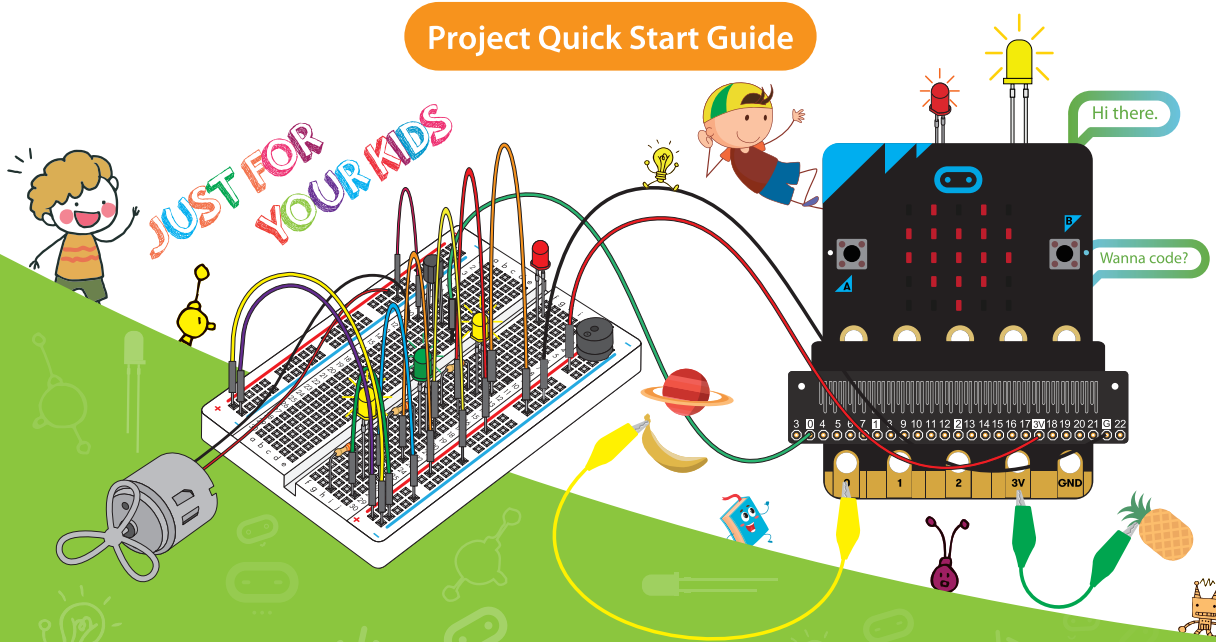
ELECROW

10 Lessons

Elecrow Basic Kit for BBC Micro:bit

Project Quick Start Guide



www.elecrow.com

About Micro:bit Basic Kit

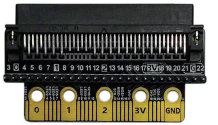
This **Micro:bit basic kit** is created to help students get started with electronics and programming. It has frequently used DIY electronics parts and step-by-step lessons.

Table of Contents

Parts List Introduction

1. Micro: bit GPIO extension board

This is a GPIO extension board for micro: bit makes it convenient to connect other devices to micro: bit.



2. Micro USB cable

Not only used for power supply, but also for data transmission.

3. Battery case

This is a unique battery case for micro: bit, which can make you take your micro: bit anywhere you want.

4. Red/ yellow/ green LED

Those blinky things, play a basic role in our lessons.

5. RGB LED

It comes with separate red, green, and blue LED chips inside, color-swirl is fun.

6. Motor with fan

It is a standard “130 size” DC motor, comes with a lovely fan.

7. Buzzer

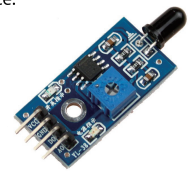
The active buzzer has built-in oscillating source, so it will keep beeping after it is electrified.

8. Potentiometer

A potentiometer is a simple knob that provides a variable resistance.

9. Flame sensor

It works as a fire detector, which can be used to find fire source.



10. Resistor

A resistor is a passive two-terminal electronic component that implements electrical resistance as a circuit element, the value we used in our lesson is 330Ω, 1kΩ, 10kΩ.

11. S9013 triode

A triode is an electronic amplifying vacuum tube, it plays a role to amplify current to drive the buzzer and motor in our lessons.

12. Self-lock switch

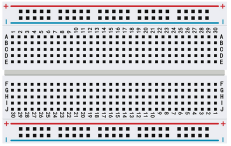
It also named as latching switch, outputs a high signal after first press; outputs low signal after second press.

13. Button

This momentary button outputs a logic high when pressed and logic low when released.

14. Breadboard

A half size breadboard with dimension 8.3 x 5.5cm, just hooks what components you need on it.

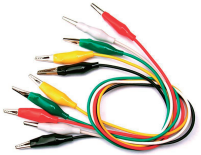


15. Jumper wire

It works like a bridge to connect each component.

16. Alligator Cable

Connect different components by the two ends of the clips makes them connect together.

