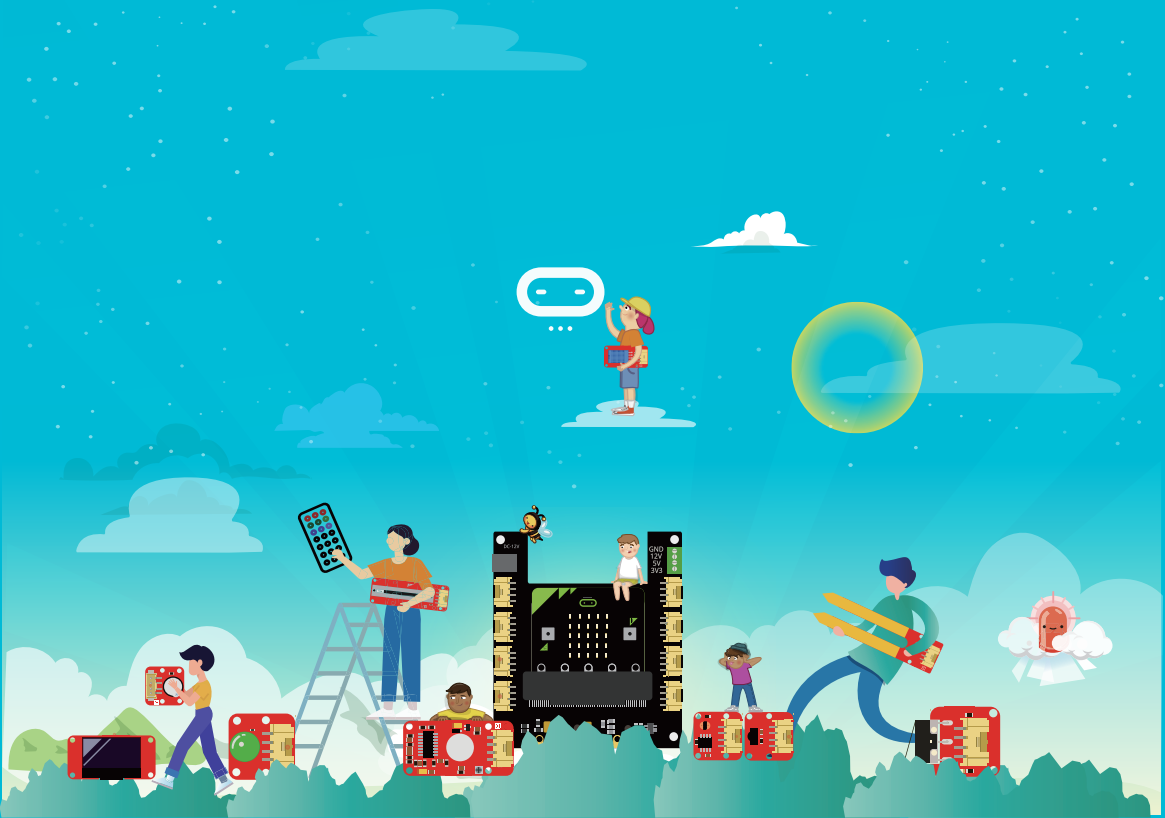


# Crowtail Starter Kit for Micro:bit

---

User Guide

V2.0 | 2020.02





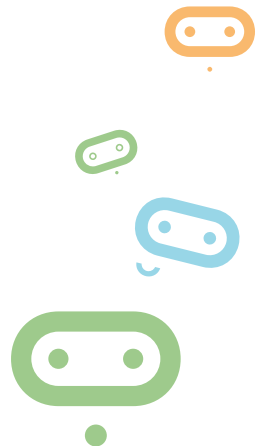
# Table of Contents

<b>Instruction</b>	1
<b>Get to know micro:bit</b>	2-6
<b>Modules list</b>	7
<b>What is Crowtail</b>	7-8
<b>Exciting experiments</b>	8
• Experiment 1 Blinking an LED	8-9
• Experiment 2 Switch light	10-11
• Experiment 3 Micro:bit singer	11-13
• Experiment 4 Do not touch	13-14
• Experiment 5 Collision check	15-16
• Experiment 6 Intelligent corridor lights	17-18
• Experiment 7 Plant soil moisture detection	19-20
• Experiment 8 Intrusion alert	21-22
• Experiment 9 Make an adjustable table lamp	23-24
• Experiment 10 Automatic control	25-26
• Experiment 11 Automatic door	27-28
• Experiment 12 Weather station	29-31
• Experiment 13 Ultrasonic ranging smart door	31-33
• Experiment 14 Make an accurate clock	33-35
• Experiment 15 Distance display	35-38
• Experiment 16 Wireless communication	38-40
• Experiment 17 Remote control your micro:bit	40-42
• Experiment 18 Smart music box	43-44
• Experiment 19 Remote control fan	44-45
• Experiment 20 Car remote control	46-48
• Experiment 21 Pick up beans	48-51
<b>Keep exploring</b>	51

