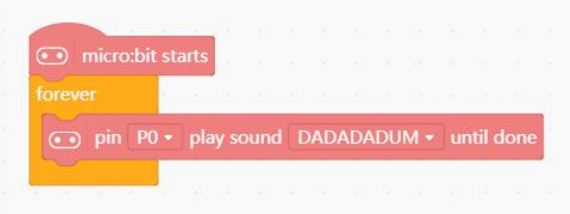
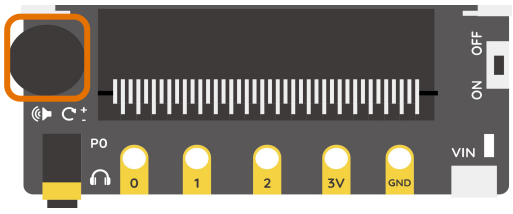
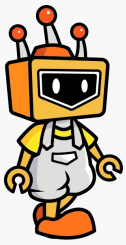


Figure 3.3.3 Run the program

Note: switch the "small turntable" near the earphone jack to turn up/down the sound.





How amazing! This small board can even play music!

Ha-ha, this is just a tip of the iceberg.
You will find more interesting functions.

5.Dream of Micko: Smart Music Box

I will play my own music in micro: bit. How cool if it can play when I am moving close and stop playing when I am leaving.

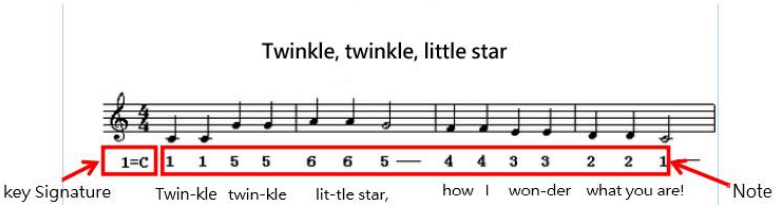
6.Dr.M's Blackboard

1. Program to generate a piece of music of *Twinkle, Twinkle, Little Star.*

Dr.M, I love music but I am not good at it. I know nothing abut musical note. What should I do?

My sweetie, do not be so nervous. Come look at the Little Tips!

Let's take a look about the numeric notation tablature of this song. The different key and note decided the different frequency of the sound.



How to use Mind+ to present the numeric notation tablature and play the sound

Note	1	2	3	4	5	6	7
	(do)	(re)	(mi)	(fa)	(sol)	(la)	(si)
Letter	C	D	E	F	G	A	B

2. Drag **"pin P0 play note-C2 for 1beat"** from the "micro: bit" module to script area. Revise the parameter **"C2"** to play different notes, and change the parameter **"1"** to adjust the beat.

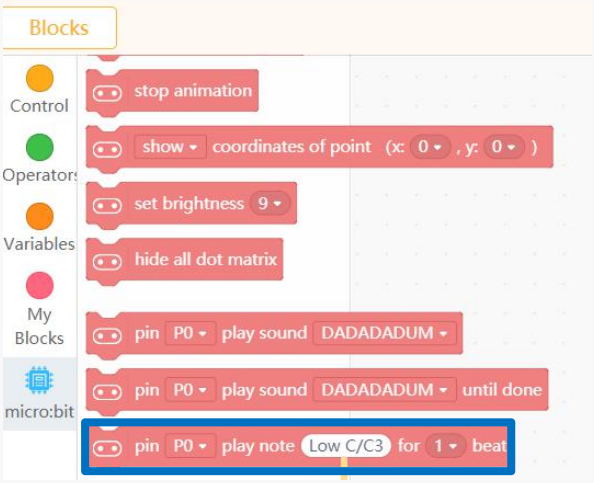


Figure 3.3.5 "pin P0 play note"

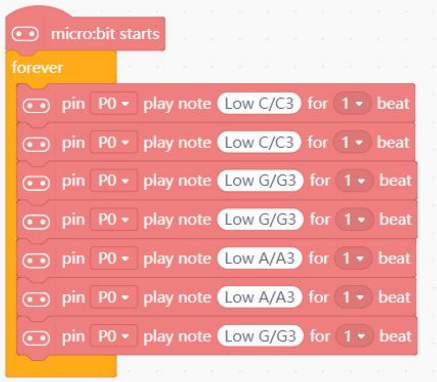
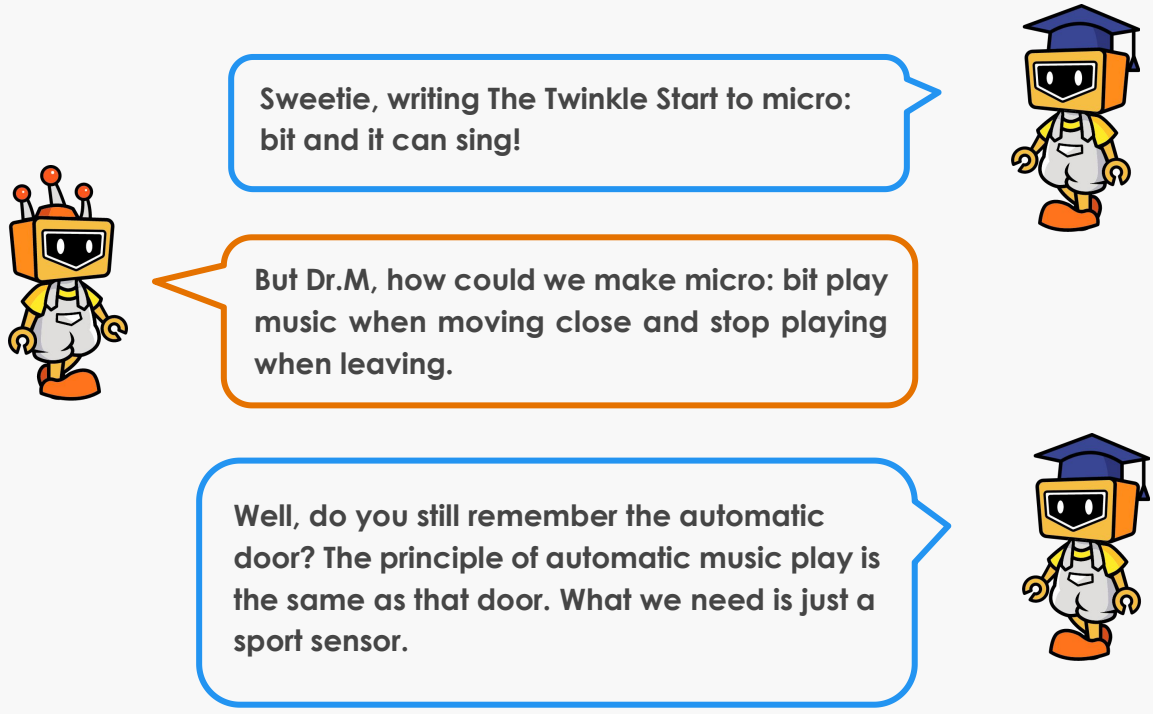


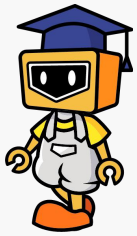
Figure 3.3.6 Write the Song



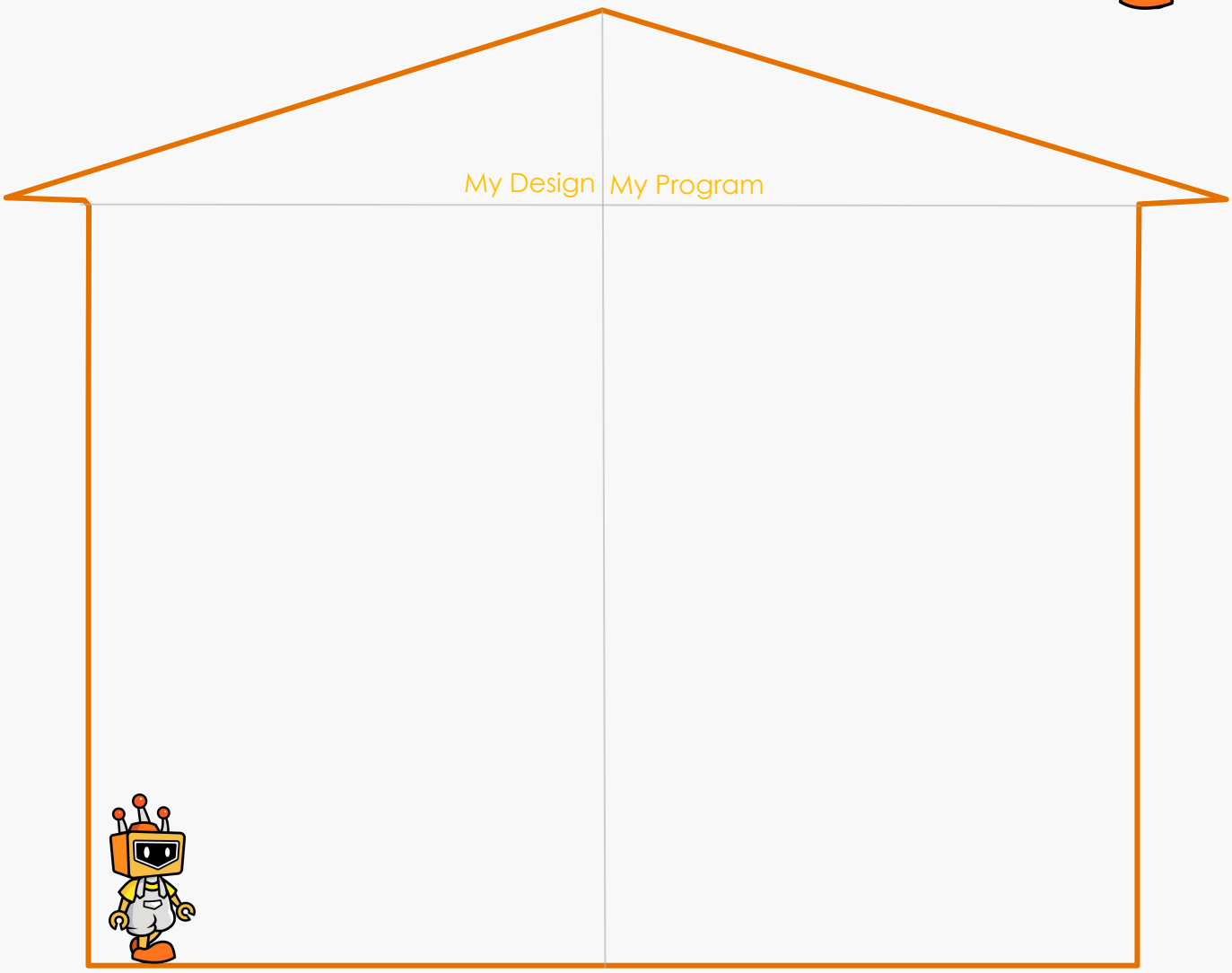
3. Motion sensor has been connected to pin P1, if someone passes near the device, the value of P1 will be 1, otherwise, it will be 0. Therefore, combining it a **"if then…else"** statement, we can make the smart music box come to life.

Figure 3.3.7 Run the Program

7.Creation of Micko

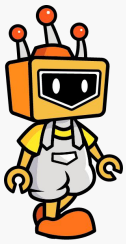


My friend, do you what to make your own music box? Get thinking and making one!



Project 4: Colorful LED Strip

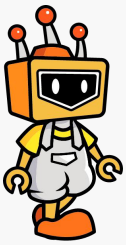
1.Story of Micko and Dr.M



Dr.M, the last music box is so cool!

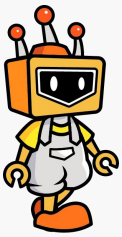
But…

Yes? Any questions?



Dr.M, I want to make a good-looking shell. How fantastic if the shell can shining like stars and dancing with the music.

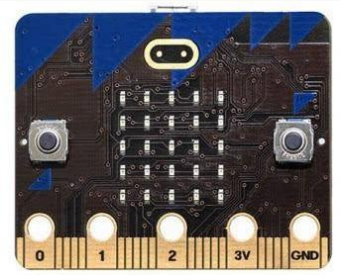
I see, sounds not bad. You know? micro: bit can achieve it!



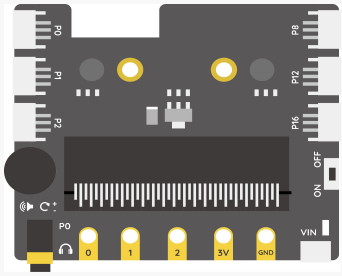
Wow! Really?! micro: bit is so powerful! Dr.M, please help me make it!

My pleasure.

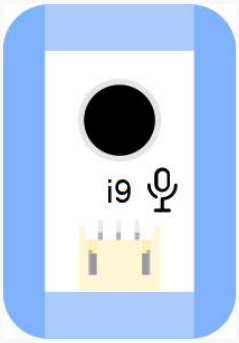
2.Dr.M's Pocket



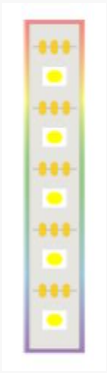
1 × micro: bit



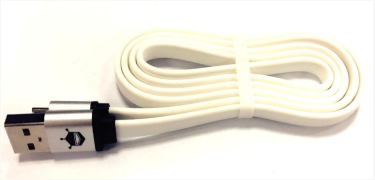
1 × micro: bit Expansion Board



1 × Sound Sensor



1 × Colorful LED Strip

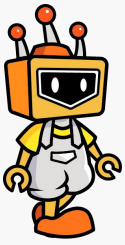


1 × micro USB Cable

First time to see LED strip, come to meet her:

We can set different parameter to each LED of the LED strip to achieve all kinds of cool effects.

