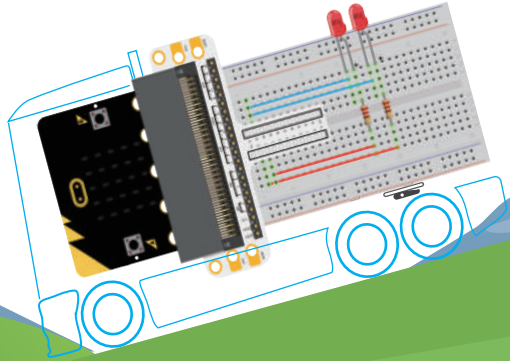
micro:bit

Starter Kit Guide

ELECFREAKS
DEVOTE TO OPEN HARDWARE



10+ YEAR

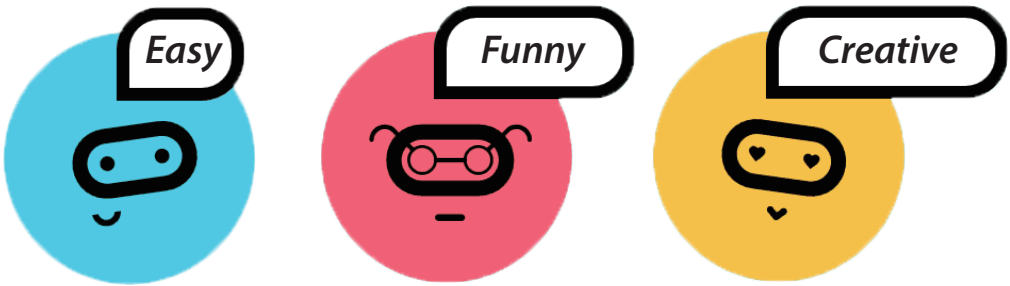
Your First Micro:bit Kit

Funny Easy Suitable for Your Beginning

STEM

open hardware

micro:bit



ElecFreaks Micro:bit Starter Kit is designed for people at the entry level of electric circuit and programming study. This kit contains some basic parts like LED, button, buzzer, temperature sensor, servo and motor etc.. You can use it to design circuit. With the help of Micro:bit programming skills, you can make your circuit become more animated. It is a good companion for you to enter into a wonderful electronic world.

PRODUCT CONTENT

1.LED Scroller

Component List

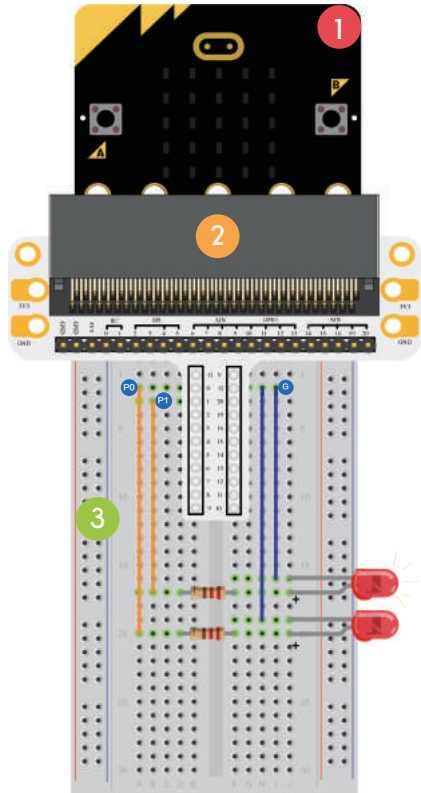
1. 1 x Micro: bit Board
2. 1 x Microbit Breadboard Adapter
3. 1 x Breadboard
4. 2 x Red LED
5. 2 x 100Ω Resistor

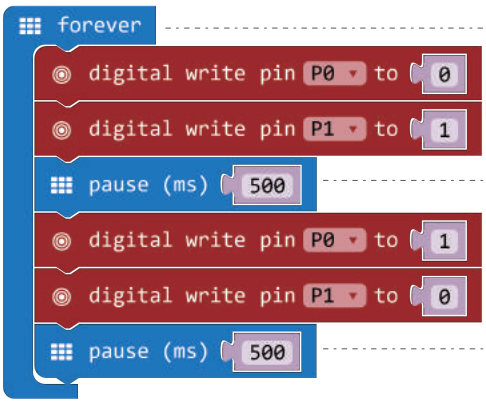
4

5

Description

In this course, we are going to use micro:bit to make 2 LED beads twinkle alternatively.