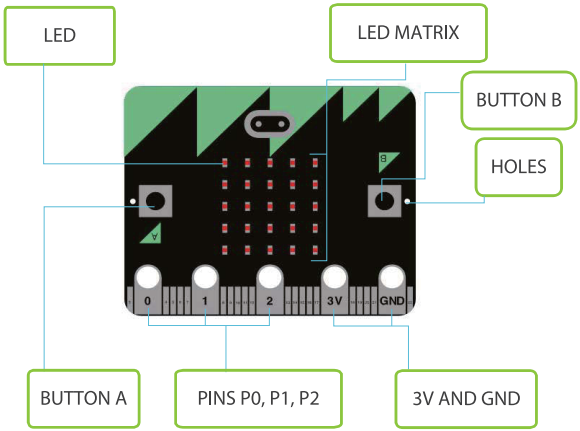


Parts List Introduction

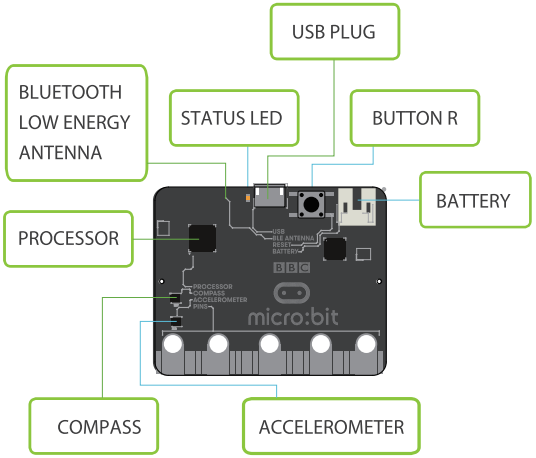
About Micro: bit

Micro:bit is a tiny programmable computer, designed to make learning and teaching easy and fun!
It' s widely used in schools around the world, and can be easily coded from any web browser in Blocks, JavaScript, Python, Scratch and more.

Front of board



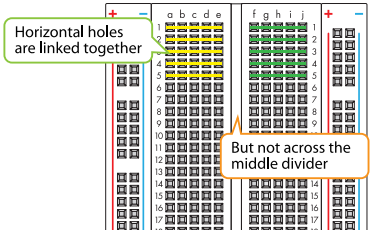
Back of board

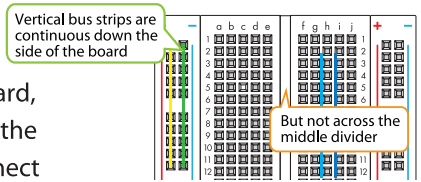


For more details, you can visit the official micro:bit page: **http://microbit.org/**

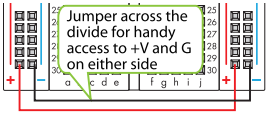
About the breadboard

Breadboard is one of the elemental components when you learn how to build an electric circuit. Basically, there are three areas functionally on a breadboard, power rails near the long sides, wiring rails (holes from a to j rows) and the middle groove. And remember that the inside of the breadboard (under the holes) is made up of sets of five metal clips.





The power rails are isolated to the right and left side of the breadboard, which allows you manage two different power supplies or voltages. the voltage and ground are handy on both side of the center if you connect left and right side of the board.



Lessons

There’ re 10 practical electronic projects in the lessons. Pupils can understand the electronics and obtain programming skills by hands on learning. After you complete these 10 lessons, you can know how different modules work and build your own projects.

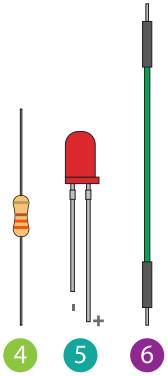
We use JavaScript blocks to programming all the lessons. You can find the editor on

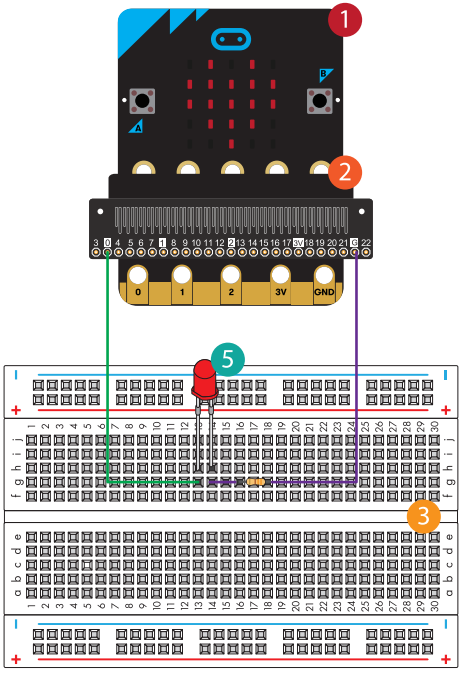
https://makecode.microbit.org/

1. Light the LED

Required parts

1. 1 x Micro: bit
2. 1 x Micro: bit GPIO extension board
3. 1 x Breadboard
4. 1 x 330 Ω resistor
5. 1 x LED
6. Jumper wires





Description

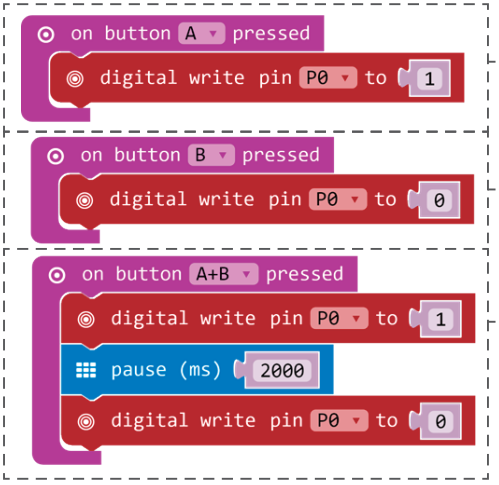
Light the LED is a simple experiment, we use the button A and B on the micro: bit as a switch.