# Instruction



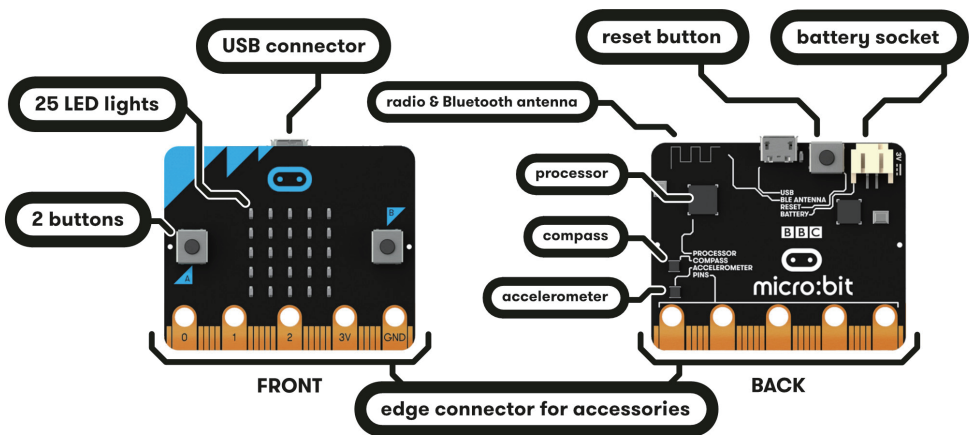
The Elecrow Starter kit for Micro:bit is designed for people who are at the doorstep of learning electric modules and programming knowledge and bring numerous creative possibilities for you to dig out and help you enter a wonderful electronic world. This kit includes more than 20 modules like the buzzer, light sensor, servo, ultrasonic ranging sensor, and OLED, etc. The kit also provides you plenty of fun experiments so you can learn how the electric modules work and use it to build innovative works and plan out unique projects. In addition, the kit also focuses on stimulating children's creativity so that they can continue to move forward rather than stagnate after completing their work. We never allow complicated jumpers to prevent children from learning about the electronic world and programming hobbies and interests, and we vowed to change it. Therefore, we have equipped this starter kit for Micro:bit with our plug-and-play crowtail series of electronic modules to make you realize the electronic works in the mind quickly and easily.



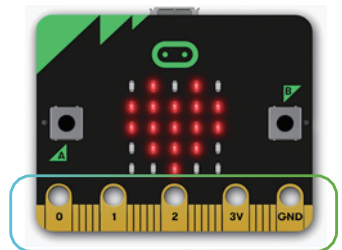
# Get to know micro:bit

Micro:bit is a pocket-sized computer that lets you get creative with digital technology. You can code, customize and control your micro:bit from anywhere! You can use your micro:bit for all sorts of unique creations, from robots to musical instruments, the possibilities are infinite. The micro:bit compatible with a number of online code editors across a number of different languages, such as [Blocks](#), [JavaScript](#), [Python](#), [Scratch](#), and etc. This guide will focus on Makecode, a block or JavaScript-based environment developed by Microsoft.

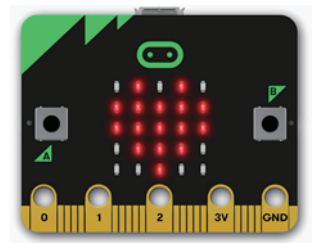
## Components on the board



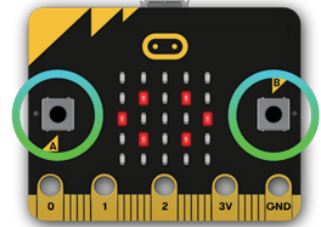
- **Edge "Pins":** The [gold tabs](#) at the bottom of the board are for hooking up external components. The tabs with larger holes can be easily used with alligator clips to prototype quickly!



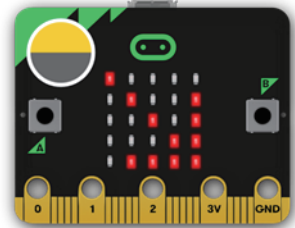
- **LED Array:** The micro:bit has [25 individually-programmable LEDs](#), these 5x5 red LEDs form a very small section of a screen, and display information like words, numbers, and pictures through a combination of ON and OFF LEDs.



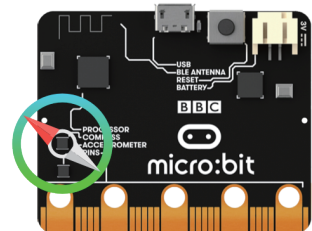
- **A/B Buttons:** There are **two buttons** on the front of the micro:bit - button A and button B, you can detect when these buttons are pressed, allowing you to trigger code on the device. You can detect events such as a single press, a double press, and a long press by programming.



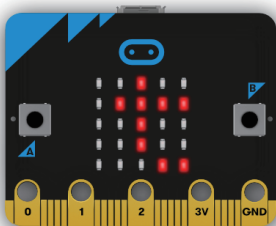
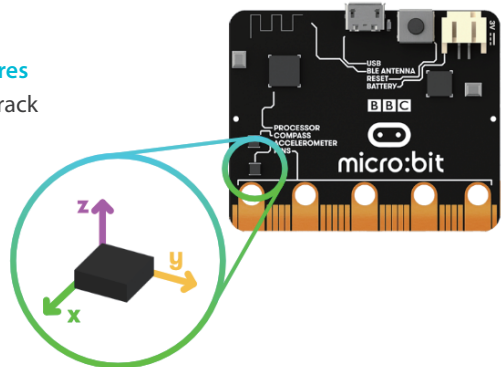
- **Light Sensor:** The LED screen works as a basic **light sensor** by reversing the LEDs of the screen to become an input which allows you to detect the ambient light.



- **Compass:** The **compass** detects the earth's magnetic field, allowing you to detect which direction the micro:bit is facing. The compass has to be calibrated before it can be used. Calibrating the compass ensures the compass results are accurate.

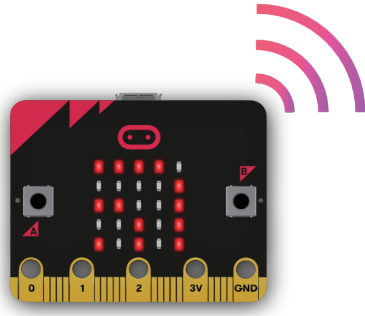


- **Accelerometer:** The accelerometer **measures the acceleration of your micro:bit**, it can track motions like tilt, free fall and shake.