1. Within *forever*, program runs circularly.
2. Set low voltage to *P0* (LED0 off);
   Set high voltage to *P1* (LED1 on) .
3. Delay time for 500ms.
4. Set high voltage to *P0* (LED0 off) ;
   set low voltage to *P1* (LED1 on) .
5. Delay time for 500ms.
6. Download the program into micro:bit.

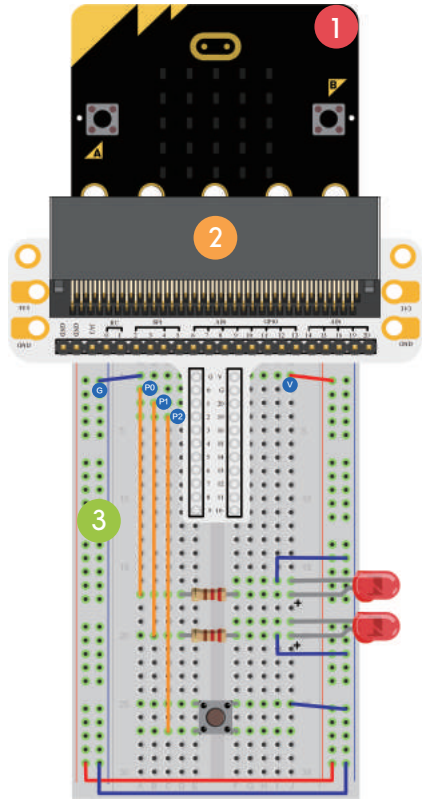
Result: You will see two LED beads flash alternatively.

Question: How to make an RGB traffic light ?

2.Button

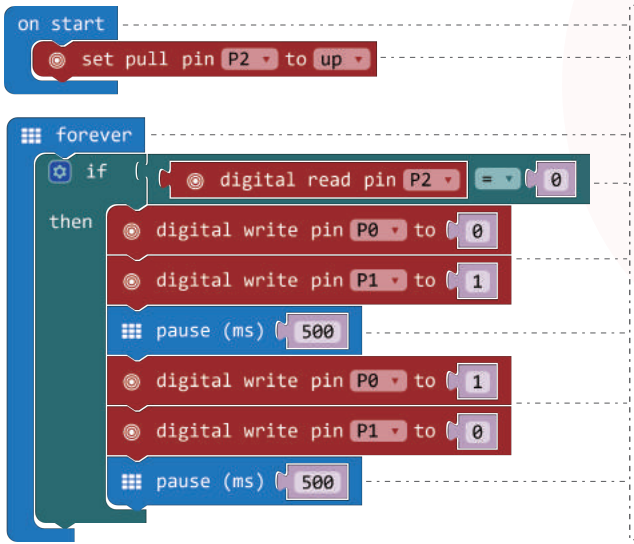
Component List

1. 1 X Micro:bit Board
2. 1 X Microbit Breadboard Adapter
3. 1 X Breadboard
4. 2 X Red LED
5. 2 X 100Ω Resistor
6. 1 X Momentary Pushbutton Switch



Description

In this course , we will use a button to control LED flash. Press down the button, 2 LED beads flash in turns; release the button, 2 LED beads stop flashing.



1. Block *on start* runs only once to start the program.
2. Set *P0* to be a pull-up.
3. Within *forever*, program runs circularly.
4. If *P2* is in low voltage, which means the button is pressed, the program will start to run in block *then*.
5. Set low voltage to *P0*, LED0 off ; set high voltage to *P1*, LED1 on.
6. Delay time for 500ms.
7. Set high voltage to *P0*, LED0 on; set low votage to *P1*, LED1 off.
8. Delay time for 500ms.
9. Download the program into micro:bit.

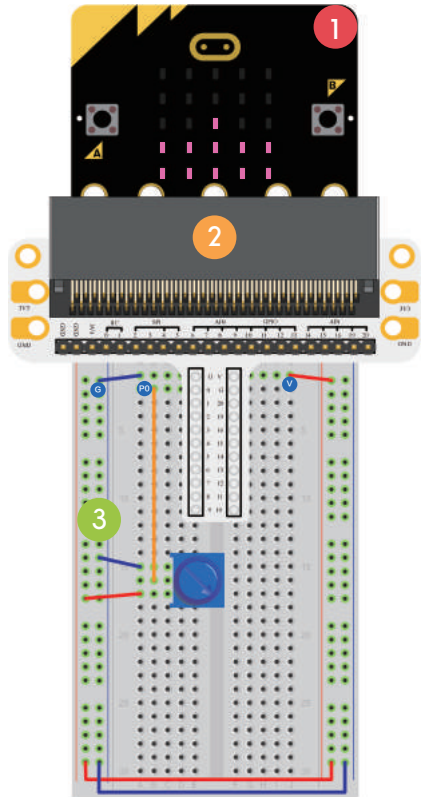
- Result: Press down the button, LED will flash alternatively.
- Question: How to light red LED with the button pressed and light green LED with the button released?