3.Dream of Micko: Turn on the RGB LED strip



This time, I want to light ON all LEDs and makes them shine like a rainbow!

4.Dr.M's Blackboard

STEP1: Connect the LED strip to pin P0 of the expansion board.

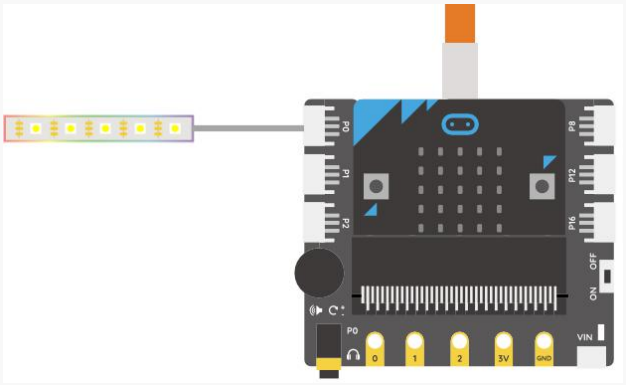


Figure 3.4.1 Connection Diagram

STEP2: Add the LED strip command module. Click **Expansions**->**Display**->**WS2812 RGB LED Strip**, then back to the former interface.

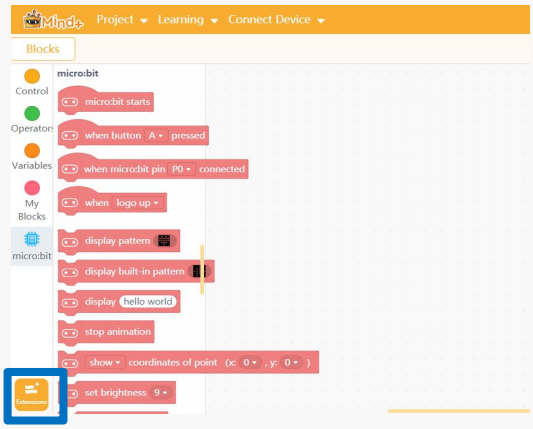


Figure 3.4.2 "Expansions"

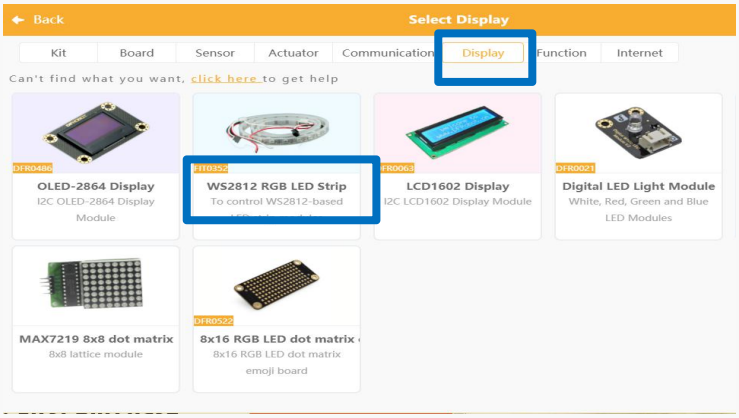


Figure 3.4.3 Add the display "WS2812 RGB LED Strip"

STEP3: Programming

1. Click **"display"** module in the left to present the LED strip commands.

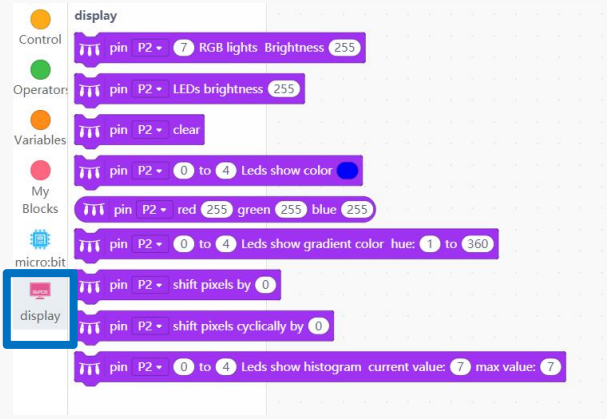
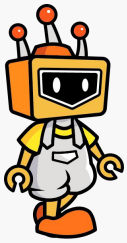
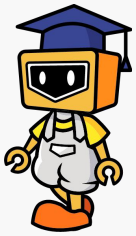


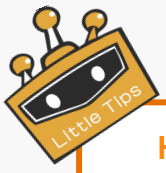
Figure 3.4.4 "WS2812 RGB LED Strip" blocks



There are so many blocks at here. I feel a little dizzy.

Take a breath! Come look at the Little Tip and check the commands one by one.

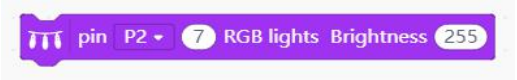




How to use the commonly used LED strip commands?

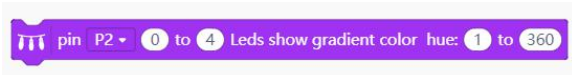
To start using the LED strip, we need to select the pin that the LED strip connected to micro: bit, and set up the LED numbers and color modes.

1. **If you want to set how many LEDs will be used and what brightness they will display, the block is shown below:**

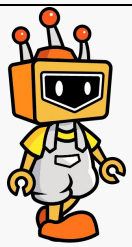


(the highest brightness is 255; the LED number is the total number of LEDs that physically included in the strip)

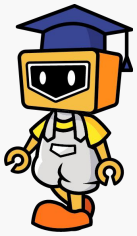
2. **If you want to set light effect, you need to use the following block:**



(LED 0 represents the first LED, and there are 360 colors.)



Dr.M, why 0 stands for 1 LED? Why we do not use 1 to represent 1 LED?



Big question! This is because the Arabic numerals are beginning with 0. So we always use 0 as the first one, so the 1 LED is represented by 0.

2. The pin P0 connected to the LED strip comes with 7 LEDs. Set the color range to 01-360 to realize a rainbow effect display (red-green-blue).

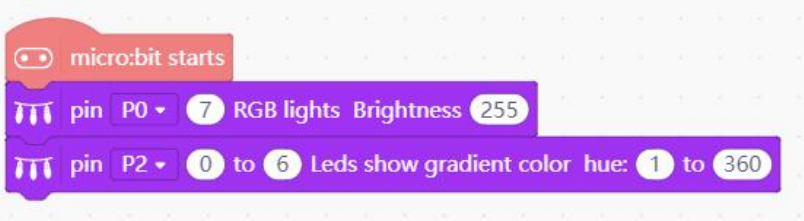
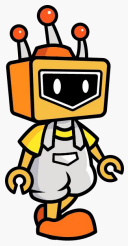


Figure 3.4.5 Run the Program



I guess the LED belt has 7 LEDs. Counts the LEDs from 0 to 6: 0,1,2,3,4,5,6. There are 7 numbers at all. So the LED number is from 0 to 6.

Result

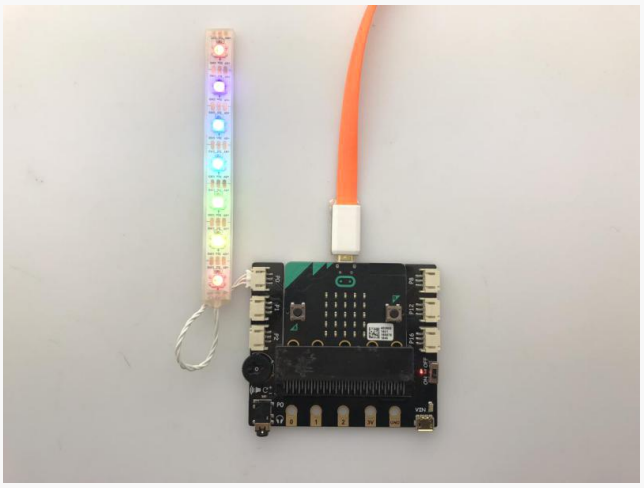
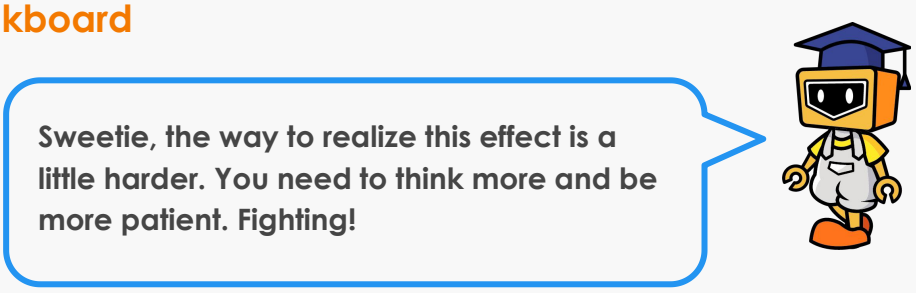


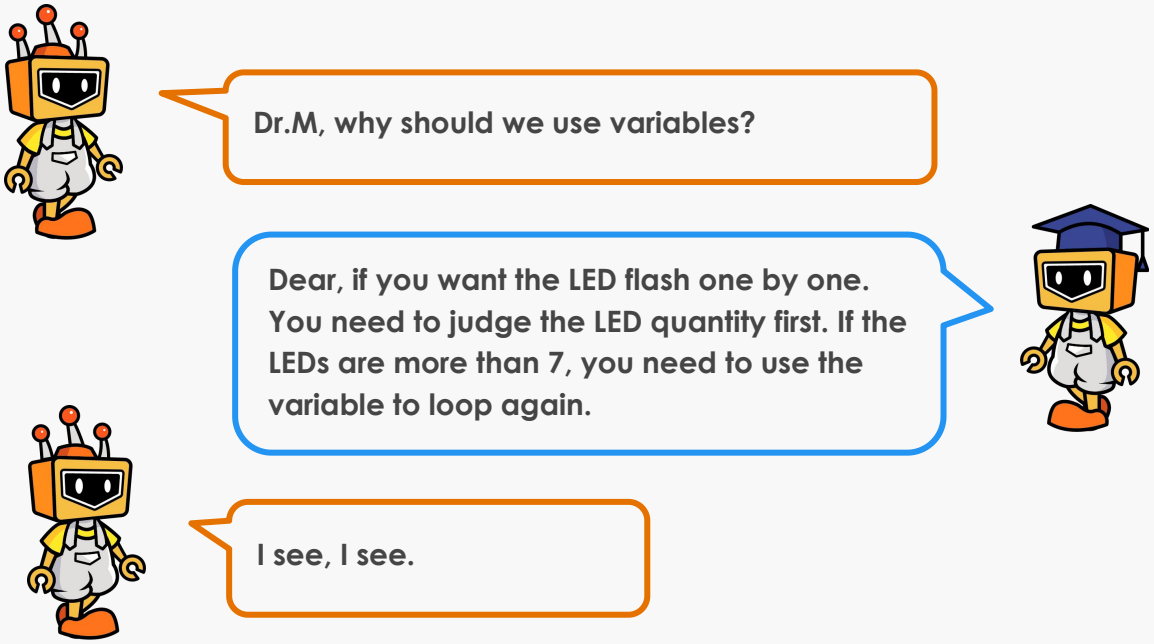
Figure 3.4.6 Result

5.Dream of Micko: Turn on the rainbow LEDs

6.Dr.M's Blackboard



1. If we want to make the LEDs turn on in a sequence, we need to use a new module: **Variables**.



To set such a variable, go to: **"Variables-> Make a Numeric Variable"**, and set the name of the variable (to be clear, we named the variable as "the number of LEDs to be lit"). Then click OK.

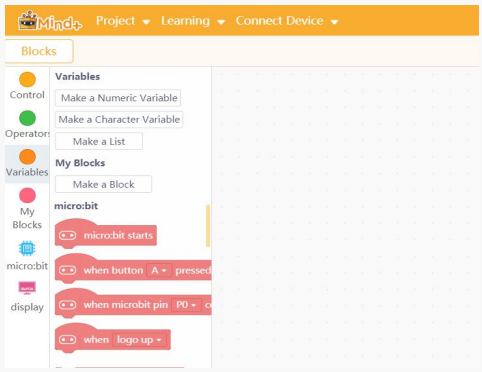


Figure 3.4.7 Create a new Variable

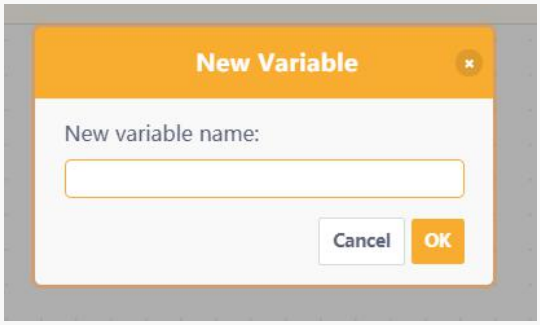
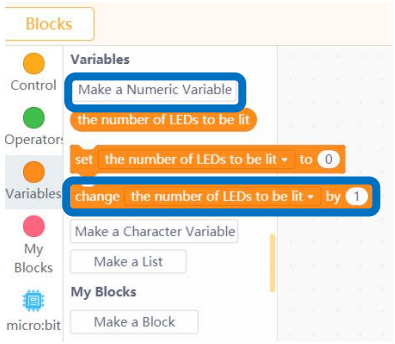


Figure 3.4.8 Name the Variable

First time to create a variable, let's get to know:

After we created the variable, the blocks of "the name of the variable", "set…to" and "change…by" will appear in the command section:



Just drag the related block to the script when you need it!