Press button A, the LED will turn on. Press button B it will turn off.

The LED will turn on for 2 seconds and then turn off when button A and B are pressed together.

Remember to add a 330Ω resistor between the LED and GND to avoid excessive current which might burn the LED.

Programming



1. "*On button A pressed*" , it will activate "digital write *pin P0 to 1*" , the LED will turn on.

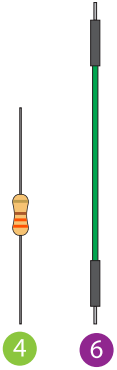
2. "*On button B pressed*" , "digital write *pin P0 to 0*" , the LED will turn off.

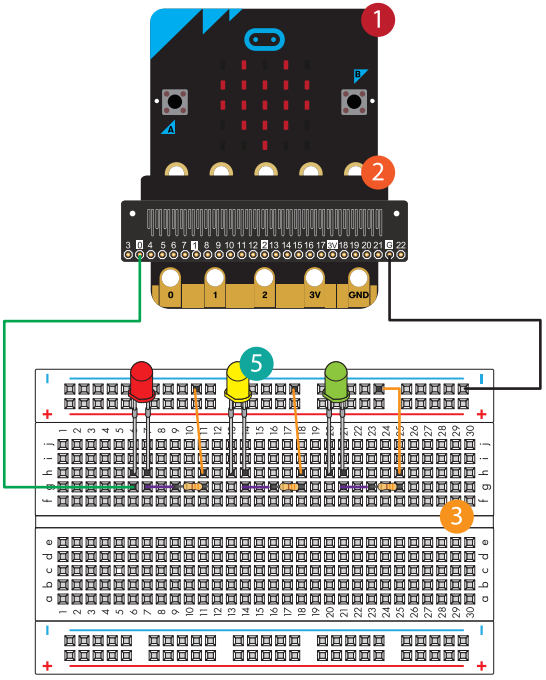
3. "*On button A + B pressed*" , "digital write *pin P0 to 1*" (turn on), "*pause 2 seconds*" , then "digital write pin P0 to 0" (turn off).

2. Traffic light

Required parts

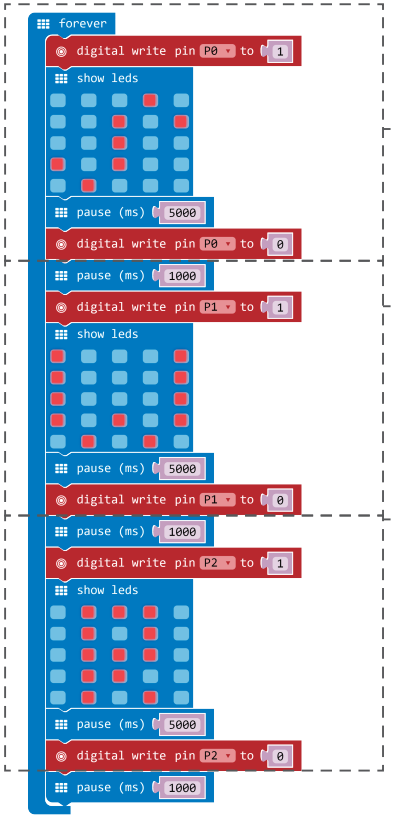
1. 1 x Micro: bit
2. 1 x Micro: bit GPIO extension board
3. 1 x Breadboard
4. 3 x 330 Ω resistor
5. 3 x LED (red, green, yellow)
6. Jumper wires





Description

Following the traffic rules is what we need to do, this lesson simulates the traffic light. when light the red LED, the 25 individually programmable LEDs shows “S” , yellow LED shows “W” , green LED shows “R” .



Programming

1. "***Digital write pin P0 to 1***" , will light the red LED. "***show leds***" , the LED screen on the micro: bit will show "S" , means stop. "pause 5 seconds" , then it will turn off.

2. "***Pause 1 second***" , then "***digital write pin P1 to 1***" (turn on the yellow LED). "***show leds***" , the LED screen on the micro: bit will show "W" , means wait. "***pause 5 seconds***" , then it will turn off.

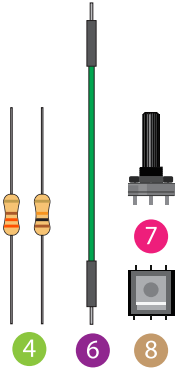
3. "***Pause 1 second***" , then "***digital write pin P2 to 1***" (turn on the green LED). "***show leds***" , the LED screen on the micro: bit will show "R" , means run. "***pause 5 seconds***" , then it will turn off.