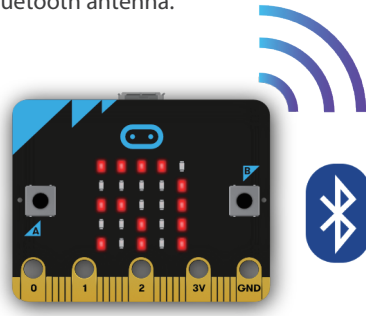


- **Temperature Sensor:** The Microcontroller **doubles as a temperature sensor!** The temperature sensor transforms the temperature in the ambient environment into digital signals that can be read by the micro:bit for calculating the current temperature of the device.

- **Radio:** The radio feature allows you to **communicate wirelessly between two or more micro:bits**.



- **Bluetooth:** The Bluetooth Low Energy antenna allows the micro:bit to **send and receive Bluetooth signals, you can wirelessly communicate between micro:bit and phones, PCs, or tablets**. You need to pair your micro:bit with another device before using the Bluetooth antenna.



## Quick start

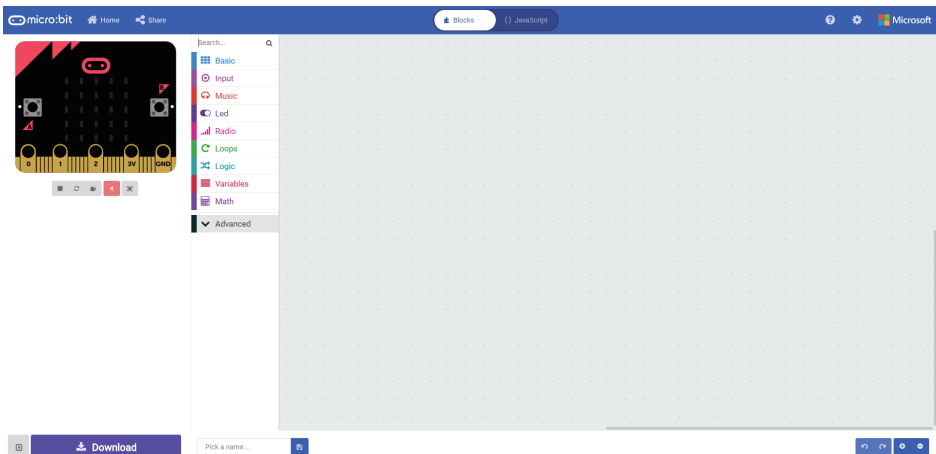
---

### ■ Step 1. Connect it

Connect the micro:bit with your computer via a USB cable, then your micro:bit will show up on your computer as a drive called "MICROBIT".

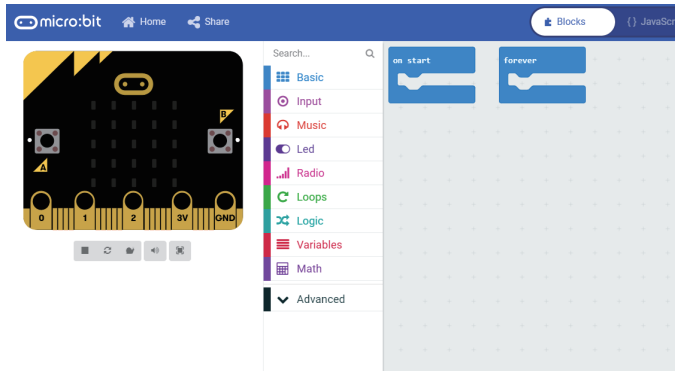
### ■ Step 2. Program it

Write your micro:bit code by using the official online editor <https://makecode.microbit.org/>

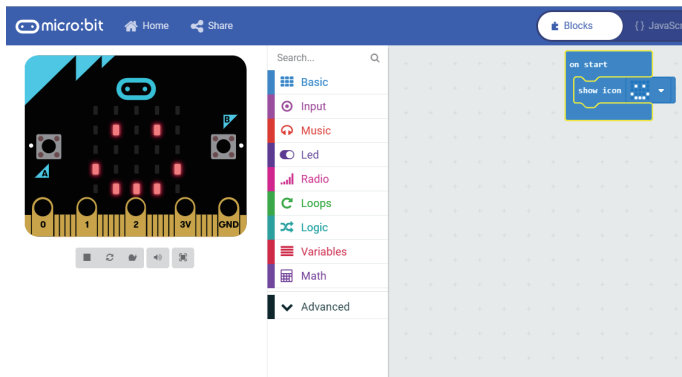


The left part is a simulator, which will simulate the work of micro:bit after finishing the program. The middle part is our command area, there are various program blocks we need. The rightmost area is the script area, you can drag and drop instructions into this area to form your program. Drag and drop some blocks and try your program out in the Simulator in the Makecode Editor, the example below shows you how to program a smiley face.

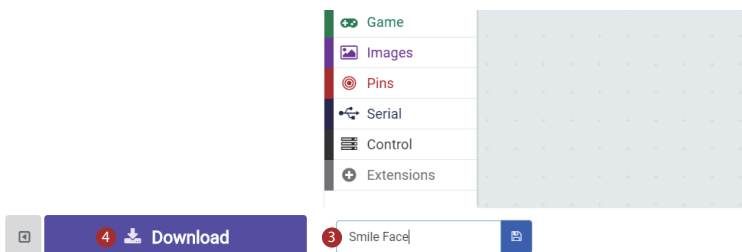
- **Step 2-1.** Select **“on start”** block in the command area, and drag it out to place it in the script area.



- **Step 2-2.** Drag out the **“show icon”** block and place it inside the **“on start”** block.



- **Step 2-3.** Pick a name for your program, We'll name this project as **“Smile Face”**.