2. Initialize the pin and total number of the LED, and set the brightness. The initial value of the variable **"the number of LEDs to be lit" (α for short)** should be set to 0(since all the LEDs are turned off at first).

Figure 3.4.9 Initialize the pin and total number of LEDs

3. When $\alpha < 6$, α LEDs to be lit, then maintain 1 second, and make the parameter getting larger by 1 in every loop. If $\alpha > 6$, make all the LEDs

turn off and start from the beginning. The flow chart below may help us to figure out what the program looks like:

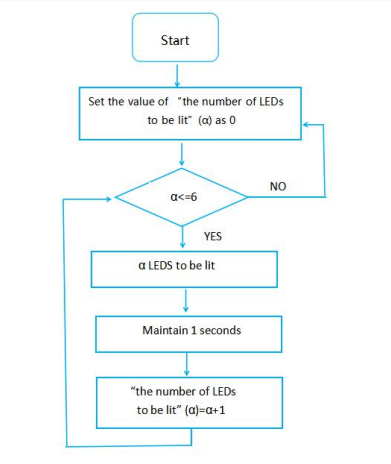
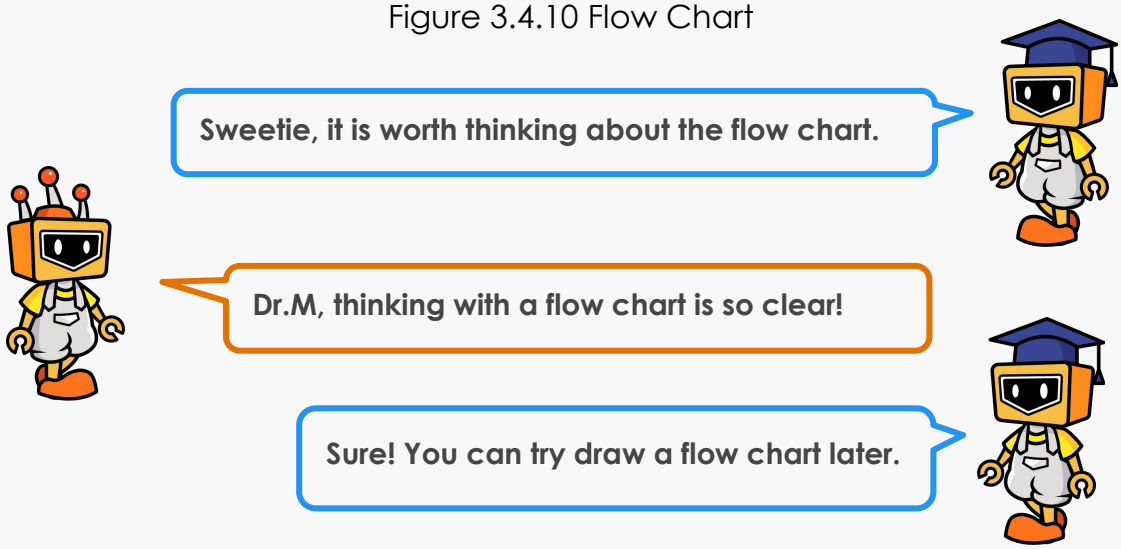


Figure 3.4.10 Flow Chart



4. Programming according to the flow chart, first setting the conditions, we need to use "if…then…else" and "forever" blocks.

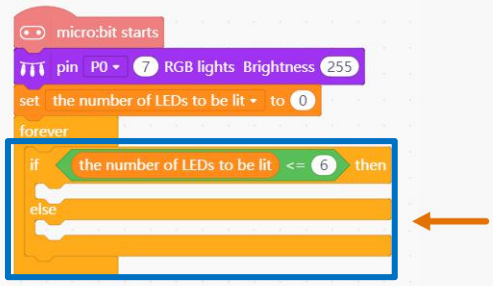


Figure 3.4.11 "if…then…else" and "forever"

5. If **"the number of LEDs to be lit"(α)** is less than 6, light up **α** LEDs, wait 1 second, then make the parameter getting larger by 1, which means to make one more LED light up.

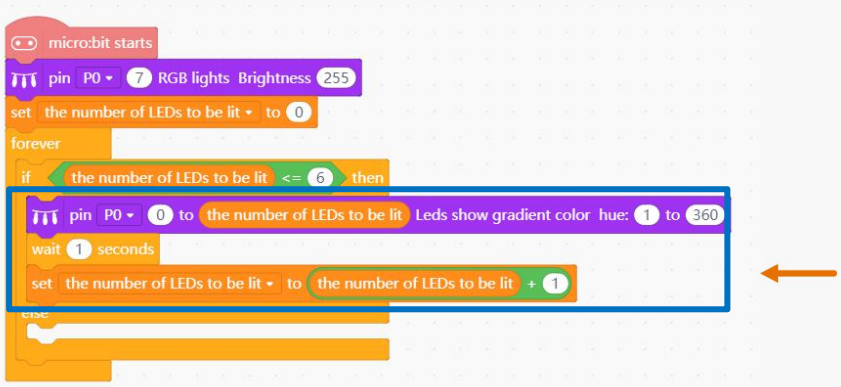


Figure 3.4.12 Commands about α

6. When the number of LEDs to be lit is over 6, **set the value** of **"the number of LEDs to be lit"** to 0, and turn off all the LEDs.

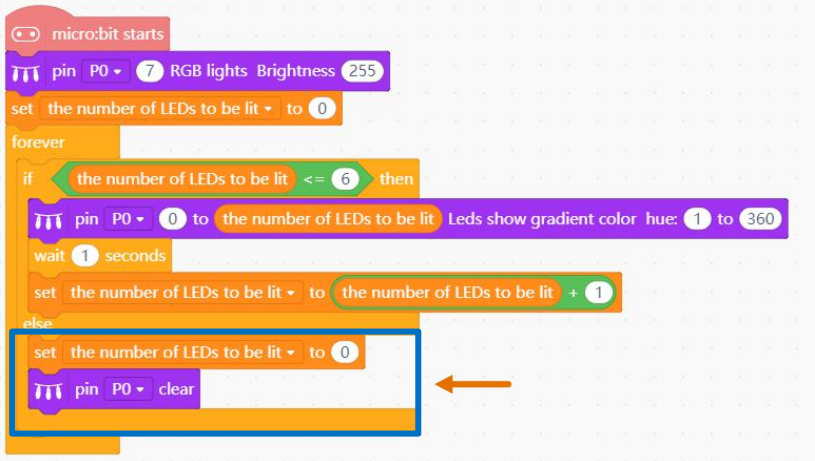
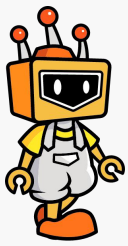


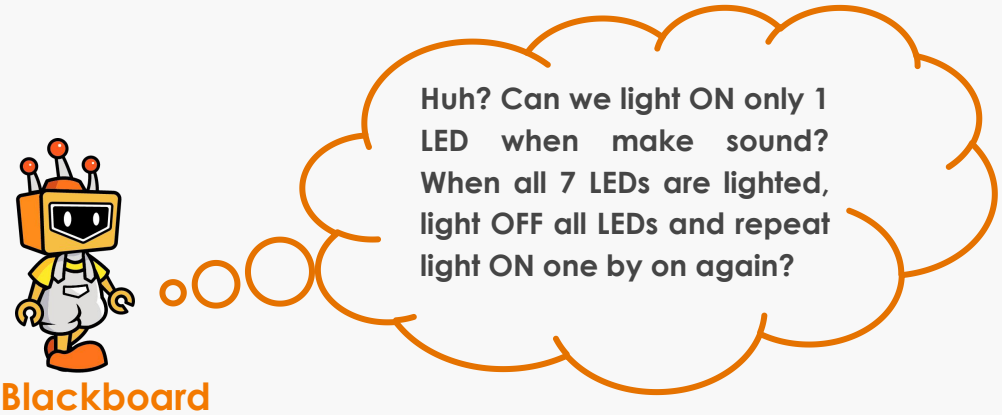
Figure 3.4.13 Run the Program



Dr.M, the program looks a little complex.

Yes? Although the programs become more and more complex, the works will be more interesting. Practice makes perfect.
Get thinking now!

7.Dream of Micko: RGB LED strip controlled by a sound sensor



8. Dr.M's Blackboard

STEP1: The LED strip still connected to P0; connect the sound sensor to pin P1.

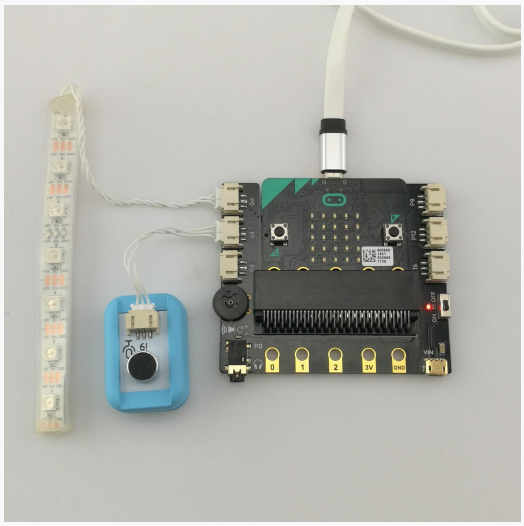
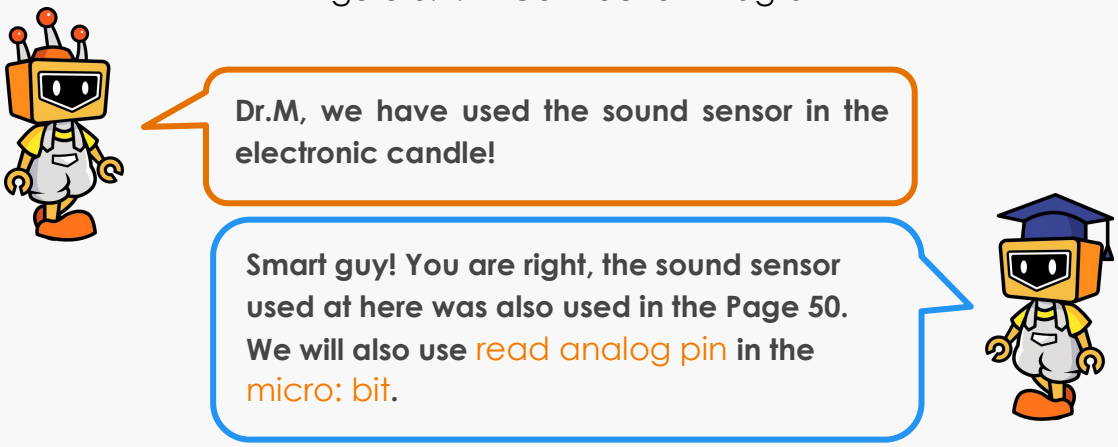


Figure 3.4.14 Connection Diagram



STEP2: Click "micro: bit" module, drag the "read analog pin P0" block to script area, and revise P0 to P1.

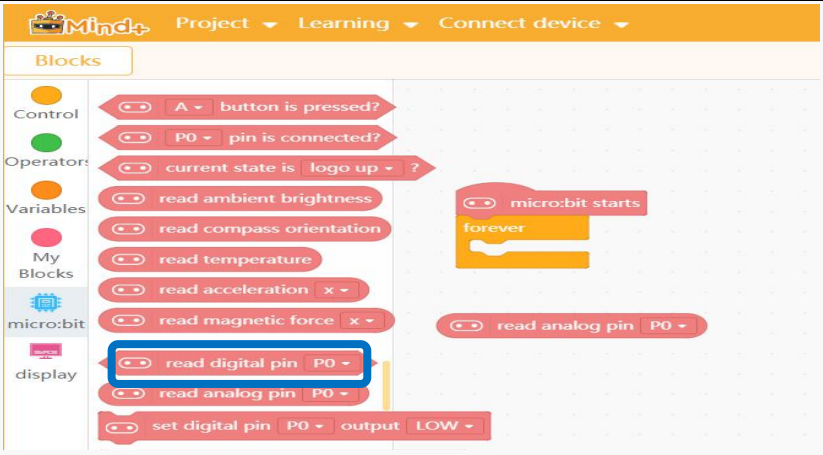


Figure 3.4.15 "read analog pin P0"

STEP3: If the sound value the sensor detected exceeds 40 and the **"the number of LEDs to be lit"** α is less than 6, control the LED strip to light up α LEDs. Then the sensor will be constantly detecting the sound value.