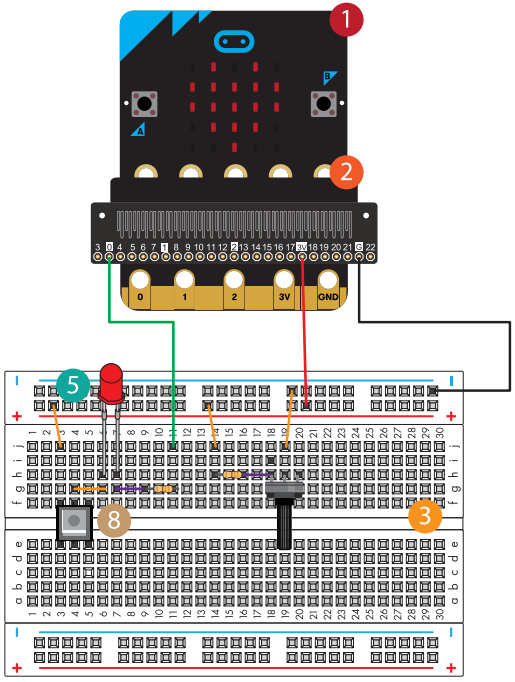
"forever" block makes the whole process in a loop.

3. Control the light switch and brightness

Required parts

1. 1 x Micro: bit
2. 1 x Micro: bit GPIO extension board
3. 1 x Breadboard
4. 1 x 330 Ω resistor; 1 x 10kΩ resistor
5. 1 x LED
6. Jumper wires
7. 1 x Potentiometer
8. 1 x Self-lock switch

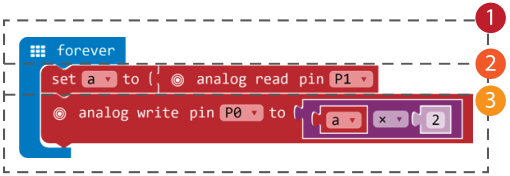




Description

Press the button, the light turn on. If the brightness is high, just adjust the potentiometer.

In this lesson, the value of the potentiometer is 10kΩ, so the resistor between the “+” and potentiometer is 10kΩ.



Programming

1. "***forever***" block makes the whole process in a loop.

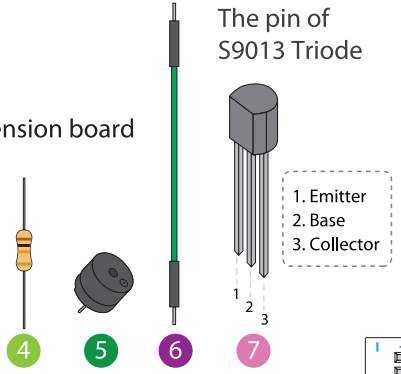
2. "***set a to analog read pin P1***" , read the analog value of the voltage of the potentiometer through the P1 pin of the micro: bit.

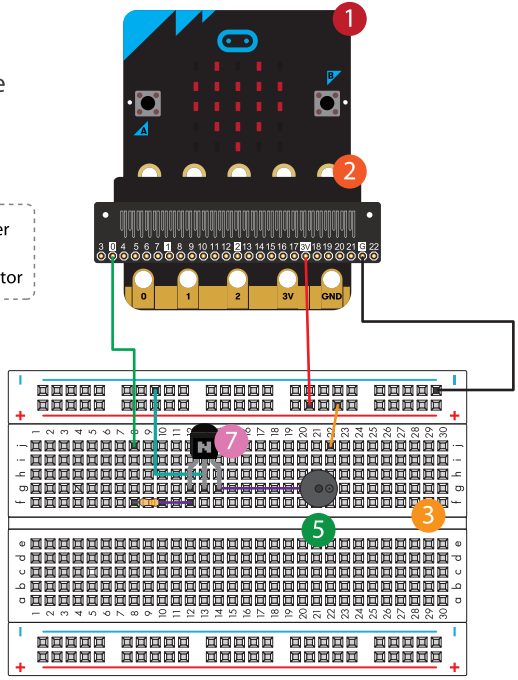
3. "***analog write pin P0 to a x 2***" , then give the analog value to LED through the P0 pin of micro: bit. So that we can adjust the different resistance value of the potentiometer to change the different voltages of the LED in order to control the LED brightness.

4. Let the buzzer sound

Required parts

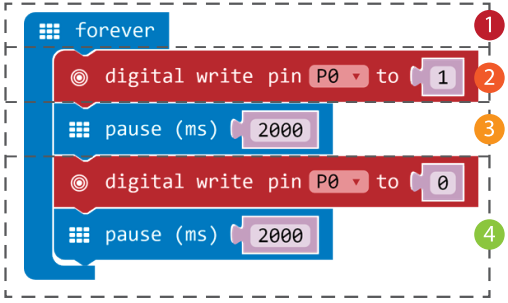
1. 1 x Micro: bit
2. 1 x Micro: bit GPIO extension board
3. 1 x Breadboard
4. 1 x 10kΩ resistor
5. 1 x Buzzer
6. Jumper wires
7. 1 x Triode





Description

The Triode is used to increase the current to drive the buzzer, you can also try to play the music with Micro: bit and buzzer. In order to drive the buzzer, we need to connect a triode to enlarge the current, see the circuit image below.



Programming

1. "*forever*" means the buzzer won't stop working until we turn off the power.

2. "*digital write pin P0 to 1*", give a signal to the buzzer and let it sound.

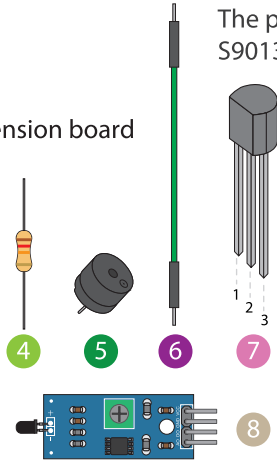
3. "*pause 2 seconds*", will sound for 2 seconds.