- **Step 2-4. Download** the code

Click the **Download button** in the editor, And then it will download a “hex” file that your micro:bit can read. Once the hex file has downloaded, copy it into your micro:bit like copying a file into a USB drive.

- **Step 3. Play**

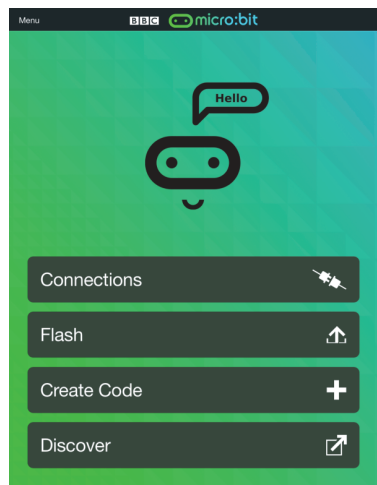
The micro:bit will pause and the yellow LED on the back of the micro:bit will **blink** while your code is programmed, the code will run automatically once it's finished. Then look at your micro:bit, the screen will show **a smiley face**.

**Note:** The micro:bit drive will automatically eject and come back each time you program it, but your hex file will be gone. It can only receive hex files and won't store anything else.

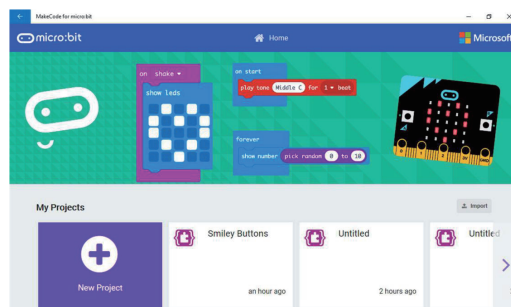
## APP for Micro:bit

There is no wire needed if you use Bluetooth to send the code to your micro:bit from your mobile device. What you need to do is to make sure your micro:bit is powered up and within easy reach of the phone or tablet while running the App. There are two types of APPs available for micro:bit which are based on Microsoft Makecode languages and Javascript: respectively micro:bit companion app and Windows 10 APP.

The micro:bit app is the essential companion to the BBC micro:bit, available for iOS and Android. Download the official micro:bit app for Android or iOS from the app store, the app contains handy instructions on how to pair your micro:bit with the app.



The Windows 10 APP has a few extra features over the editor at [makecode.microbit.org](https://makecode.microbit.org) that will work on micro:bits with up-to-date interface firmware: automatically program your micro:bit over USB, without needing to drag-and-drop the file onto the micro:bit drive; directly read serial data from your micro:bit for data logging and other fun experiments.



# Modules list

Crowtail-Base Shield for Micro:bit x1	Crowtail-Ultrasonic Ranging Sensor x1
Crowtail-LED (Yellow) x1	Crowtail-Linear Potentiometer x1
Crowtail-LED (Green) x1	Crowtail-Temperature&Humidity Sensor x1
Crowtail-Buzzer x1	Crowtail-Relay x1
Crowtail-Switch x1	Crowtail-9G Servo x1
Crowtail-Touch Sensor x1	Crowtail-RTC x1
Crowtail-Collision Sensor x1	Crowtail-OLED x1
Crowtail-Light Sensor x1	Crowtail-IR Receiver x1
Crowtail-Moisture Sensor x1	Infrared Remote Control x1
Crowtail-PIR Motion Sensor x1	DC motor x1

## What is Crowtail

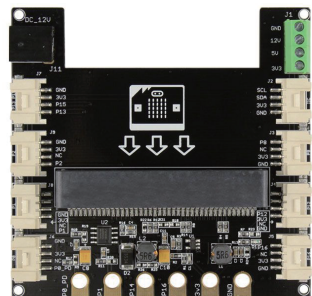
Welcome to the world of Crowtail! Crowtail is a modular, ready-to-use toolset, it takes a building block approach to assemble electronics. It simplifies and condenses the learning process significantly. In our Crowtail repository, there are over 150 Crowtail modules and Crowtail shields!

The Crowtail products are basic-functional modules that consist of a Base Shield and various modules with standardized connectors, each Crowtail module has its specific functions, such as light sensing and temperature sensing. It will satisfy all you need for your project!

Crowtail is a series of products that we made to solve the messy jumper when connecting electronic circuits. It consists of a Base Shield and some basic Crowtail modules, which help you create small, simple, and easy-to-assemble circuits. In other words, when you use Crowtail, your electronic project will not be messy wiring. Instead, it will be simple and easy to manage an electronic project!

## Crowtail-Base shield for Micro:bit

The Crowtail-Base shield for Micro:bit is the core part of this kit, it acts as a bridge between Micro:bit board and Crowtail modules. You can plug and then play with Crowtail modules and micro:bit through this base shield, which allows you to create more interesting and innovative works with micro:bit. This shield integrated 6 crowtail connectors and 2 crowtail UART/I2C connectors onboard. So you can use plenty of crowtail sensors. All you need to know is that there is no need for any soldering or jump wire about Crowtail.



In addition, there is a power port, from which you can draw 3.3V/5V/12V three voltages! This means that when you use a module such as a motor, you no longer need an external drive!

**Note:** When you connect a high-power device, such as a motor, you need to connect a 12V power supply to the board.

## Crowtail – Modules

We've made more than 100 kinds of electronic modules into Crowtail modules. They include a variety of sensors, displays, inputs and outputs modules, communication types include I2C, UART, digital or analog, which aim to provide more options to fully meet all needs for your electronic projects! All modules can be used by simply connecting them to the Crowtail- Base shield for Micro:bit with Crowtail cable, which is a huge improvement over the previously troublesome jumper connections.