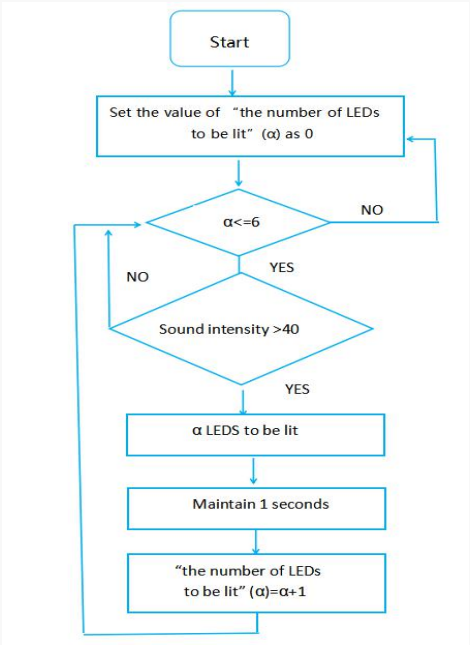
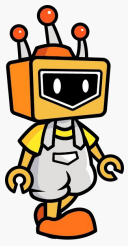


Figure 3.4.16 Flow Chart



Dr.M, we have used the sound sensor in the electronic candle!

Smart guy! You are right, the sound sensor used at here is the same as what used in the Page 47. We will also use read analog pin in the micro: bit.

STEP4: Programming

Based on the program of the last project, we need to add a "if…then" statement to judge the sound value. That is to say, when **"the number of LEDs to be lit"** α ⩽6, **if** the analog pin P1 is over 40, **then** α LEDs will be turned on, wait 1 second, change α by 1, and repeatedly to detect if the sound value exceeds 40. When **"the number of LEDs to be lit"** α >6, all the LEDs will be turned off and start the program from the beginning.

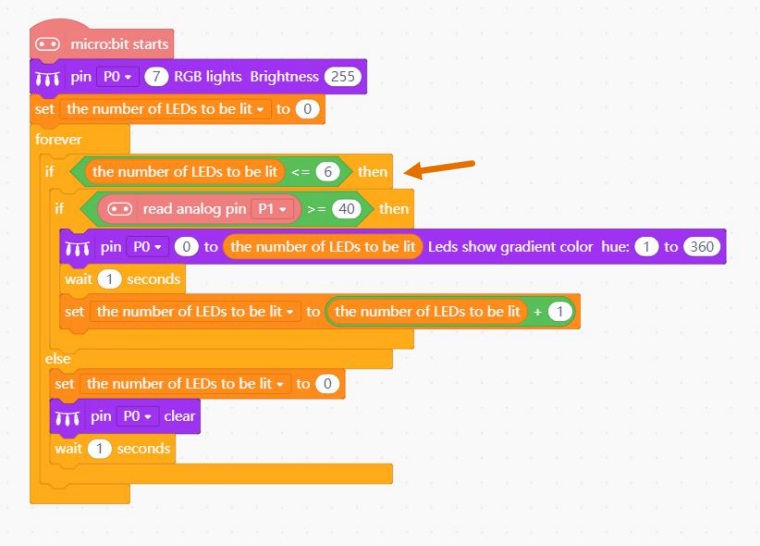
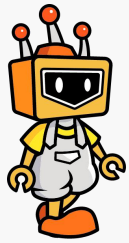


Figure 3.4.17 Run the Programs

9.Creation of Micko



Wow! Such a difficult program! Seems like I need to practice now. How about make a funny sound control light? Come to join me!

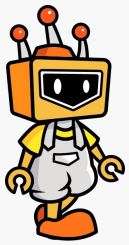
My Design | My Program

Chapter 4: Become an expert

We have built so many funny works with different types of modules in previous chapters. You may now have a deeper understanding of the micro: bit. Is that enough for the exploration? The answer is doubtlessly NO! Hang on there and we will take another further step into the micro: bit.

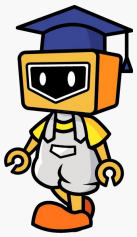
Project 1: Electronic Stabilizer

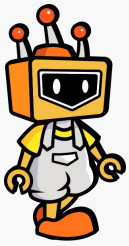
1.Story of Micko and Dr.M



Dr.M, I found an amazing phenomenon when playing outside.

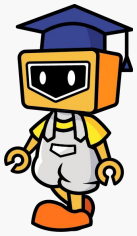
What's that?

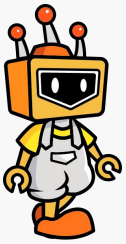




Dr.M, my telephone was lying in my hand. But when my hand turns to left, the telephone page magically turns to right! Could you please tell me why?

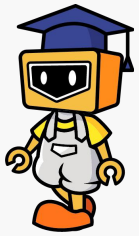
You are good at observing. That's because the self-balancing device in the telephone. We call this device stabilizer. The display pages should keep upright so that we can conveniently check it. So the telephone is set to know direction. And if you turn the telephone to left, the display page will turn to



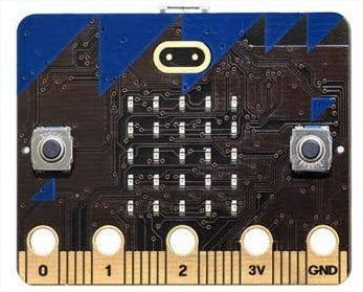


Amazing! Dr.M, can we make a e-stabilizer like that?

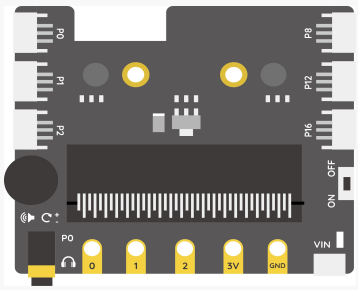
Of course! Let's explore the magic electronic stabilizer with micro: bit!



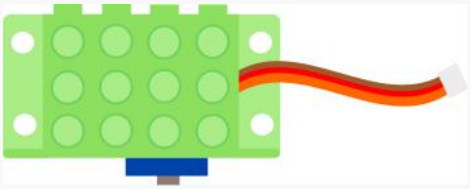
2.Dr.M's Pocket



1 × micro: bit



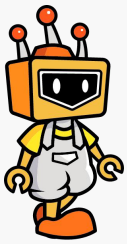
1 × micro: bit Expansion Board



1 × Servo

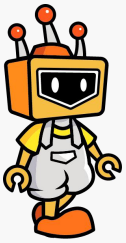
1 × micro USB Cable

3.Dream of Micko



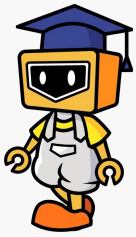
How could I make a servo pointer keeps vertical no matter how to rotate?

My sweetie, have you heard the acceleration gravity?



Yes, Dr.M. I have heard the acceleration gravity in Physics. But is there any connection between stabilizer?

If you understand the acceleration gravity, you can understand the principle of self-balancing device more easily. Let's come to the Little Tips to check it!



What is gravity?

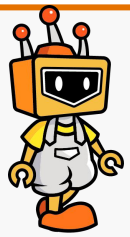
It is the force that attracts an object toward the center of the earth, or toward any other physical object having mass.

What is acceleration of gravity?

The magnitude of the acceleration due to gravity, denoted with a lower-case g, is 9.8 $m/s^{2.}$

What can we make using acceleration of gravity?

Micro: bit comes with an accelerometer, which can detect the gesture of the micro: bit board automatically. Through the detection of accelerator and programming design, we can realize to make an on-board servo keep pointing upright regardless how we tile the board.



Dr.M, will we connect the servo to board and control the servo direction by the acceleration gravity sensor in the board?

Yes, you have understood it. But the acceleration gravity has different directions. Let's turn to the Little Tips to learn which direction we will use.

Micro: bit Accelerometer:

Micro: bit can detect the gravitational force along X, Y and Z directions, whereas X is along the left-right direction, Y is along the front-back direction, Z is perpendicular to the board and along the up-down direction. Show as below:

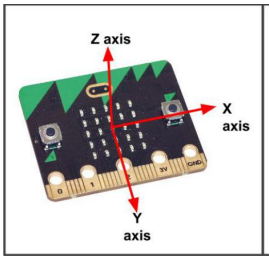


Figure 4.1.1 X, Y, and Z axes

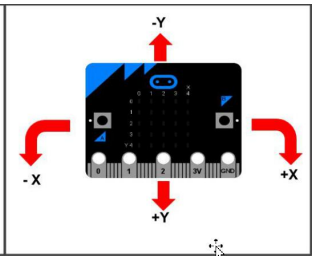


Figure 4.1.2 Value Changing

When the micro: bit board lays flat on the desktop, gravitational force goes along up-and-down direction (Z axis), so that the values of X and Y are close to zero.

When we tilt the board, the value of X axis changes. When it leans to the left, its value is negative, conversely, positive. Shown as figure 4.1.2.

Dr.M, in my wishes, we need to calculate the acceleration gravity in the direction x?

Bingo! Good answer.

4.Dr.M's Blackboard

STEP1: Connect the servo to the pin P0 of micro: bit expansion board.

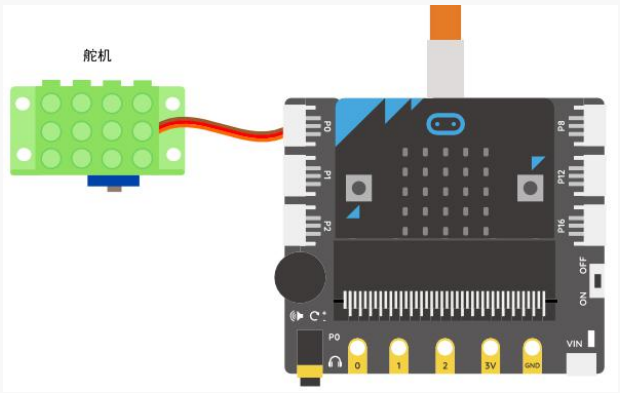


Figure 4.1.3 Connection Diagram

STEP2: Programming

1. Drag the "read acceleration" block in micro: bit module to script area. It can be used to get the acceleration of X, Y and Z axes.

Select X, Y and Z axis in the drop-down box.

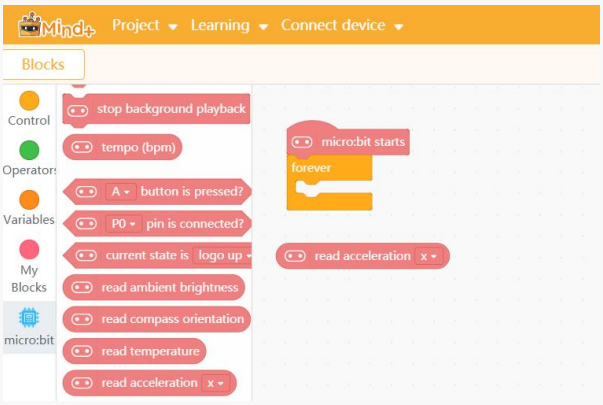


Figure 4.1.4 "read acceleration"

2. The output value of the accelerometer ranges between -1023 and 1023, and we can use "display" block to check it.

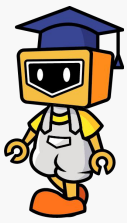
Figure 4.1.5 "display" the acceleration value

Figure 4.1.6 Result

3. When micro: bit board is titled to the left, a negative output value will be produced along the X direction, so we need the servo to rotate toward right to compensate this angle.

Dr.M, is there any connection between the acceleration gravity and servo rotate angle?

Little friend, such a good question. Continue reading the chapters below.



Here we have to use "map" function. Mapping the value of accelerometer's X axis (-1023~1023) onto the degree that the servo rotates (0~180 degree). After mapping, when the board lays flat on the desktop, the angle of the servo will be 90 degrees. The corresponding relations show as Table 4.1.1

Degree of servo	0°	90°	180°
micro: bit accelerometer	-1023	0	1023

Find "map…from…to…" block in "Operators" module.