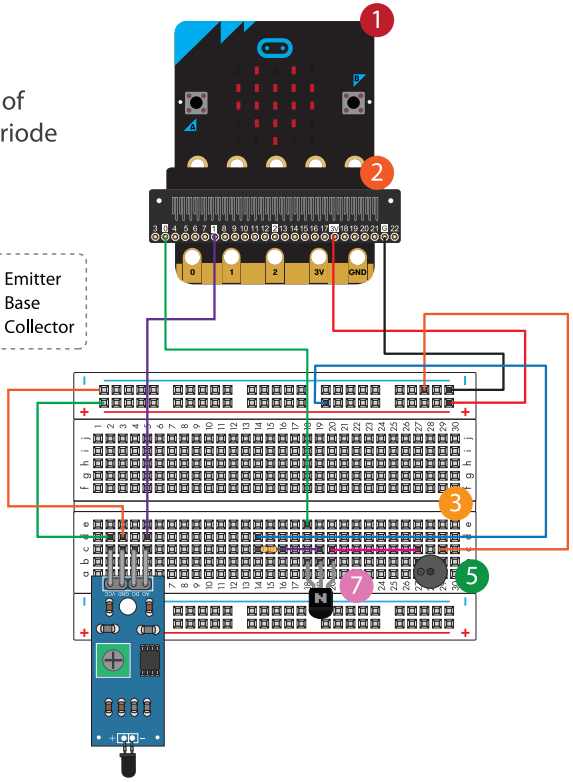
4. "*digital write pin P0 to 0*", it will stop.

5. Fire alarm

Required parts

1. 1 x Micro: bit
2. 1 x Micro: bit GPIO extension board
3. 1 x Breadboard
4. 1 x 1kΩ resistor
5. 1 x Buzzer
6. Jumper wires
7. 1 x Triode
8. 1 x Flame sensor

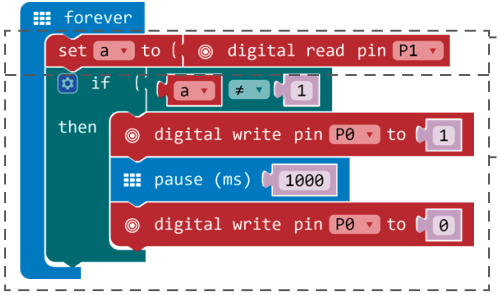




Description

We use a flame sensor to detect if there were any fire, if it was, then the buzzer will alarm.

Programming



1. "*set a to digital read pin P1*" read the digital voltage of flame sensor through the pin P1 of micro: bit.

2. The second block is a logic process, "*if a ≠ 1*" , means the flame sensor detects the fire, then "*digital write pin P0 to 1*" , the buzzer will alarm until "*digital write pin P0 to 0*" , the continued time is 1second.

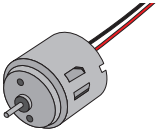
"forever" block makes the whole process in a loop.

Try to think what we should to do to put out the fire after hearing the alarm?

6. Self-lock switch to control a motor

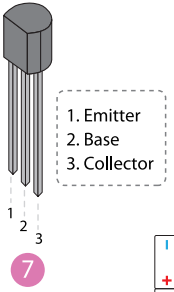
Required parts

1. 1 x Micro: bit
2. 1 x Micro: bit GPIO extension board
3. 1 x Breadboard
4. 1 x 1kΩ resistor
5. 1 x Motor (with fan)
6. Jumper wires
7. 1 x Triode
8. 1 x Self-lock switch

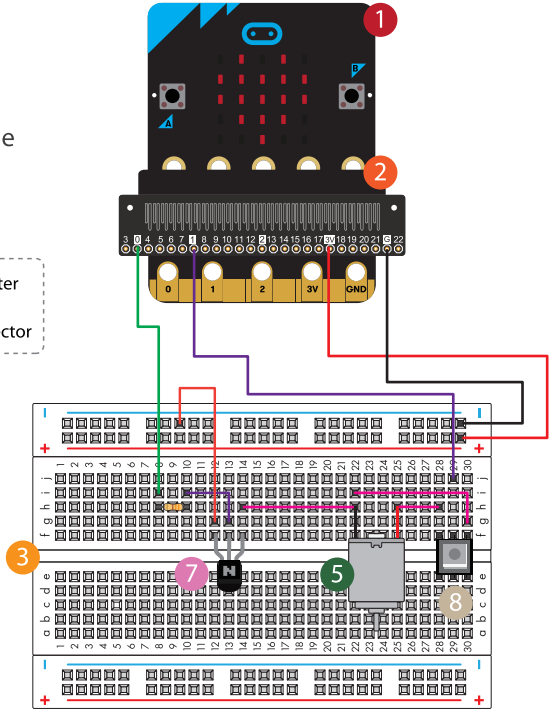


5

The pin of
S9013 Triode



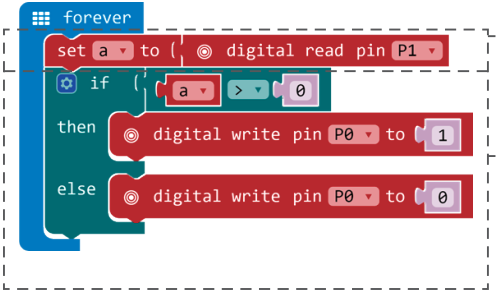
7



Description

Use a switch to control a motor start and stop.