# Exciting experiments

## Experiment 1 – Blinking an LED

### Instruction

Electric lights are ubiquitous in our daily life which we can't even leave. It was the invention of the electric light that marked the human society's entry into the electric era, and made people welcome the light in the dark night. Here, we will learn how to use the LED and control its on and off.

### Target

- Learn how the LED work and use it to make a blinking light.
- Learn how to run the code forever.
- Learn how to output logic high and low signals.
- Learn how to pause the code.

### Required Parts

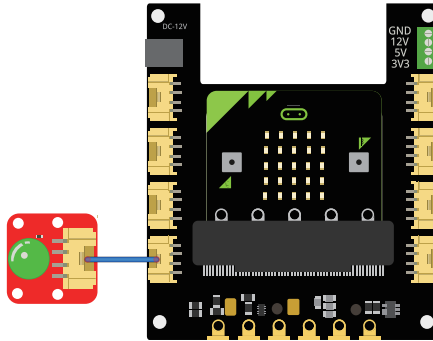
- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-LED (Green) x1
- Crowtail-Cable x1
- USB cable x1

### Hardware learning and connection

LED is the best choice to help you learn I/O pins. What you need to do is connecting the LED module to the port of the base shield, then download the program to the Micro:bit. Besides the very basic usage, you can make the LED blink with the frequency you want, or adjust the brightness with PWM. Actually, LED is the most popular used for human interface.

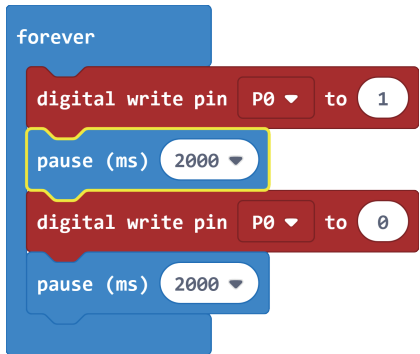


Connect Crowtail-LED to P0 port of Crowtail-Base shield for Micro:bit. The hardware connections are as follows:



## Programming and note

- **Forever:** The **forever** block is a block that loops any other command blocks inserted into it over and over again. It starts from the top and executes your code in order by working its way to the bottom and then starts from the top again.
- **Digital write:** The **digital write** block enables you to turn a pin on or off. There is a dropdown option for you to choose which pin you want to control, and it accepts a variable as the pins state. You could use **1** as on and **0** as off. If you prefer, you can also use Boolean states of **true** and **false**, but we will use **0** and **1** as our standard throughout this guide.
- **Pause:** If you were to just turn pins on and off with the **digital write** block without a pause, the LED would blink really, really fast. The **pause** block enables you to slow the micro:bit down and lets you control the timing of things happening. It accepts a number or variable as the number of milliseconds you want the micro:bit to pause. Think of this block as a stoplight for your code!
- **Download the program to micro:bit to see what happens.**



## Result

You will see the LED blink on and off at 2 seconds interval. If it doesn't, make sure you have connected the LED to correctly port.



*Just like the lights at home, can we add a switch to control the lights on and off at any time?*

## Experiment 2 – Switch light

### Instruction

Congratulations, you have mastered the method of controlling the LED on and off. But just like the lights at home, can we add a switch to control the lights on and off at any time? Of course, in this experiment, we will use our switch module to control the lights anytime and anywhere.

### Target

- Learn how the switch work and use it to make a switch light with LED.
- Learn how to use the if/else logic statement.
- Learn how to read the state from the digital module.

### Required Parts

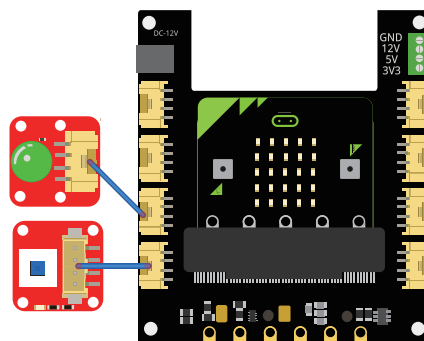
- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-LED (Green) x1
- Crowtail-Switch x1
- Crowtail-Cable x2
- USB cable x1

### Hardware learning and connection

The Crowtail- Switch is a Latching switch. When the switch is pressed for the first time, the switch maintains current regulation and the button outputs a HIGH signal in the self-locking state. When the switch is pressed for a second time, the switch button pops up and the switch turns off and then outputs a LOW signal. In fact, it is very similar to the button, except that the switch has a self-locking function so that it can output logic high level signal without pressing it all the time.



Connect Crowtail-Switch and Crowtail-LED to P0 and P1 ports of Crowtail-Base shield for Micro:bit. The hardware connections are as follows:



## Programming and note

```
forever
  if digital read pin P0 = 1 then
    digital write pin P1 to 1
  else
    digital write pin P1 to 0
```

- **Digital Read:** Just like the way **digital write** block turns a pin on (1) or off (0), the **digital read** block check at the state of a pin, which is either HIGH (1) or LOW (0). In this case, we can detect if the switch is pressed or not by using this block.
- **If/else:** The **if/else** block is a logical structure. If the logical statement that is attached to it (switchState=1) is true, then it will execute the code blocks inside of the **if**. If that statement is false, it will execute the **else** blocks. In this case, if the statement is true(the value is digital read from P0 port equal to 1), then turn on the LED on pin 1; else, turn off the LED on P1.
- **Download the program to micro:bit to see what happens.**

### Result

When you press the switch, the LED lights up, and when you press the switch a second time, the LED turns off.



*Do you have this experience? whenever you go up the stairs at night, the lights will light up in front of you. Can we make such a lamp?*

## Experiment 3 – Micro:bit singer

### Instruction

Do you like music? Have you ever dreamed of becoming a singer? The power of music is really great, it can make people happy physically and mentally. In this experiment, we will make the micro:bit to be a “singer”, let’s see how to do it!