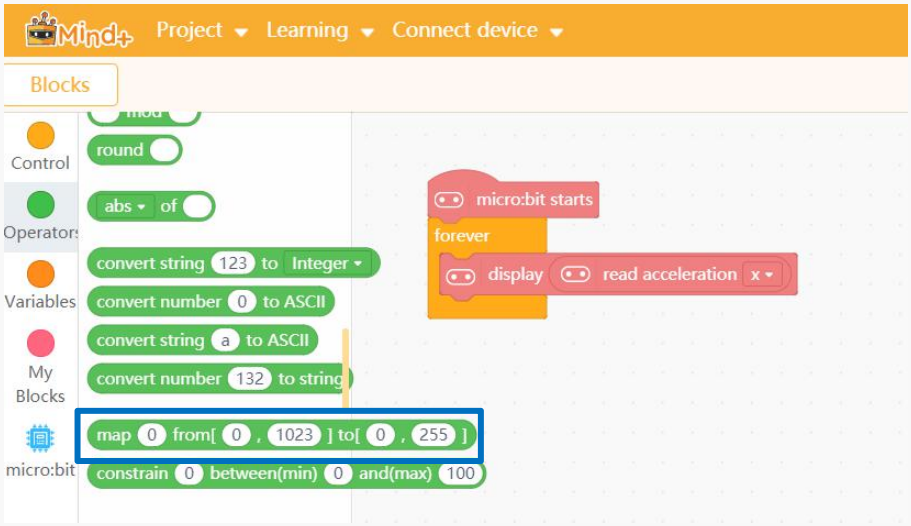


Figure 4.1.7 "map" block

4. Click Expansions->Actuator->Micro Servo and back to the former interface.

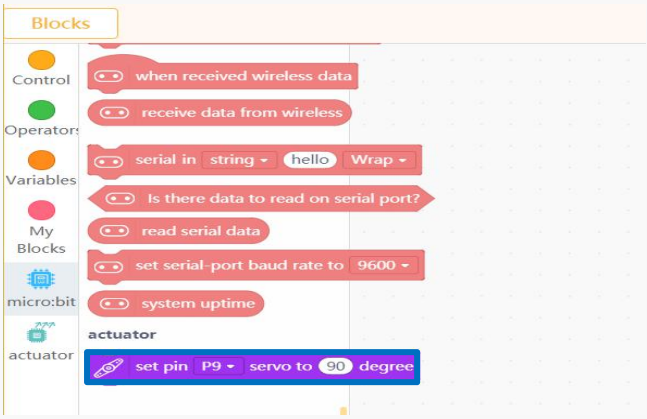


Figure 4.1.8 Servo Module

5. Drag "set pin servo to… degree" block to script area, and put it inside the "forever" block, then nest the "map…" block into it. Shown as below:

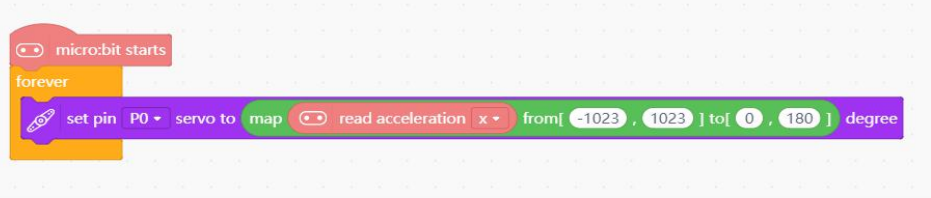


Figure 4.1.9 Run the Programs

6. Fix the servo onto the expansion board with double-side adhesive tape so that they both remain relatively still.

Figure 4.1.10 Fix servo and board

Figure 4.1.11 Keep Still

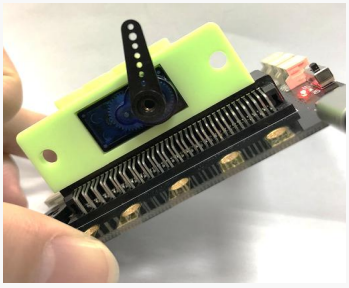
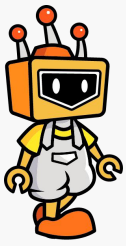


Figure 4.1.12 Tilt to left

5.Invention of Micko



Dr.M, I can't believe my eyes, the self-balancing device is so amazing! I found a Compass module in Mind+. Would micro: bit can be a compass?

read compass orientation

Yes, sweetie, now you can try using the electronic compass in micro: bit and make one compass!

The servo of micro: bit ranges from 0~180°, while the compass need to rotate from 0~360°, so we can directly use "display" and "read compass orientation" to get the position. Read the operation instruction of micro: bit LED panel, then you can make it!

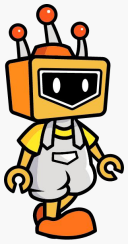
My Design

My Program

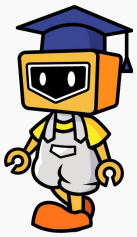
Project 2: DJ panel

1.Story of Micko and Dr.M

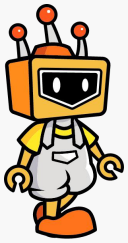
Christmas is coming, what kind of program will you perform?



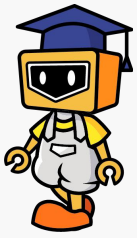
Dr.M, I want to play a new trick this year. I want to be a DJ and present dynamic music with colorful lights.



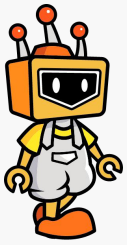
Sounds good, I guess you must need one DJ bandstand.



Yes! And I want to make one. Dr.M. Could you please help me make it?

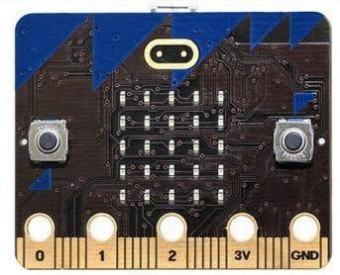


Of course, I am so glad to lend you a hand.

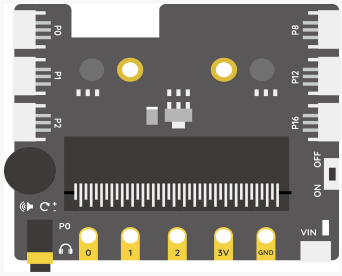


Wonderful! Time is flying, let's begin the work.

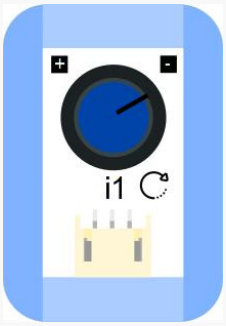
2.Dr.M's Pocket



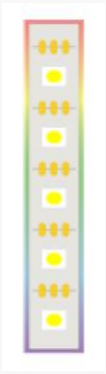
1 × micro: bit



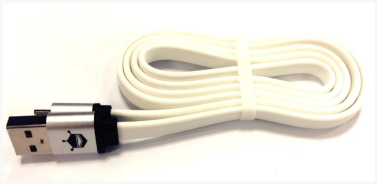
1 × micro: bit Expansion Board



1 × Knob

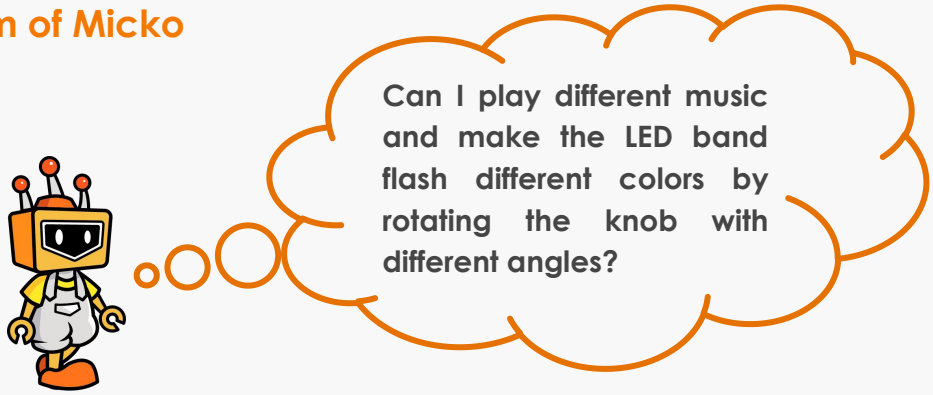


1 × Colorful LED strip



1 × micro USB Cable

3.Dream of Micko



When switching the knob, the analog value of the pin the knob connected changes from 0~1023. We can divide them into 8 small ranges, corresponding to one octave "Do~Do" and the number of LEDs to be lighted.

The input value of Knob(P1)	Sound	The LED strip
0~128	Do	Light up one LED
128~256	Re	Light up two LEDs
256~384	Mi	Light up three LEDs
384~512	Fa	Light up four LEDs
512~640	So	Light up five LEDs
640~768	La	Light up six LEDs
768~896	Xi	Light up seven LEDs
896~1023	Do'	Light up seven LEDs in blue

Table 4.2.1

4.Dr.M's Blackboard

STEP1: Connect the knob to Pin P1 of the expansion board, the LED strip to P2.

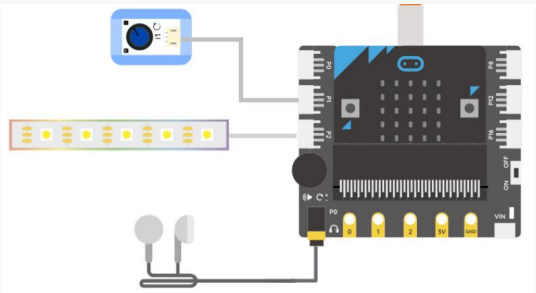


Figure 4.2.1 Connection Diagram

STEP2: The following logic diagram will help you to build a general idea about how to realize a DJ Bandstand.

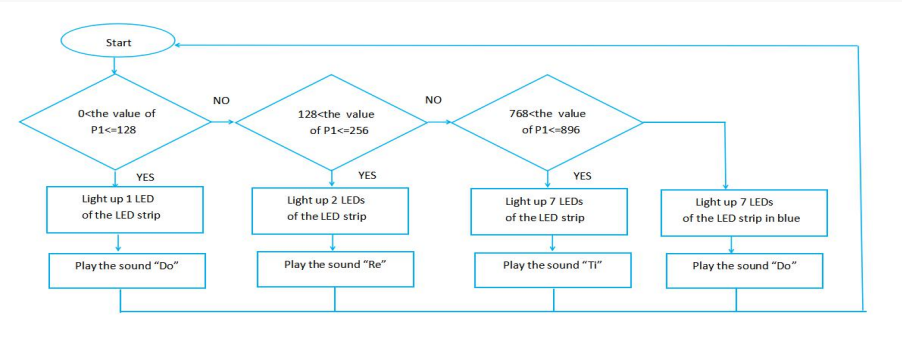


Figure 4.2.2 Flow Chart

STEP3: Programming

1. Take programming "0<knob P1<128" as an example, set the range of Knob P1 through the "and" function in the "Operators" module, then put the set range into the condition of "if...then...else" block, nest the LED strip and sound effect blocks inside the "...then..." function, and add a "wait 1 second" block. Show as below. The rest 6 programs are similar to this one.

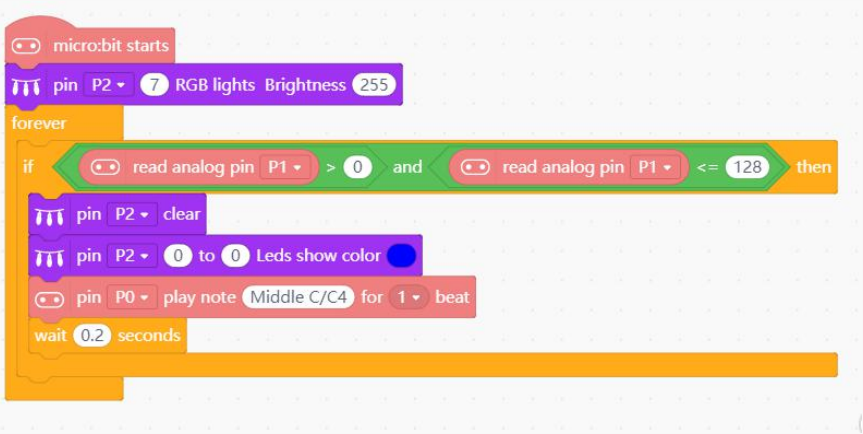
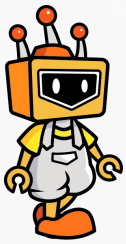


Figure 4.2.3 Sample Program 1

2. Since there are only 7 LEDs, so we can use different light effect to express the sound "Do" in the eighth section as shown below.

Figure 4.2.4 Sample Program 2



Dr.M, this program is not too hard. It is just a little complicated.

Yes, here we can use the copy and paste in coding.

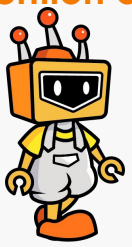
Figure 4.2.5 Run the Program

3. To convenient operation, we can make a little bit improvement, such as add a handle to the knob, or mark the range of each sound.

Figure 4.2.7 Improvement 1

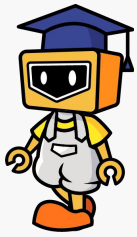
Figure 4.2.8 Improvement

5.Invention of Micko



Dr.M, I am so exciting to make my own DJ bandstand.

My dear, do you want to improve your DJ bandstand? Try using the 5 x 5 LED dot-matrix in micro: bit to display notes!

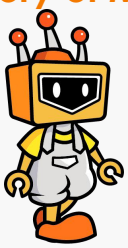


My Design

My Program

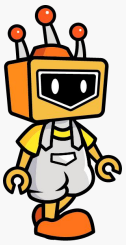
Project 3: Automatic Door

1.Story of Micko and Dr.M



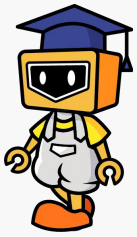
Dr.M, please help me. Yesterday a friend of mine visited me and knocked the door. But that time I was listening music in the room and did not hear the knock. Then he was mad and went away.

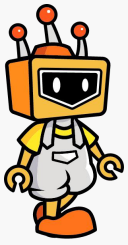
Poor baby, you would better explain it to your friend



Yes, I will. But I am afraid it will happen again in the future. What should I do to avoid this problem?

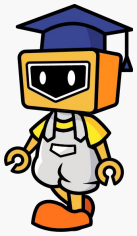
Why not try make a remote-control doorbell? So you can hear the knock even you are in your room.



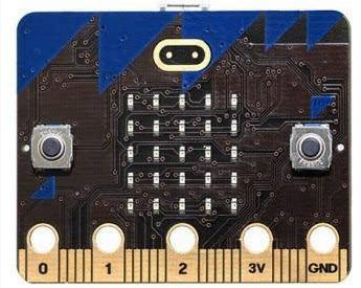


Bravo! Dr.M, how wonderful to make a remote-control doorbell! So I will never need to worry about missing knocks

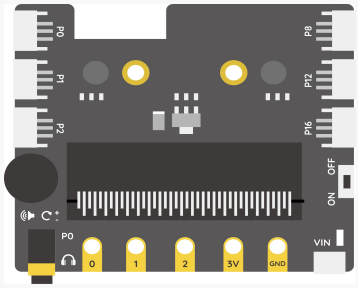
Ha, ha, let's make it!