3.Trimpot

Component List

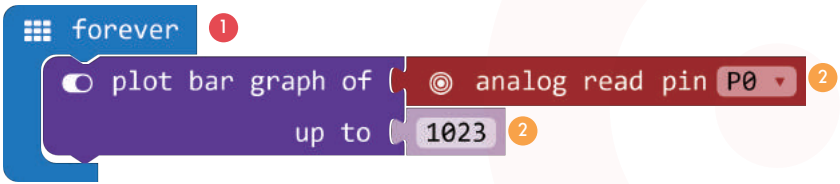
1. 1 x Micro:bit Board
2. 1 x Microbit Breadboard Adapter
3. 1 x Breadboard
4. 1 x 10kΩ Trimpot

4



Description

In this course, we are going to read the output voltage of trimpot and display it on micro:bit screen with bar chart.



1. Within *forever*, program runs circularly.
2. Read the analog voltage of *P0*(0 to 1023） and display it on the LED screen with bar graph.
3. Download the program into micro:bit.

Result: Rotate trimpot button, voltage value will be displayed on micro:bit screen with bar graph. When the voltage is 0, LED screen displays a pixel spot only. When it is 3.3V, the whole screen will be illuminated.

Question: How to use trimpot to adjust the brightness of a LED ?

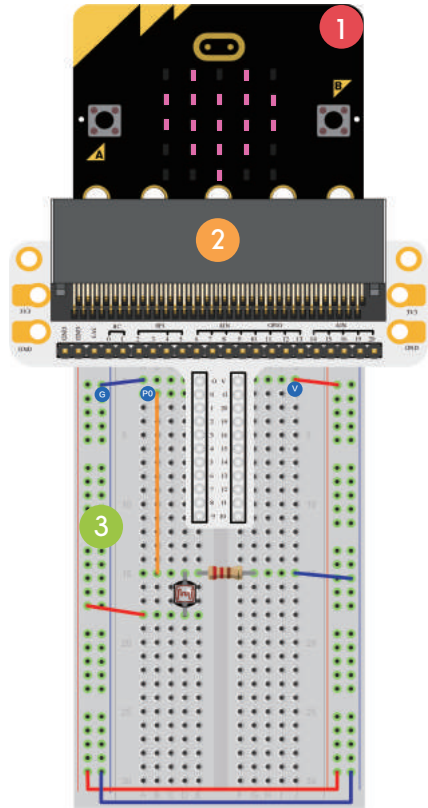
4.Photocell

Component List

1. 1 x Micro:bit Board
2. 1 x Microbit Breadboard Adapter
3. 1 x Breadboard
4. 1 x Photocell
5. 1 x 10kΩ Resistor

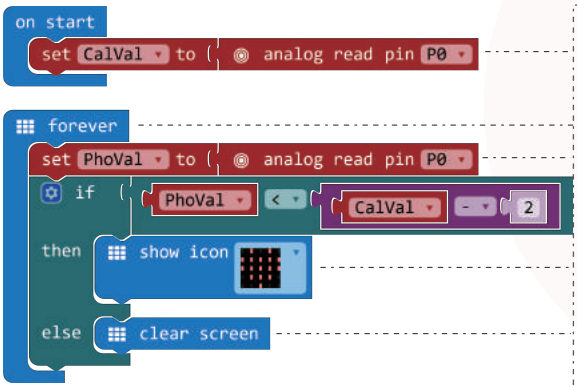
Description

In this course, we are going to use photocell to control the brightness of micro:bit screen.



4.Photocell

Step



1. Block *on start* runs only once to start the program.
2. Read *P0*(photocell) analog voltage and store into *CalVal*. It is a reference for current environment brightness.
3. Program in block *forever* runs circularly.
4. Read *P0*(photocell) analog voltage and store into *PhoVal*.
5. When *PhoVal<CalVal-2* is right, then the environment light becomes dim. It will implement program in block *then*.
6. Display heart icon.
7. When *PhoVal<CalVal-2* is wrong, the screen closed.
8. Download the program into micro:bit.