## Target

- Learn how the buzzer work and use it to play songs.
- Learn how to run the code once at the beginning of the program.

## Required Parts

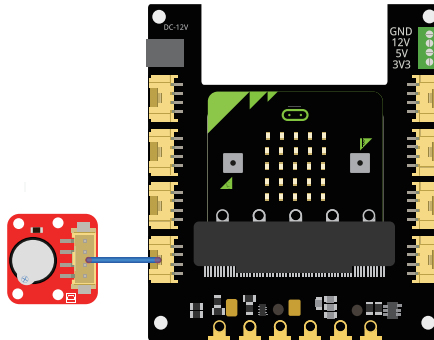
- Micro:bit x1
- Crowtail-Cable x1
- Crowtail-Base shield for Micro:bit x1
- USB cable x1
- Crowtail-Buzzer x1

## Hardware learning and connection

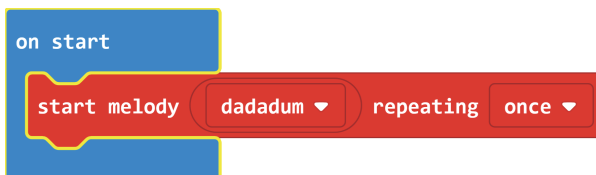
The buzzer module is used for making sound in your project. It makes a sound when activated by a logic HIGH signal. Connect the buzzer to any port of Crowtail- Base shield for Micro:bit, you can easily make it sound with setting the related port to logic HIGH. The buzzer module can also be connected to an analog pulse-width modulation(PWM) output to generate various tones, which means you can use it to create your own melody!



Connect Crowtail-Buzzer to P0 port of Crowtail-Base shield for Micro:bit. The hardware connections are as follows:



## Programming and note



- **Start melody repeating:** The **Start Melody Repeating** block eliminates all of the frustration from getting music from a microcontroller. It is as simple as selecting one of the many songs that were pre-programmed into MakeCode and repeating it as many times as possible! Note that: There's no other to run while the melody is playing. Which is called "blocking" code and must be Considered in your program.

- **On start:** Different from **forever** block that loops your code forever, The **On start** block is a block of code that only runs once at the very beginning of your program. In this case, we use it to set the melody to play once.
- **Download the program to micro:bit to see what happens.**

### Result

Once you upload the program to micro:bit, you can hear the buzzer playing the “dadadum” melody once and then stop.



*How to make a buzzer play multiple melodies and control the previous and next song by pressing a button?*

## Experiment 4 – Do not touch

### Instruction

There are so many dangerous areas that we can't touch with our hands directly, otherwise, we may be in danger. For example, we can't touch the power socket with our hands because the danger of electric shock! Therefore, this time we will make a touch alarm, put this alarm in a simulated danger area, and when someone touches it, make the buzzer shout loudly so that they will never touch this area again!

**Note:** Do not use it in a truly dangerous area, it can be dangerous!

### Target

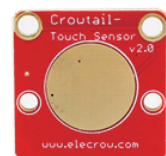
- Learn how the touch sensor work and use it to make a touch alarm with buzzer.

### Required Parts

- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-Touch Sensorx1
- Crowtail-Buzzer x1
- Crowtail-Cable x2
- USB cable x1

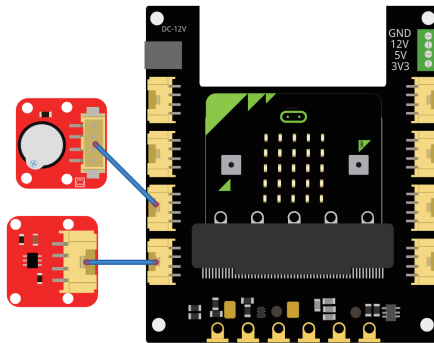
### Hardware learning and connection

The touch sensor can detect the human touch by sensing changes in capacitance. When it detects a touch, it outputs a HIGH logic level signal. Based on the touch IC TTP223-B, this module can detect human finger in 0~3mm, that is, you can place this sensor on a non-metallic surface such as the glass or paper, with a thickness of less than 3MM, this would be useful for applications that waterproof is needed, or you want to keep the buttons secret.





Connect Crowtail-Touch Sensor and Crowtail-Buzzer to P0 and P1 ports of Crowtail-Base shield for Micro:bit. The hardware connections are as follows:



## Programming and note


```
forever
  if digital read pin P0 = 1 then
    digital write pin P1 to 1
  else
    digital write pin P1 to 0
```

- **Digital Write:** Different from the last experiment, this time we will use **digital write** block to set the buzzer 1(on), which can make the buzzer make a big bee noise to prevent touching. The buzzer will stop sounding until you stop touching.
- **Download the program to micro:bit to see what happens.**

### Result

If you put your hand on the touch sensor, the buzzer will make a large beep to prevent touch. The buzzer will not stop sounding until you remove your hand from the touch sensor.



 By learning this experiment, we know the touch sensor can sense our hand. So, if we have many touch sensors and a buzzer, We can make a touch-type musical instrument that allows us to play own melody?

# Experiment 5 – Collision check

## Instruction

Do you know there is a collision sensor inside the car? It is used to detect collisions to make sure the car can open airbag immediately and alarming after the impact. In this experiment, let's make a collision detector to simply simulate the process of automobile collision detection.

## Target

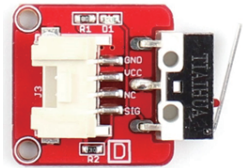
- Learn how the collision sensor work and use it to make a collision detector with buzzer.