Programming



1. "*set a to digital read pin P1*" read the digital voltage of self-lock switch.

2. The second part uses logic block, "*if a > 0*" , the switch is clocked, then "then *digital write pin P0 to 1*" , the motor start working. "else *digital write pin P0 to 0*" , it will stop.

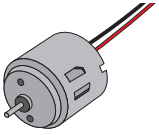
"forever" block makes the whole process in a loop.

Does the motor drive the fan give you inspiration on the fire extinguishing?

7. Fire extinguishing

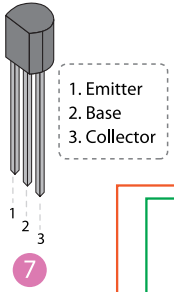
Required parts

1. 1 x Micro: bit
2. 1 x Micro: bit GPIO extension board
3. 1 x Breadboard
4. 2 x 1kΩ resistor
5. 1 x Motor (with fan)
6. Jumper wires
7. 2 x Triode
8. 1 x Buzzer
9. 1 x Flame sensor

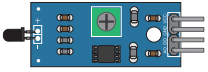


5

The pin of S9013 Triode



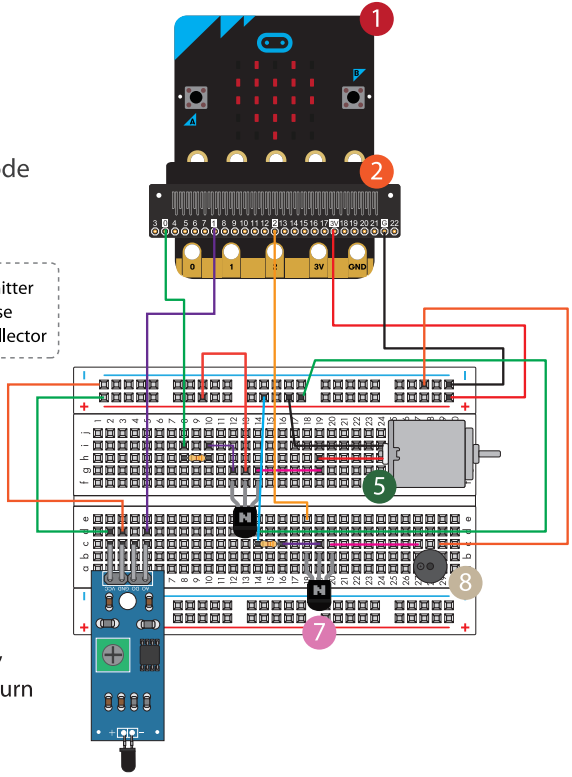
7

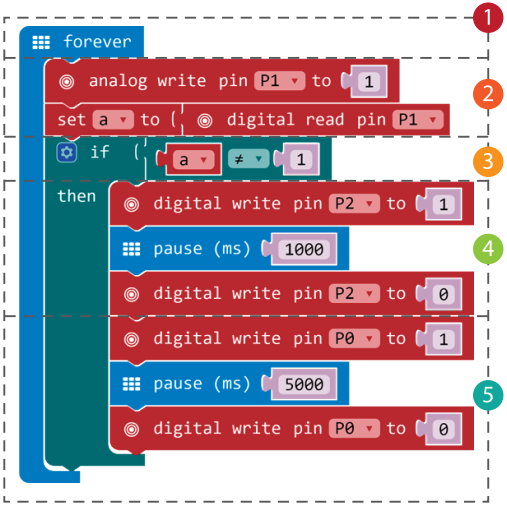


9

Description

We use a flame sensor to detect if there were any fire, if it was, then the buzzer will alarm. After 1s, the motor with a fan will turn on to extinguish the fire.





Programming

1. "***forever***" block makes the whole process in a loop.

2. "***set a to digital read pin P1***" read the digital voltage of flame sensor through the pin P1 of micro: bit.

3. The second part is a logic process, "***if a ≠ 1***" , means the flame sensor detects the fire.

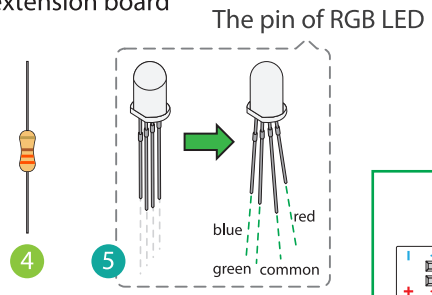
4. Then "***digital write pin P2 to 1***" , the buzzer will alarm until "***digital write pin P2 to 0***" , the continued time is 1 second

5. The fan starts immediately after the buzzer stops ringing (***digital write pin P0 to 1***), after 5 seconds (***pause (ms) 5000***), the fan will stop

8. Light the RGB LED

Required parts

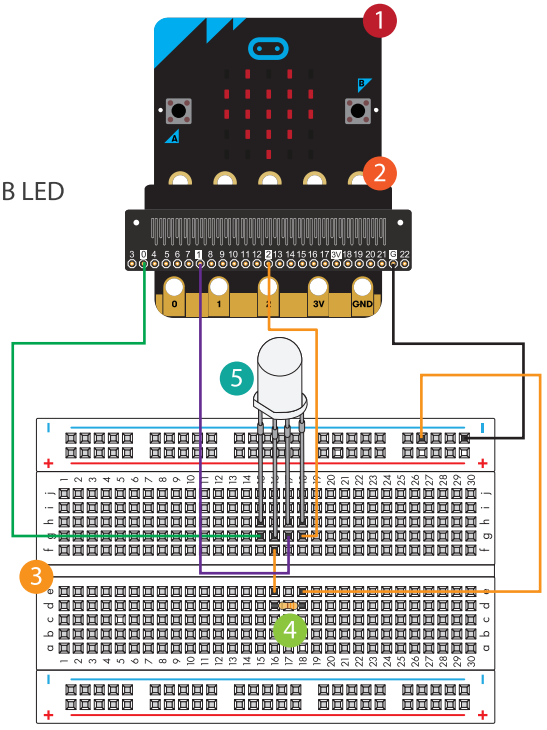
1. 1 x Micro: bit
2. 1 x Micro: bit GPIO extension board
3. 1 x Breadboard
4. 1 x 330Ω resistor
5. 1 x RGB LED
6. Jumper wires



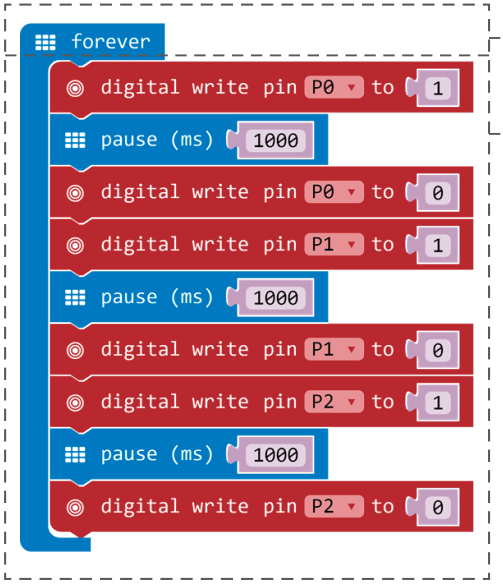
Description

Use the RGB LED to built a colorful world, just digital write to its three pins, and the pause time can be decided yourself.

In this lesson, we use the 4-pin RGB LED, use the multimeter to detect the common anode, the circuit of this lesson is as below, just light your RGB follow the circuit!



Programming



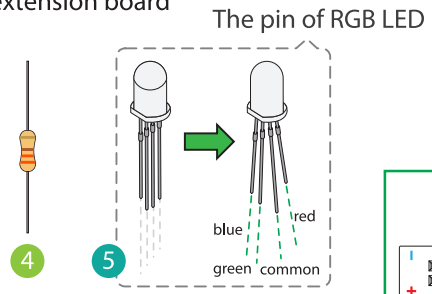
1. "*forever*" block makes the whole process in a loop.

2. The first block "*digital write pin P0 to 1*" , the RGB LED will show green, after 1 second ("*pause (ms) 1000*" , can be adjusted), it will show blue. Then it will show red after 1 second.

9. Shaking to light the RGB LED

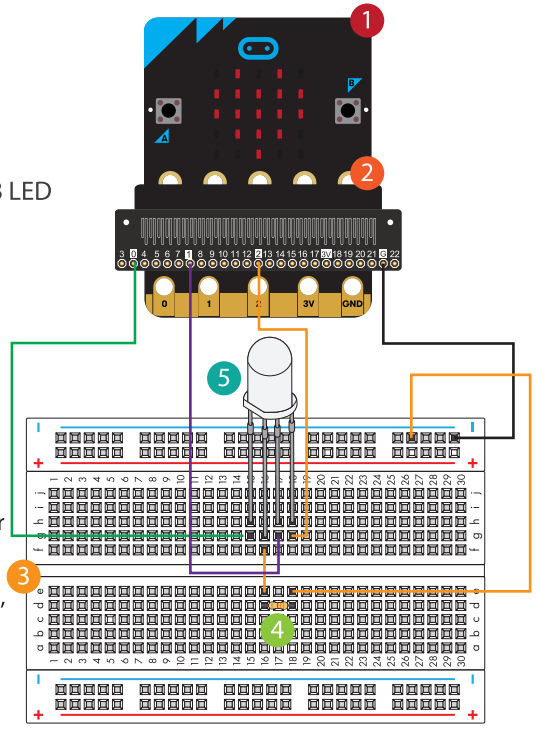
Required parts

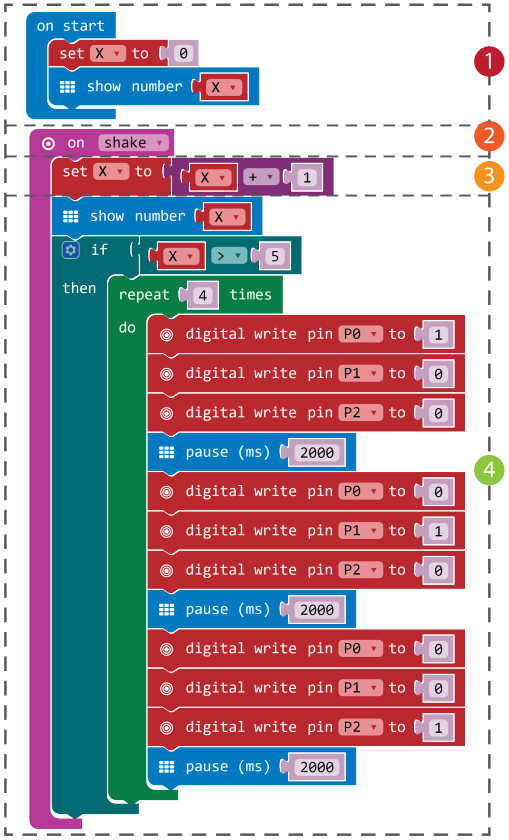
1. 1 x Micro: bit
2. 1 x Micro: bit GPIO extension board
3. 1 x Breadboard
4. 1 x 330Ω resistor
5. 1 x RGB LED
6. Jumper wires



Description

Micro: bit itself has an accelerometer, what is it? An accelerometer measures the acceleration of your micro: bit, this component senses when the micro: bit moves. It can also detect other actions, e.g. shake, tilt, and free-fall. This lesson we light the RGB LED by shaking, when the number on micro: bit LED screen is up to 6 or larger than 6, the RGB LED will cycle light four times.





Programming

1. The program of this lesson consists of two parts, the first part will show the initial acceleration value on the LED screen of micro: bit ("***show number X***"). We set the initial acceleration to zero ("***set X to 0***").

2. The second part "***on shake***" , the system monitors the acceleration values on the three dimensions of the accelerometer in real time.

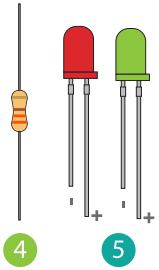
3. "***set X to X+1***" , means the value on the LED screen will increase 1 when detect shaking.

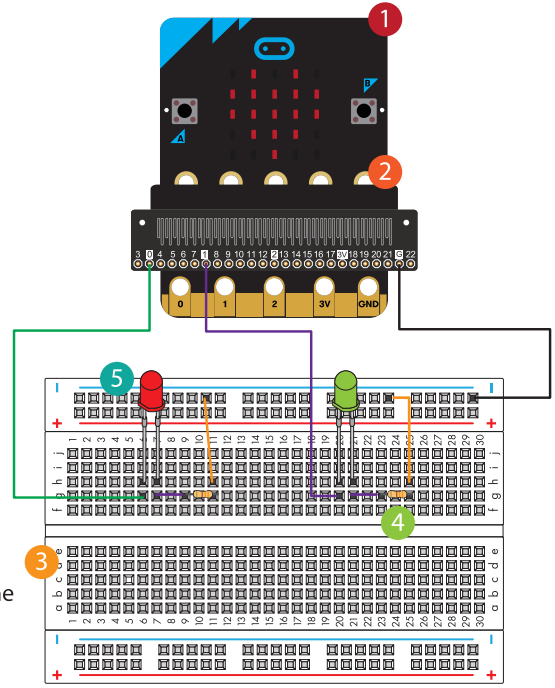
4. We add a logic block to light the RGB LED, "***if X > 5***" , the RGB LED will show green, blue, red in turn, and will loop four times.

10. Electronic thermometer

Required parts

1. 1 x Micro: bit
2. 1 x Micro: bit GPIO extension board
3. 1 x Breadboard
4. 2 x 330Ω resistor
5. 2 x LED (red, green)
6. Jumper wires

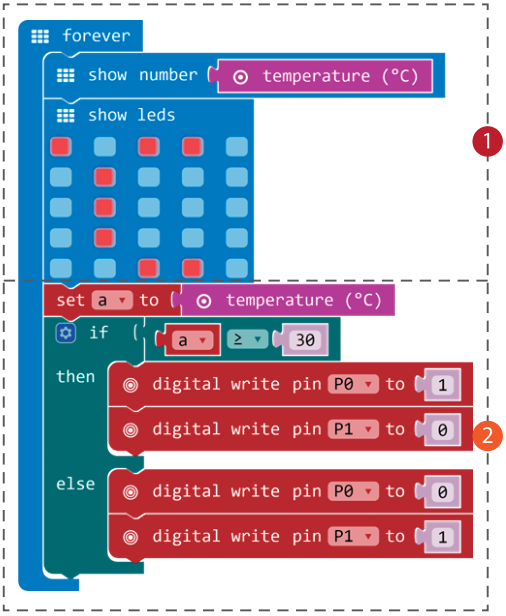




Description

The micro: bit itself has a temperature sensor, this sensor allows the micro: bit to detect the current ambient temperature, in degrees Celsius.

In this lesson, we use two LEDs, one is red, another is green. If the temperature is up 30℃, the red led will light on. Otherwise, the green led will light. You can see the temperature value through the LED screen on micro: bit.



Programming

1. "***show number temperature (℃)***" , the LED screen of micro: bit will show the current temperature.

2. The logic block "***if a ≥ 30 ℃***" , the red LED will light, otherwise, the green LED will light.

Does this DIY temperature reminder attract you ? How about connecting a buzzer? Let it sound when the temperature when the temperature is up 30℃? Just try it!

ELECROW

MAKE YOUR MAKING EASIER

ELECROW

+86 0755-23204330

www.elecrow.com

info@elecrow.com

8th Floor, F-building, Fusen industry park, Gushu Town, Bao'an District, Shenzhen, China.