Note: Reset micro:bit, it will calibrate the reference value according to current brightness. To run the program properly, we must start with the light turned on.

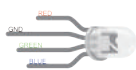
◉ Result: Light on, nothing appears on micro:bit screen. Light off, a heart icon appears.

◉ Question: How to use photocell to control an LED?

5.RGB LED

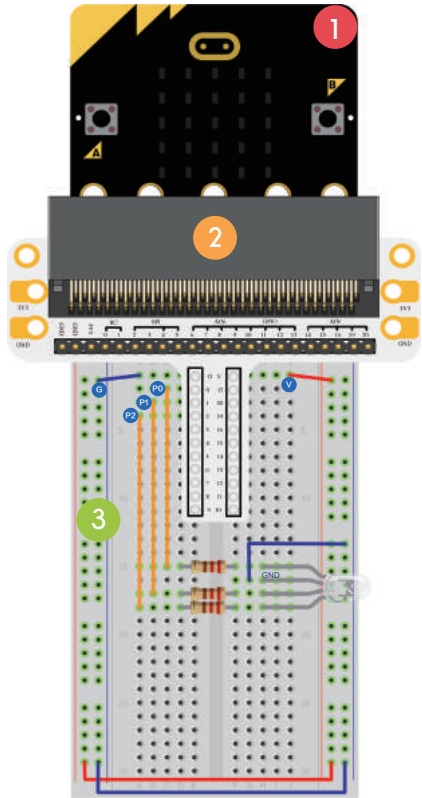
Component List

1. 1 x Micro:bit Board
2. 1 x Microbit Breadboard Adapter
3. 1 x Breadboard
4. 1 x RGB LED
5. 3 x 100Ω Resistor



4

5

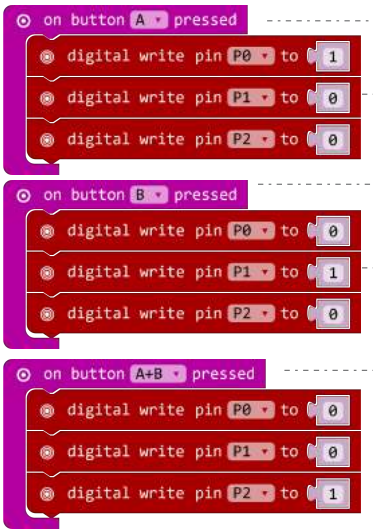


Description

In this course, we are going to make RGB LED gradually shift its light among red, green and blue.

5.RGB LED

Step



1. Press button *A* to run the block.
2. RGB LED emits red light.
3. Press button *B* to run the block.
4. RGB LED emits green light.
5. Press button *A+B* to run the block.
6. RGB LED emits blue light.
7. Download the program into micro:bit.

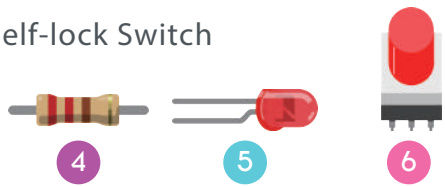
◉ Result:Button A brings red light, button B green light, button A+B blue light.

◉ Question: How to realize soft gradient for RGB LED light?

6.Self-lock Switch

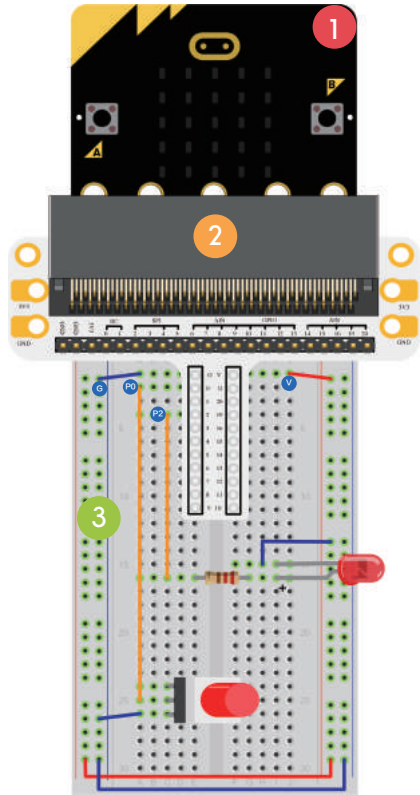
Component List

1. 1 x Micro:bit Board
2. 1 x Microbit Breadboard Adapter
3. 1 x Breadboard
4. 1 x 100Ω Resistor
5. 1 x Red LED
6. 1 x Self-lock Switch



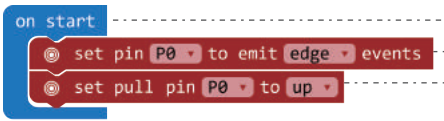
Description

In this course, we are going to use self-lock switch to control LED light.