## Required Parts

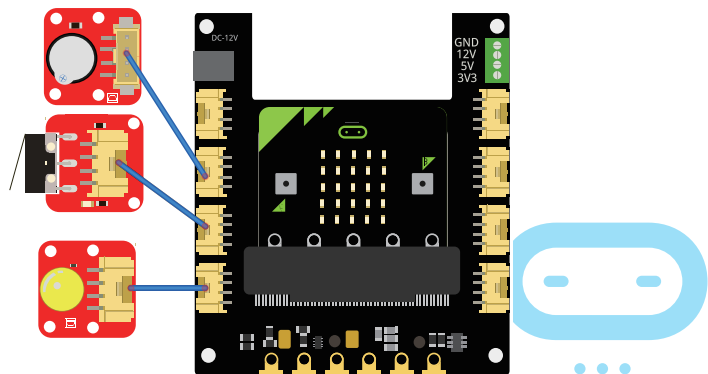
- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-Collision Sensor x1
- Crowtail-Buzzer x1
- Crowtail-LED(Yellow) x1
- Crowtail-Cable x3
- USB cable x1

## Hardware learning and connection

This is a robot model car crashes or collision switch sensor module, it usually outputs a logic HIGH signal, but when the sensor crash something such as the wall, the on-board switch will be pressed, and the module outputs a logic LOW signal. This module can be installed into any mobile platform to achieve collision detection function via 4 pin sensor cable and Micro:bit sensor expansion board connector.



Connect the Crowtail-Collision Sensor, Crowtail-LED and Crowtail-Buzzer to P1, P0 and P2 ports of Crowtail-Base shield for Micro:bit. The hardware connections are as follows:



## Programming and note

```
forever
  if digital read pin P1 = 0 then
    digital write pin P0 to 1
    digital write pin P2 to 1
  else
    digital write pin P0 to 0
    digital write pin P2 to 0
```

- **If/else:** The **if/else** block is a logical structure. If the logical statement that is attached to it (digital read P1 = 0) is true, then it will execute the code blocks inside of the if. If that statement is false, it will execute the **else** blocks. In this case, if the statement is true (the value is digital read from P1 port equal to 0, note that it will output a logic LOW signal when the collision sensor is crashed), then turn on the LED and buzzer on P0 and P2; else, turn off the LED and buzzer on P0 and P2.
- **Digital Write:** There are two modules we need to use the **digital write** block to set they output 1 (logic HIGH signal) or 0 (logic LOW signal). Here, we set the LED and buzzer output to 1 when the collision sensor detect a collision, otherwise, set it to 0.
- **Download the program to micro:bit to see what happens.**

### Result

When the on-board switch of the collision sensor is pressed, LED will light on and the buzzer will make a big beep noise to remind collision to occur, otherwise, LED will light off and buzzer stop work.



*The car can detect a collision, but how the car detects the obstacle to prevent a crash?*

# Experiment 6 – Intelligent corridor lights

## Instruction

You must have a lot of questions if pay attention to the various phenomena of daily life, such as corridor lights? How does it work? How did it automatically turn on the lights when we went up the stairs? With these questions in mind, let's start our sixth experiment: intelligent corridor lights.

## Target

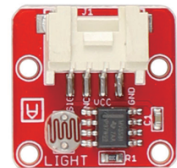
- Learn how the light sensor works and use it to make an intelligent corridor light with LED.
- Learn how to read the values from an analog module.

## Required Parts

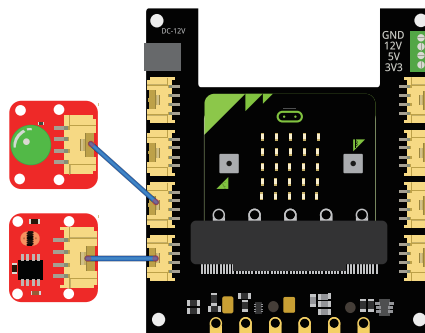
- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-Light Sensor x1
- Crowtail-LED(Green) x1
- Crowtail-Cable x2
- USB cable x1

## Hardware learning and connection

The Light sensor module uses the GL5516 photoresistor to detect the light intensity of the environment. The resistance of the sensor decreases when the light intensity of the environment increases. The chip LMV358 is used as a voltage follower to enable you to get accurate data. This module outputs an analog signal that shows the light intensity. It can be used in many occasions, such as Intelligent street light, intelligent corridor light, etc.



Connect Crowtail-Light Sensor and Crowtail-LED to P0 and P1 ports of Crowtail-Base shield for Micro:bit. The hardware connections are as follows:



## Programming and note

```
forever
  if < analog read pin P0 > < 300 > then
    digital write pin P1 to 1
  else
    digital write pin P1 to 0
```

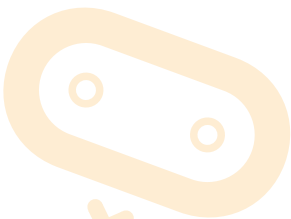
- **Analog Read:** Micro:bit uses an **analog read** block to read the value as a 10-digit number that ranges from 0 to 1023. The **analog read** block is a value based block, Which means that you have to insert it into a block with a matching shape. In this case, we use this value and compare it with the value( It indicates that the brightness is bright and we don't need to turn on the LED when threshold we set is larger than this value; when it is less than this value, it indicates that the brightness is dark and we need to turn on the LED).
- **Download the program to micro:bit to see what happens.**

### Result

The LED will not light up during the day (in a bright environment). And when the daylight gradually dims into the night, the LED will light up automatically. If the LED does not change, try to adjust the threshold you set.



*Do you have plants in your home? Did you know that a plant needs light to grow? If there is not enough light, plants will wither. Let's protect them and think about how to use this light sensor to solve this problem.*



## Experiment 7 – Plant soil moisture detection

### Instruction

People need water to survive, so do plants. Soil moisture indicates the moisture in the soil. As long as the soil moisture value is suitable for the plant, the plant can get water from the soil. Therefore, in this experiment, we will make a plant soil moisture detector to ensure that our plants can get enough water and grow.