2.Dr.M's Pocket



2× micro: bit



2× micro: bit Expansion Board

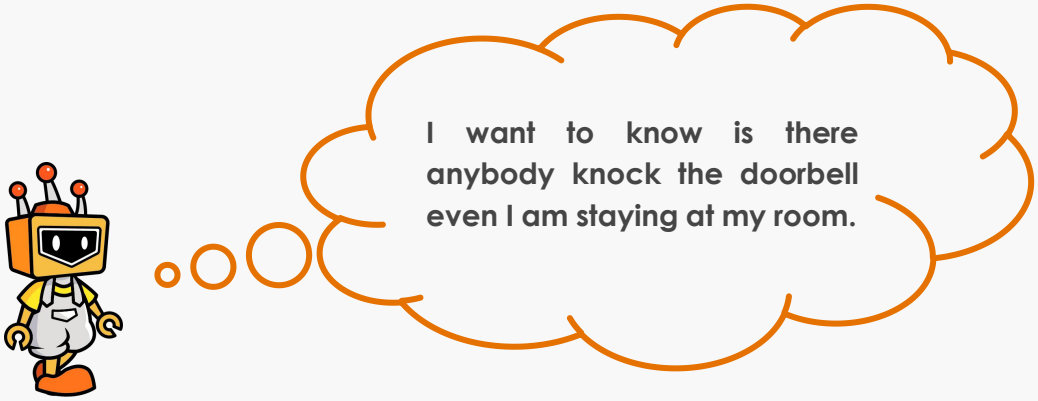
1× LED Module

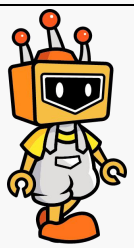
1× Push Button

1× Motion Sensor

2× micro USB Cable

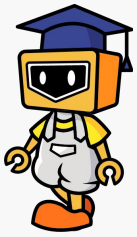
3.Dream of Micko

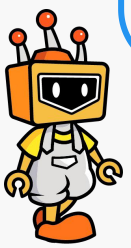




Dr.M, why should we use 2 boards now?

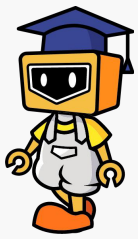
Well, we need to use these two boards to imitate remote doorbell. We will put one of them into outside, which is equals to doorbell. This is the server. And we will put the other one into your room to receive the signals from the server. That is the client. So you can know the outdoor even you are staying your bedroom.



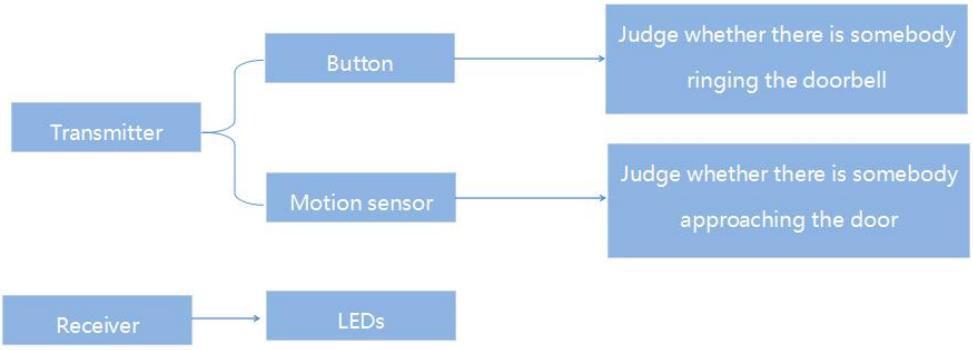


Dr.M, what should we connect between the server and the client?

At this time, we need to apply the knowledge in the flash LED, the notification light and the automatic door. Let's come to the Little Tips to check what should be connected!



Connection Diagram of transmitter and receiver



4.Dr.M's Blackboard

STEP1:

Hardware connection of micro: bit transmitter.

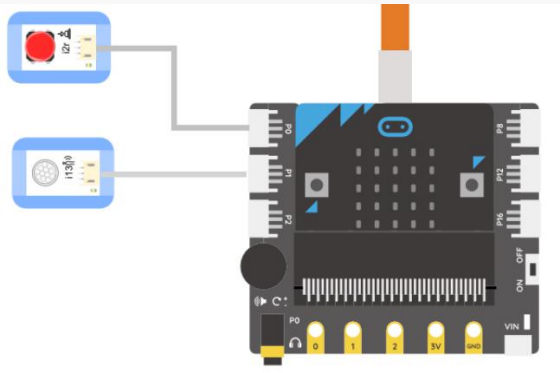


Figure 4.3.1 Transmitter End

Hardware connection of micro: bit receiver

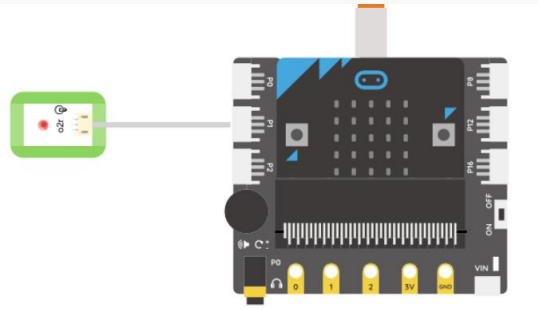
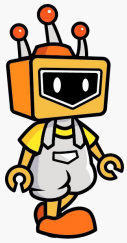


Figure 4.3.2 Receiver End

STEP2: Get to know about wireless transmission module

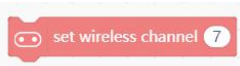


Dr.M, How could we send the signal if someone knock the doorbell from the

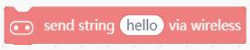
Here we need to learn a new thing: wireless transport module. When the server receives the signals from users, it will pass the signals to the client. So we need to program the server and the client separately.

The common wireless transmission module

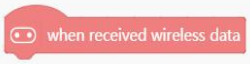
Signal Setting:



Transmitting wireless signal:



Receiving wireless signal:



Notice: the receiver end needs to recognize the received signals so as to execute the related actions. The transmitter can send out both figures and letters.

STEP3: Program the transmitter

1. The following flow chart can give us a roughly idea of how to write the program.

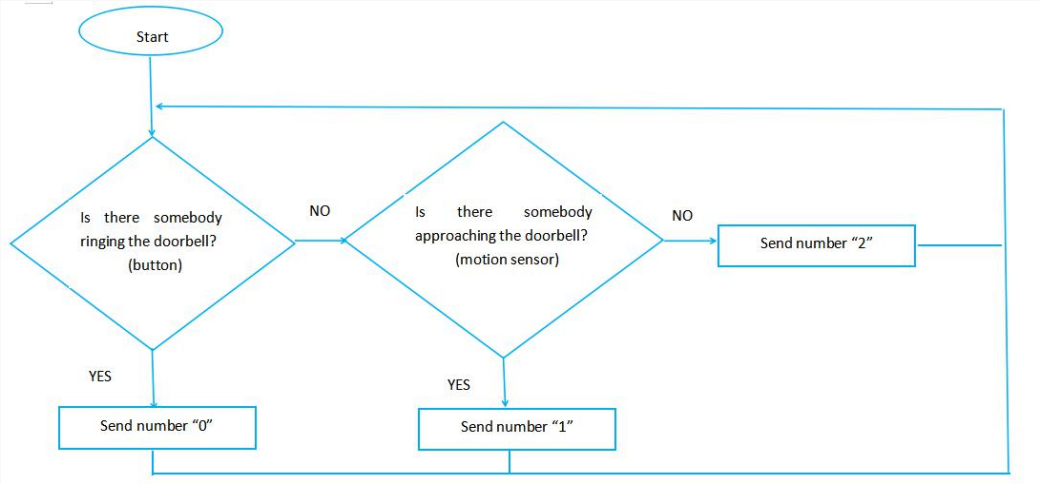
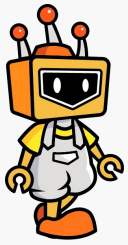
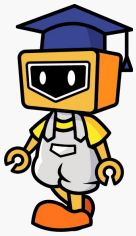


Figure 4.3.3 Transmitter Flow Chart



Dr.M, how to set the wireless connection?

Good question, we must open the wireless communication **module before programming and** set the wireless program.



2. Programming according to the flow chart.

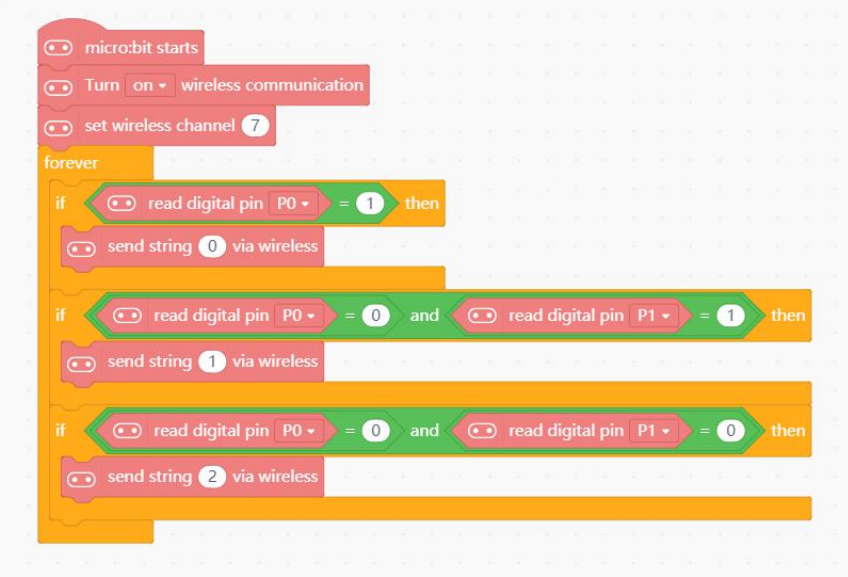


Figure 4.3.4 Transmitter Programs

STEP4: Program the receiver

1. Draw the flow chart, design the receiver programs.

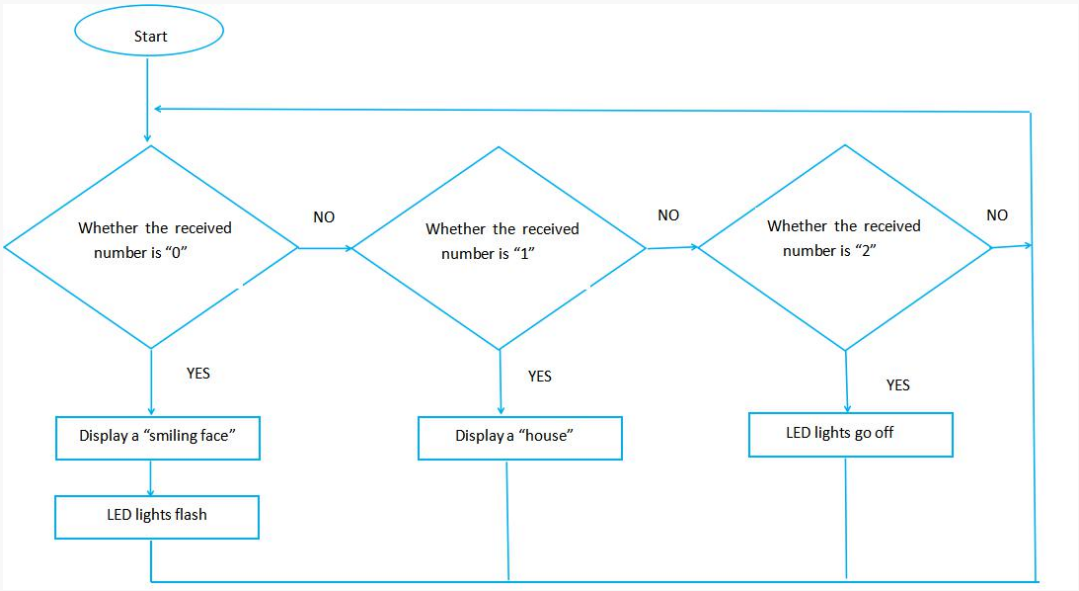


Figure 4.3.5 Receiver Flow Chart

Little friend, here we also need to open the wireless communication **module before programming and** set the wireless program**, and execute the program when the client** receives the wireless signals**.**

2. Programming according to the flow chart.

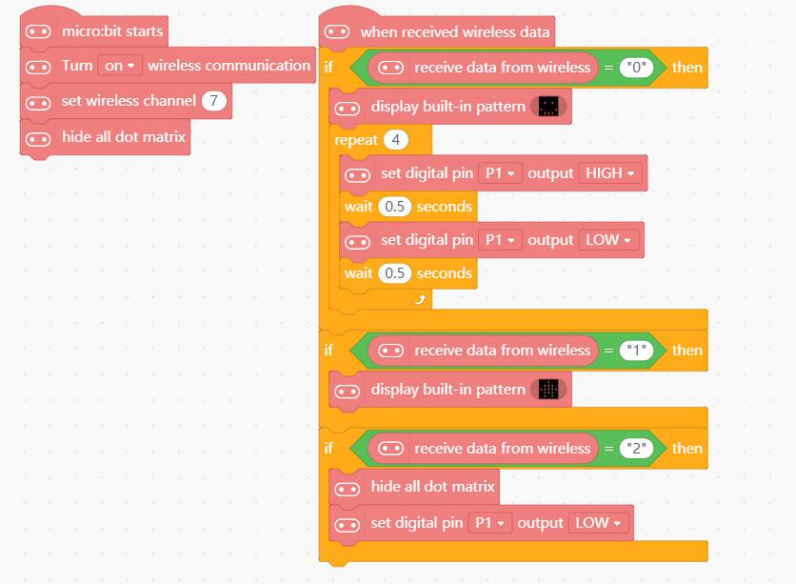
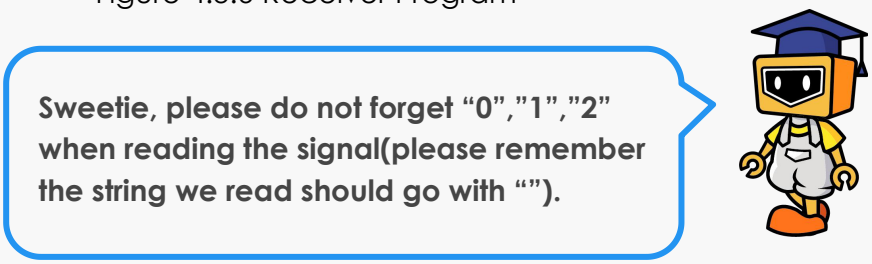


Figure 4.3.6 Receiver Program



Result

1. When there is someone passing by but not ringing the bell:

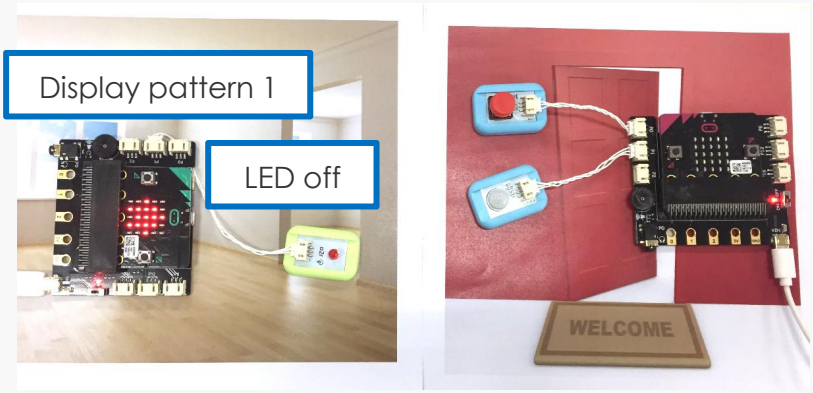


Figure 4.3.7 Result 1

2. When there is someone ringing the bell

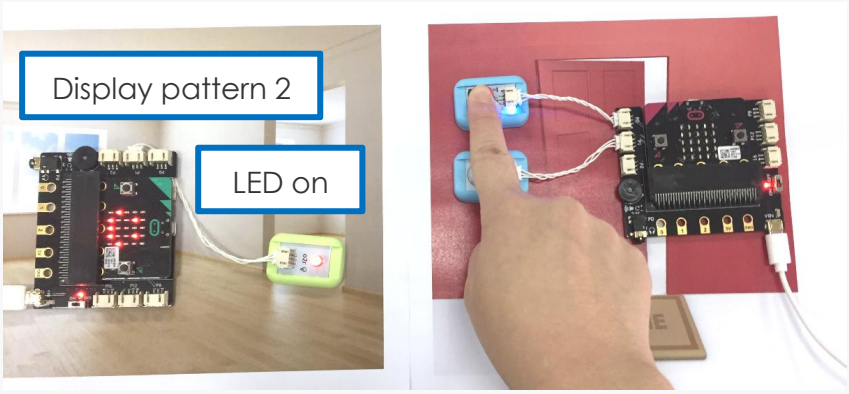
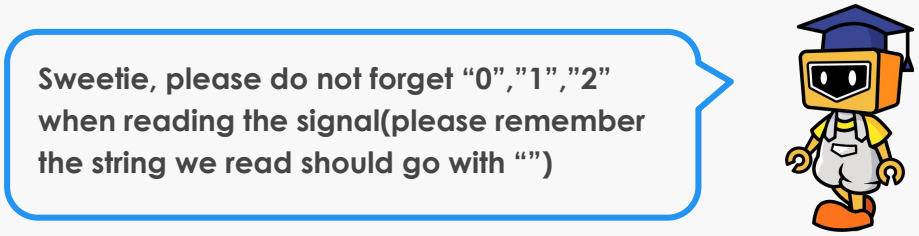


Figure 4.3.8 Result 2

5.Invention of Micko

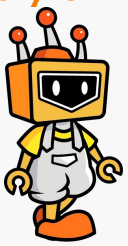


My Design

My Program

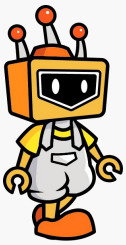
Project4: The Bomb Escape

1.Story of Micko and Dr.M



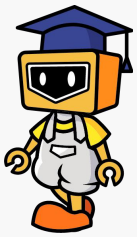
Dr.M, my best friends will visit me.
I want to play games with them.

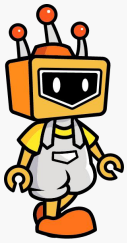
Good idea. Which games would you prefer?



Some interesting games. Such as remote-control cars, and planes. But I want to play new and cool toys.

Well, but why not make it by yourself?

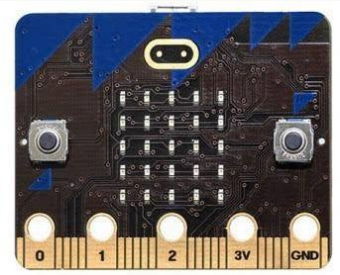




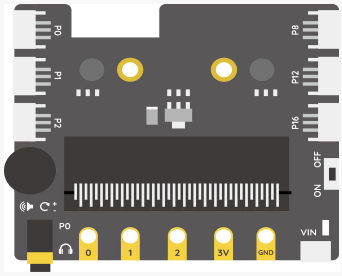
Yes, I was planning to make a game to escape from bomb. This game can test our reaction ability. Dr.M, will you do me a favor?

No problem! It's my pleasure. Let's start to make it.

2.Dr.M's Pocket

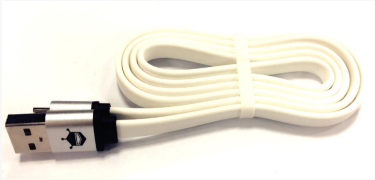


1 × micro: bit



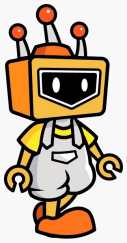
1 × micro: bit Expansion Board

1 × Colorful LED Strip



1 × micro USB Cable

3.Dream of Micko



When we start play the game Bomb, the LEDs also enter the countdown mode at the time. When all LEDs are OFF, which means the countdown is over. The bomb will explode imminently.

If we went to disposal the bomb, we need to lean the micro: bit to the directions that LEDs show.

The explosive ordnance disposal device includes following three parts:

1. Action Recognition: identify if the tilt direction of the micro: bit board is same to the direction its LEDs pointed.
2. LED Strip Countdown: set countdown light effect to the LED strip, turn off 1 LED every 10 seconds.
3. Score System: if the 20 actions are finished before all the LEDs of LED strip go out, you win; otherwise, game over.

4.Dr.M's Blackboard

STEP1: Connect the colorful LED strip to pin P0 of the expansion board.

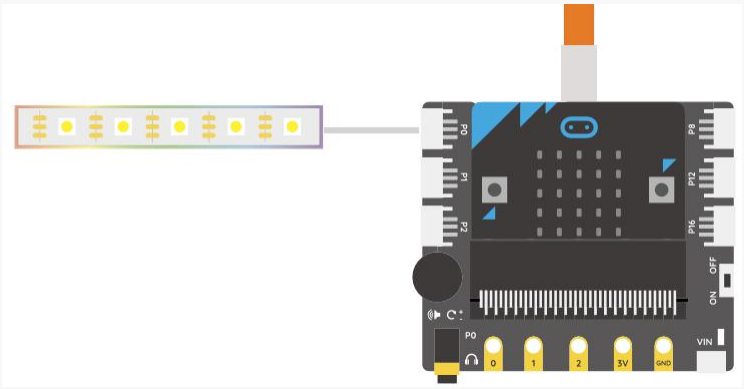


Figure 4.4.1 Connection Diagram

STEP2: Action Recognition

1. Learn the related commands

   Firstly, we need the micro: bit board to generate and display a direction randomly, then tilt the board, if the direction the board tilts to is same to the displayed direction, generate a new random direction, if not, micro: bit LEDs panel keeps displaying the current direction. The flow chart is shown as below:

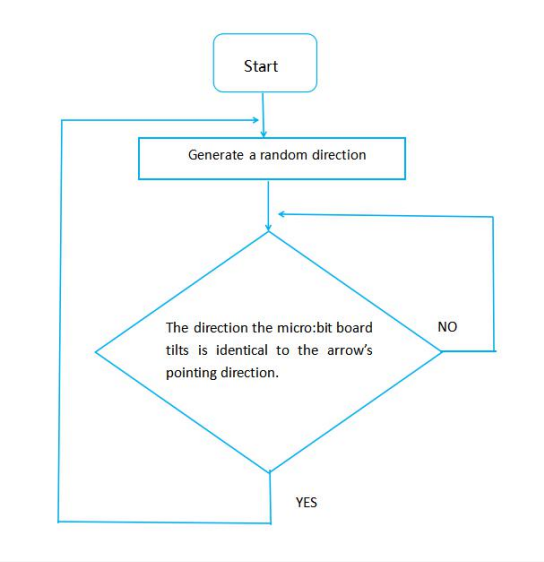
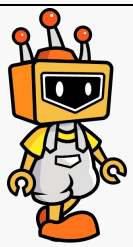
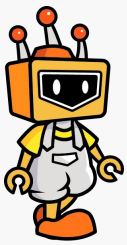


Figure 4.4.2 Action Recognition Flow Chart



Dr.M, how to create a direction in arbitrary?

Here we need to use the module random.

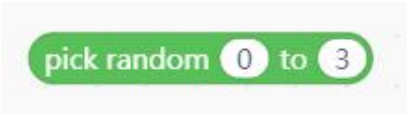


Dr.M, how could we combine the module random with directions?

Such a good question! Here we need to learn something new, that is the module "Create a function", check it at the Little Tips!

Select a random number:

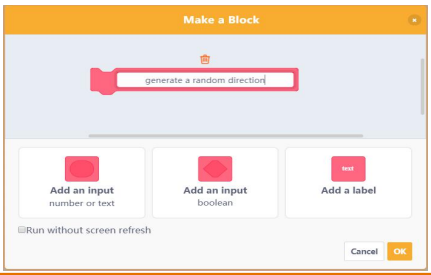
Use the "pick random...to..." function in "Operators" module to select the number in a certain range. Since there are four directions (up, down, left and right), we can use number 0~3 to represent four target directions respectively.



Construct functions:

Use the "make a block" in "My blocks" module to construct a function. When it is constructed, drag the block to script area if we need it.

Now we need to make the four numbers "0, 1, 2, 3" correspond to the four directions "up, down, left and right". We can make a "generate a random direction" block. The constructed block will appear in the left.



Get the current gesture :

Use the "current gesture" function under "micro: bit" module to sense which direction the micro: bit board tilts.

2. Programming

1) Create a variable "target direction" to represent the arrow's direction on micro: bit LED panel, set the variable value to "pick random 0 to 3"

Figure 4.4.3 Set the value of target direction

If you forgot how to create a variable, review it at project 4-light up a colorful LED strip.

2) Create a function "generate a random direction", and make the four numbers "0, 1, 2, 3" correspond to the four directions of the arrow displayed on the LED panel.

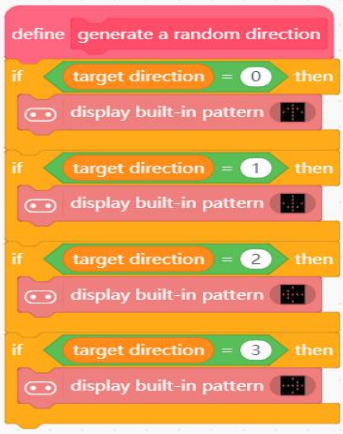


Figure 4.4.4 "generate a random direction" function

3) Make a variable "operation direction" to represent the current direction the board tilts, and use it with "current gesture" to make the four "operation direction" correspond to the four numbers.

Figure 4.4.5 Get the current gesture

4) Here we need to use a "if…then…else". If the operation direction is same to the target direction, then continue to generate a new random direction.

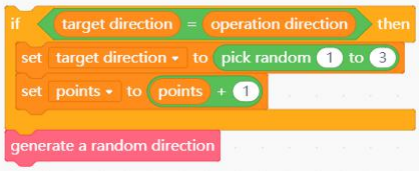


Figure 4.4.6 if…then…else

5) The whole program of this part is shown below:

Figure 4.4.7 Programs of action recognition

STEP3: LED Strip Countdown

1. Draw a program flow chart

The LEDs on the LED strip will go off one by one as the time goes by. The faster you react; the more LEDs keep on at last.

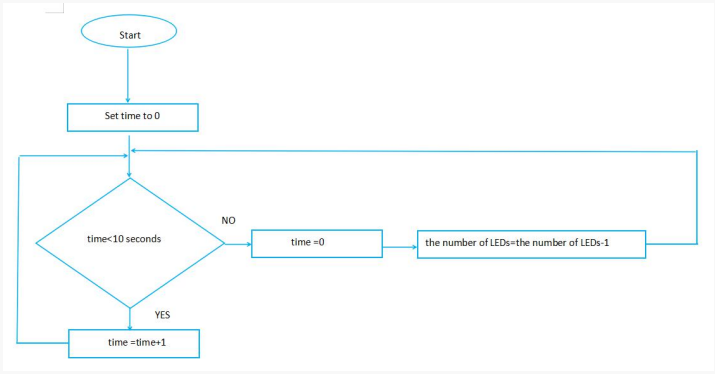


Figure 4.4.8 LED strip countdown flow chart

2. Programming

1) Create variables "time" and "the number of LEDs", set "time" to 0, and set "the number of LEDs" to 7(means 7 LEDs keep on). Combine them with "if…then…else" to write the programs as the flow chart.

Note: micro: bit needs extra time to process the data, so the clock will run a bit slower than we set.

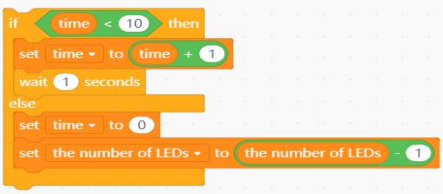


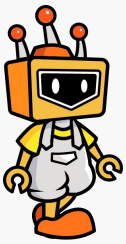
Figure 4.4.9 "if…then…else"

2) Remember to set the LED's lighting effect:

Figure 4.4.10 LED Lighting Effect

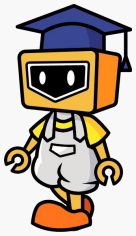
3) The whole programs of the last two parts:

Figure 4.4.11 Whole programs of the last two parts



Dr.M, 10 seconds is a little too long. We want to speed it up!

No problem, if you want to shorten the countdown time, try set the time to 5s or 1s.



STEP4: Score System

1. Draw the flow chart

   If the operation direction is same as the target direction, then we get 1 point. If the number of LEDs (keep on) is over 0 when we got 20 points, we are the winner, the final score is the number of LEDs (keep on). Otherwise, we failed.

Here is the flow chart:

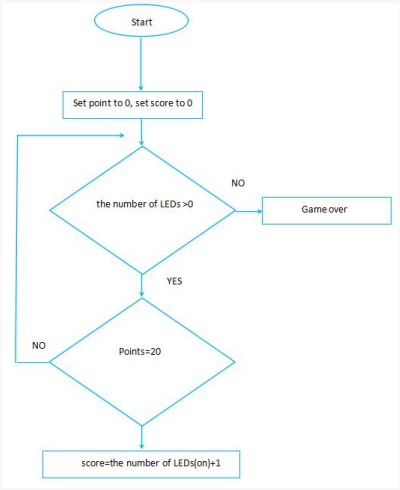


Figure 4.4.12 Flow chart of score system

2. Programming

1) Create variables "points" and "score", set them to 0. Combine them with "if…then…else" function to meet the requirements of the flow chart.

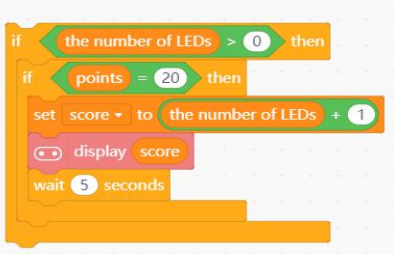
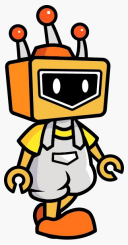


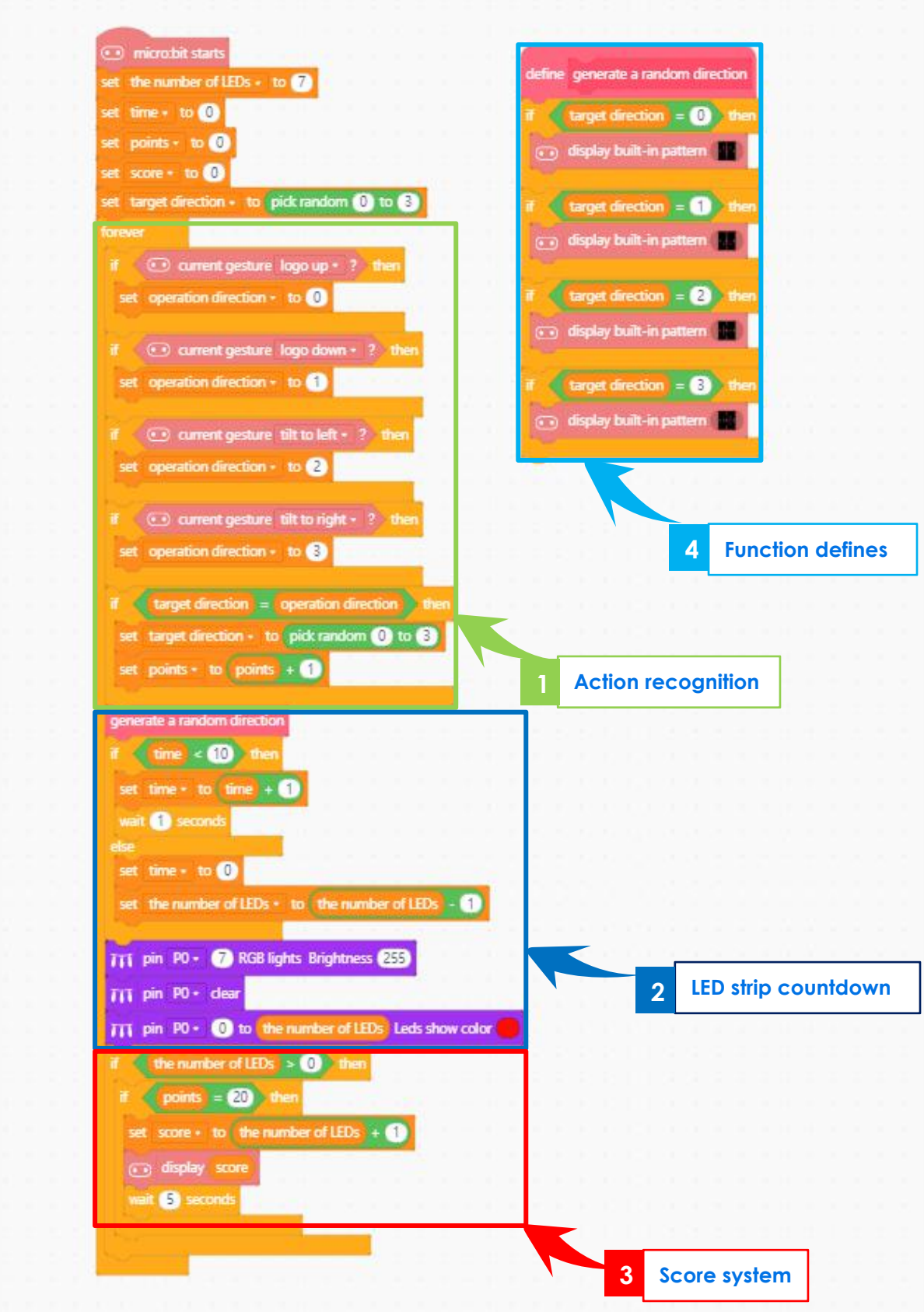
Figure 4.4.13 Part of the Program



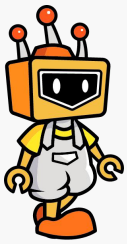
Dr.M, why the final score is equals to the quantity of LEDs + 1?

My sweetie, do you remember? The LEDs are counted from 0. So the number of LEDs turn ON in reality is just 1more than the number in coding.

2) The final program is shown as below:

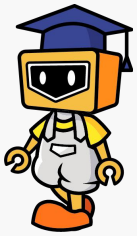


5.Invention of Micko



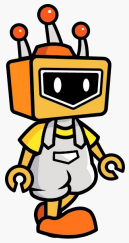
Dr.M, the game Bomb is so interesting! We all have fun!

Little friend, if you want to upgrade this game, you can create a mark in the end of the game. Come to explore this interesting work with your friends!



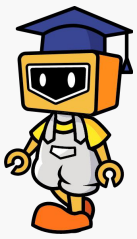
My Design

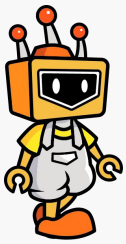
My Program



Dr.M, micro: bit and Mind+ are so interesting! They are so powerful!

My little friend, what we played is just the tip of the iceberg. There are so many functions in micro: bit and Mind+, and they are waiting for your exploration.



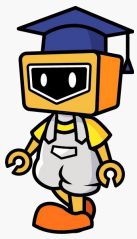


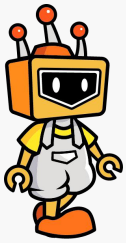
Dr.M, thanks for all your help all this time. I will explore it with enthusiasm as always. But where could I find you and ask you my problems?

Do not worry, I am always with you on line. If you got any problems and any good ideas to share, come to visit the

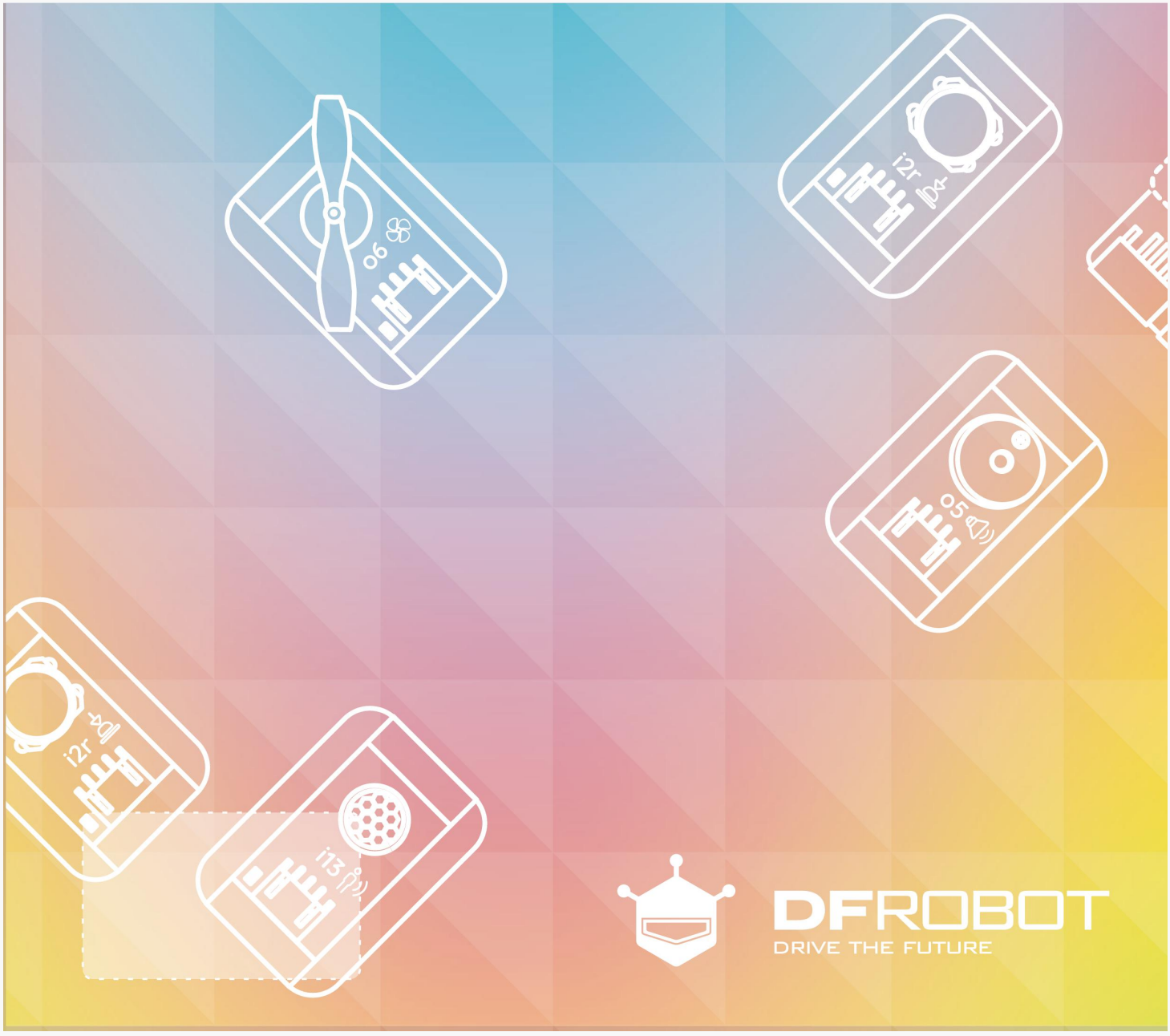
DFRobot Community: www.dfrobot.com

In this community, you can find many makers like you. Wish DFRobot community can record your growth.





Wow! So wonderful! Dr.M, see you in the DFRobot community, wishing I could find more friends in the community.


DFROBOT
DRIVE THE FUTURE