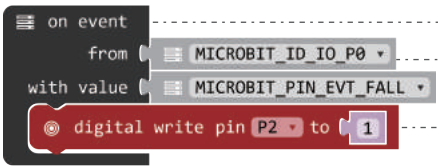


6.Self-lock Switch

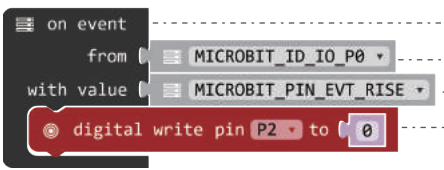
Step



1. Block *on start* runs only once to start the program.
2. Configure the type of events emitted by *P0*.
3. Set *P0* to be a pull-up .



4. Raise an event .
5. Set *P0* as an event emit port.
6. Falling edge is effective(switch pressed).
7. Set high voltage to *P2*(LED on).



8. Raise an event.
9. Set *P0* as an event emit port.
10. Rising edge is effective (switch released).
11. Set low voltage to *P2*(LED off).
12. Download the program into micro:bit.

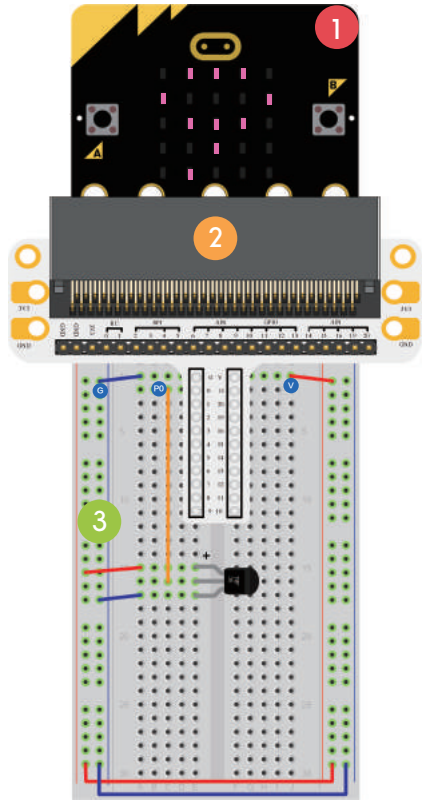
◉ Result: Press down self-lock switch, LED turned on; press again, LED turned off.

◉ Question: How to control micro:bit screen with self-lock switch?

7.Temperature Sensor

Component List

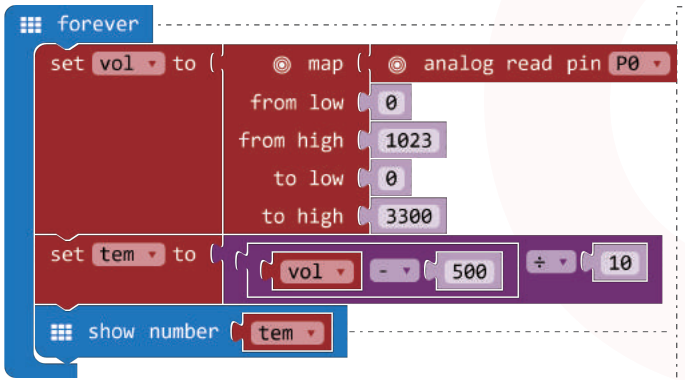
1. 1 x Micro:bit Board
2. 1 x Microbit Breadboard Adapter
3. 1 x Breadboard
4. 1 x TMP36 Temperature Sensor



Description

In this course, we are going to learn analog temperature sensor–TMP36 and display its data on the micro:bit.

7.Temperature Sensor



Step

1. Within *forever*, program runs circularly.
2. Map *P0* analog voltage into actual voltage(mV).
   The analog value is 0 to 1023.
   So *from low* is *0*, *from high* is *1023*.
   The basic voltage is 3300mV.
   Relatively *to low* is *0*, *to high* is *3300*.
3. Convert the value of variable voltage into temperature value.
4. Temperature display.
5. Download the program into micro:bit.

Note: You can calculate the temperature value of TMP36 according to the formula bellow.

$$\text{Temperature}(℃) = \frac{\text{Output voltage(mV)-500}}{10}$$

◎ Result: You will see two LED beads flash alternatively.

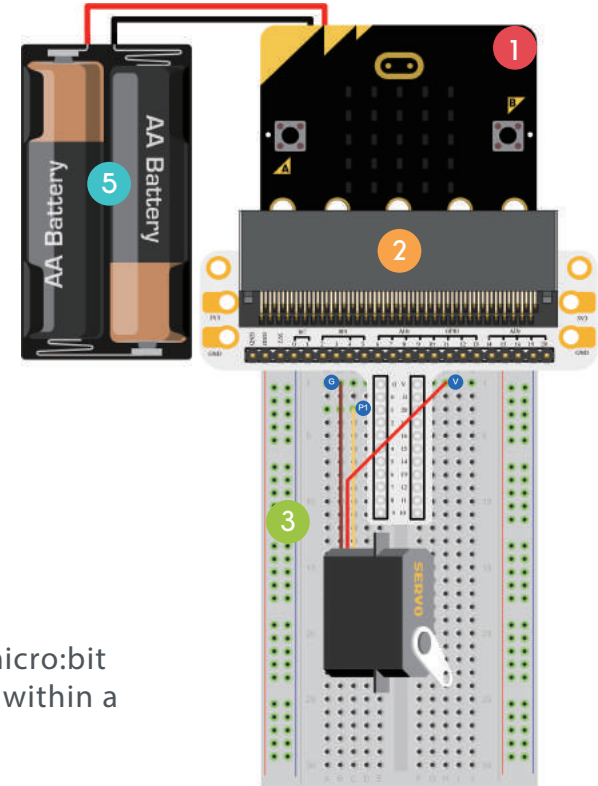
◎ Question: How to display Fahrenheit temperature on micro:bit ?

8.Servo

Component List

1. 1 x Micro:bit Board
2. 1 x Microbit Breadboard Adapter
3. 1 x Breadboard
4. 1 x Mini Servo
5. 1 X Battery Holder

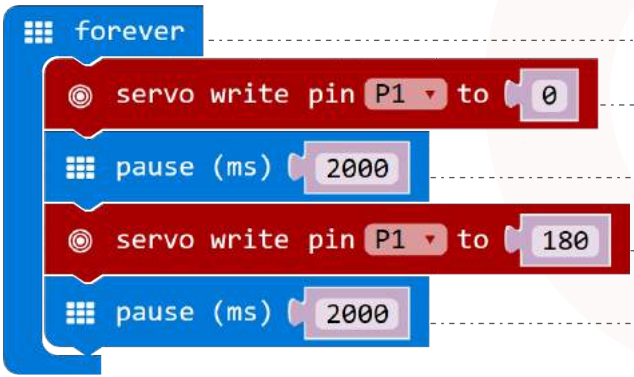
4



Description

In this course, we are going to use micro:bit to make a servo rotate continuously within a travel range.

Step



1. Within *forever*, program runs circularly.
2. Rotate servo to *0* degree.
3. Delay time for 2000ms.
4. Rotate servo to *180* degree.
5. Delay time for 2000ms.
6. Download the program into micro:bit.

◉ Result: We can see the servo rotating from 0 degree to 180 degree.

◉ Question: How to make a dial thermometer with temperature sensor and servo?

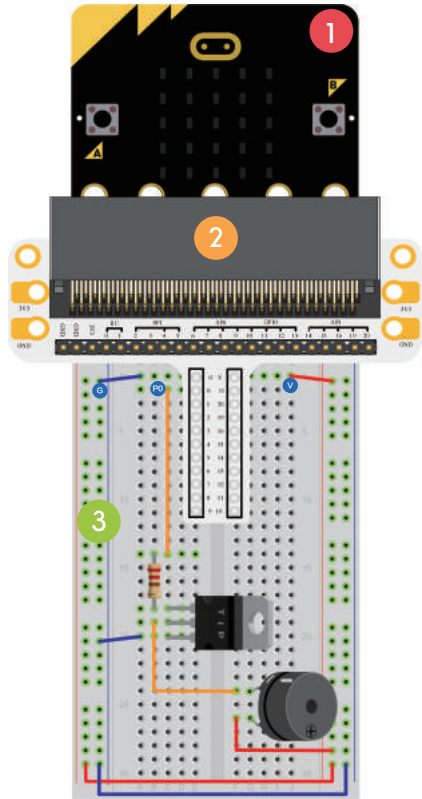
9.Buzzer

Component List

1. 1 x Micro:bit Board
2. 1 x Microbit Breadboard Adapter
3. 1 x Breadboard
4. 1 x Mini Speaker (Buzzer)
5. 1 x NPN Transistor
6. 1 x 100Ω Resistor

Description

In this course, we are going to use micro:bit to drive a buzzer.



9.Buzzer

Step

1. Within *forever*, program runs circularly.
2. Play tone *Middle C*.
3. Delay time 100ms.
4. Play tone *Middle E*.
5. Delay time 100ms.
6. Play tone *Middle G*.
7. Delay time 100ms.
8. Play tone *Middle E*.
9. Delay time 100ms.
10. Download the program into micro:bit.

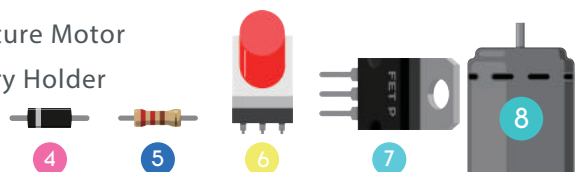
◉ Result: We can hear the waving rhythm from the buzzer.

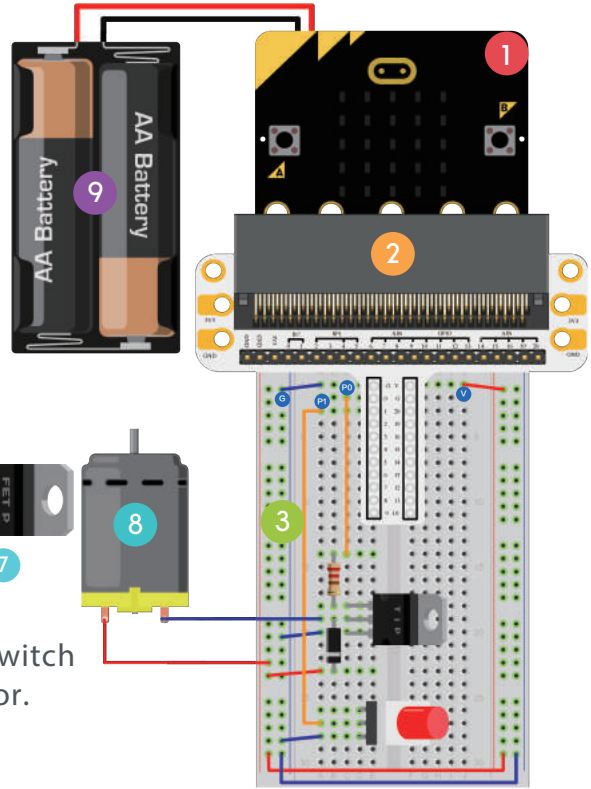
◉ Question: How to play the song of *Little Stars* with micro:bit?

10.Motor

Component List

1. 1 x Micro:bit Board
2. 1 x Microbit Breadboard Adapter
3. 1 x Breadboard
4. 1 x Diodes
5. 1 x 100Ω Resistor
6. 1 x Self-lock Switch
7. 1 x NPN Transistor
8. 1 x Miniature Motor
9. 1 X Battery Holder

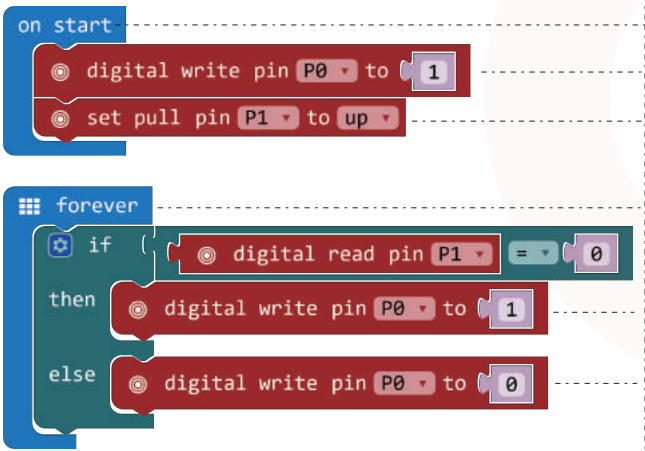




Description

In this course, we are going to use a switch to control the start and stop of a motor.

Step



1. Block *on start* runs only once to start the program.
2. Set high voltage to *P0*.
3. Set pull to *P1*(self-lock switch).
4. Within *forever*, program runs circularly.
5. Judge *P1* voltage. Low voltage means switch is pressed.
6. Once the switch pressed, set high voltage to *P0* , the motor starts running.
7. Once the switch released, set low voltage to *P0*, the motor stops running.
8. Download the program into micro:bit.

Note: Since micro:bit voltage is 3.3V only, it may not enough to support fan sometimes. To make fan run, you have to stir its blade for startup.

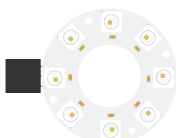
◉ Result: Press switch, the motor runs; release switch, the motor stopped.

◉ Question: How to use trimpot to control the motor speed?

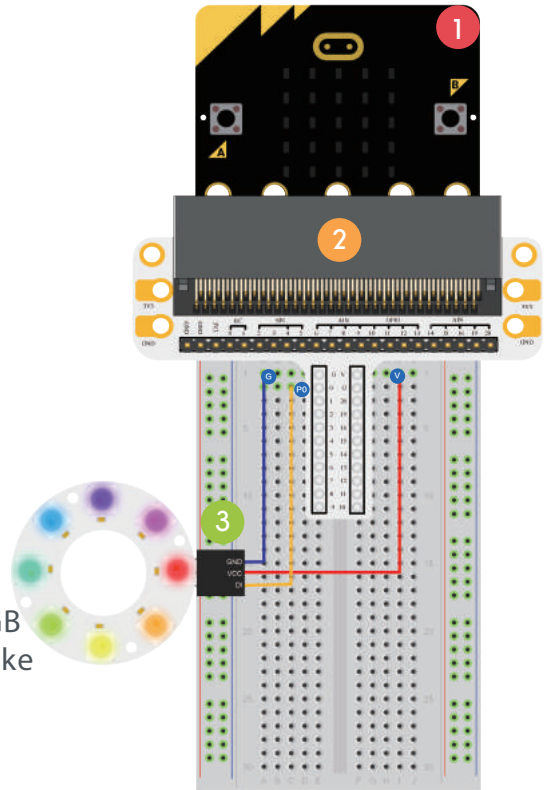
11.Rainbow LED

Component List

1. 1 x Micro:bit Board
2. 1 x Microbit Breadboard Adapter
3. 1 x Breadboard
4. 1 x 8 RGB Rainbow LED Ring



4

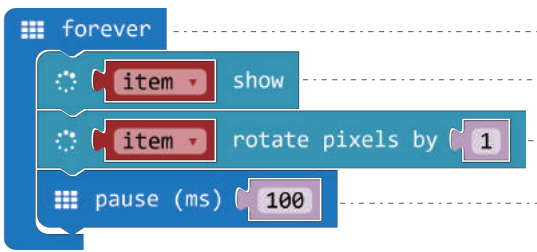


Description

In this course, we are going to drive 8 RGB Rainbow LED Ring with micro:bit and make it realize rainbow color gradual change.

11.Rainbow LED

Step



1. Search and add *neopixel* library from *Add Package*.
2. Block *on start* runs only once to start the program.
3. Initialize LED ring.
4. Set rainbow color parameters for 8 LED beads.
5. Within *forever*, program runs circularly.
6. Make the ring emit the designated color.
7. Move the color data of the ring for a pixel point.
8. Delay time for 100ms.
9. Download the program into micro:bit.

◎ Result: We can see a rainbow rotating on the LED ring.

◎ Question:How to make the ring blinking like an eye?

For More Information

Please visit

www.elecfreaks.com/11017.html

ABOUT ELECFREAKS

DEVOTE TO OPEN HARDWARE

ELECFREAKS was founded by a group of electronic enthusiasts in 2011. It is ltocated in Shenzhen, China.

We mainly devote to provide the superior open hardware and service to makers.

Our goal is to make creation become more convenient, easier and flexible.

We focus on developing compatible accessories and modules for open source platforms such as Arduino, Raspberry Pi, Micro:bit etc..

We open all documents about schematic , source code, user guide etc..

We create teaching blogs and video tutorials with content covering from starters to senior players.

Center on users. We focus on users' experience. We aim to provide you best products and services you need.

Emphasis on quality and cost performance ratio.

Efficient. Fast delivery and fast product update speed.

Sincere and trustful. More than 6 years' development with perfect after-sales support and product quality guarantee.

We have senior engineer team with strong development ability.

We have more than 1000 SKU, rich product categories.

We have more than 500 WIKI and blogs, abundant and complete product information and tutorials.

We have above 100 distributors all over the world.

www.elecfreaks.com