### Target

- Learn how moisture sensor work and use it to make a moisture detector with LED and buzzer.
- Learn how to draw icon on Micro:bit's matrix LEDs.
- Learn how to use if/else if/else statement.

### Required Parts

- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-Moisture Sensor x1
- Crowtail-LED(Yellow) x1
- Crowtail-Buzzer x1
- Crowtail-Cable x3
- USB cable x1

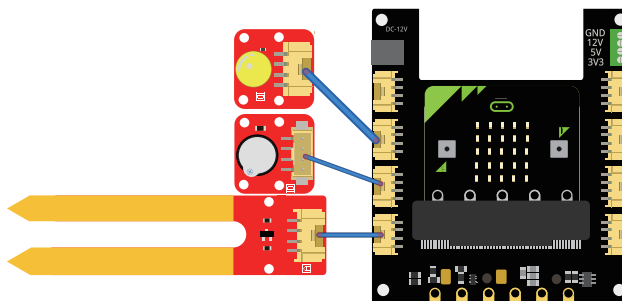
### Hardware learning and connection

This Moisture sensor can be used to detect the moisture of soil and thus to monitor if the plants in your garden need some water.



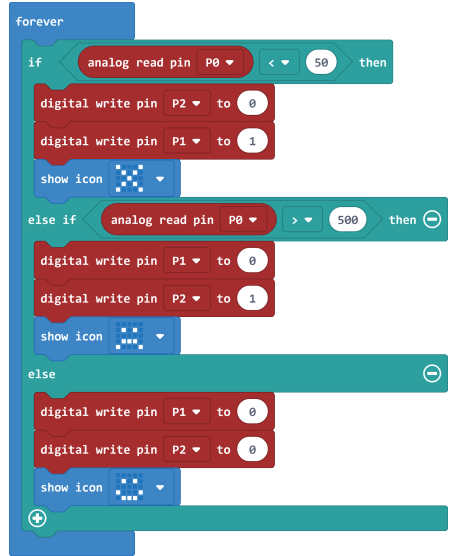
This sensor uses the two probes to pass electricity through the soil, and then reads the resistance to get the moisture level. More water makes the soil conduct electricity more easily (less resistance), while dry soil conducts electricity poorly (more resistance). Compared to the other moisture sensors using the same moisture test method, this module has super long legs, making it suitable for actual applications.

Connect Crowtail-Moisture Sensor, Crowtail-Buzzer and Crowtail-LED to P0, P1 and P2 ports of Crowtail-Base shield for Micro:bit. The hardware connections are as follows:



## Programming and note

- **Show Icon:** We can use this **show icon** block to draw icons on the Micro:bit's matrix LEDs. Many icons have already been created in this block, so we just need to select the icon to be drawn.
- **If/else if/else statement:** If/else if/else is a logical structure that is very similar to if/else. Actually, it is derived from the if / else structure. It can resolve judgments when there are more than two cases. If the logical statement that is attached to if (analog read from P0 is less than 50) is true, then it will execute the code blocks inside of the **if**. If the logical statement that is attached to **else if** (analog read from P0 is more than 500) is true, then it will execute the code blocks inside of the **else if**. Else (analog read from P0 is larger than 50 and less than 500), it will execute the **else** blocks.
- **Download the program to micro:bit to see what happens.**



```
forever
  if (analog read pin P0 < 50) then
    digital write pin P2 to 0
    digital write pin P1 to 1
    show icon [X icon]
  else if (analog read pin P0 > 500) then
    digital write pin P1 to 0
    digital write pin P2 to 1
    show icon [sad face icon]
  else
    digital write pin P1 to 0
    digital write pin P2 to 0
    show icon [happy face icon]
```

### Result

When the soil moisture sensor is inserted into the dry soil, the buzzer will beep and the X icon will be displayed on the micro: bit matrix LED; when it is inserted into the Water-filled soil, the LED will light up and a sad icon will be displayed on the micro: bit's matrix LED; when it is inserted into the moist soil, the LED will not light up, the buzzer will not beep, and a happy icon will be displayed on the micro: bit's matrix LED.



*Can we make an automatic watering system for plants after detecting low soil moisture? And how to make it?*

## Experiment 8 – Intrusion alert

### Instruction

Have you ever played any adventure games? They can always open the institutions when the enemy enters their private territory. How does this work? In fact, this is the role of intrusion alarms. Through this experiment, we will teach you how to make your own intrusion alarm, and quickly put it in your private space to protect your own space!

### Target

- Learn how PIR motion sensor works and use it to make an intrusion alarm with buzzer.
- Learn how to clear Micro:bit's matrix LEDs.

### Required Parts

- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-PIR Motion Sensor x1
- Crowtail-Buzzer x1
- Crowtail-Cable x2
- USB cable x1

### Hardware learning and connection

Crowtail- PIR Motion Sensor(Passive Infrared Sensor) can detect infrared signals caused by motion. If the PIR sensor notices the infrared energy, the motion detector will be triggered and the sensor outputs HIGH on its SIG pin. The detecting range can be adjusted by a potentiometer soldered on its circuit board, the max detecting range of it up to 6 meters. ***PO is a port with pull-down, please do not connect PIR motion sensor to this port!*** For the micro: bit, when the PIR module detects an object movement, it will change from low level to high level and output high level continuously for 5 seconds, which means you don't need to set extra delay.



Connect Crowtail-PIR Motion Sensor and Crowtail-Buzzer to P2 and P1 ports of Crowtail-Base shield for Micro:bit. The hardware connections are as follows:

