

Programming and note

```
forever
  if digital read pin P2 = 1 then
    digital write pin P1 to 1
    show icon [angry face]
  else
    clear screen
    digital write pin P1 to 0
```

- **If/else:** In this case, when the value digital read from the PIR motion sensor is true(1), turn on the buzzer and show an angry icon for 5 seconds, otherwise, clear the micro:bit's matrix LEDs and turn off the buzzer.
- **Clear screen:** This `clear screen` block will clear all the information displayed on micro:bit's matrix LEDs when it runs. Here, we clear matrix LEDs when PIR motion sensor does not detect the movement of Infrared energy.
- **Download the program to micro:bit to see what happens.**

Result

Wave your hand in front of the PIR motion sensor, and you will see an angry icon on the micro: bit matrix LED, and the buzzer will beep for 5 seconds. If you stop any movement, the icon will no longer be displayed and the buzzer will stop beeping.



What does infrared energy mean? Can PIR motion sensor detect your pet's movement?

Experiment 9 – Make an adjustable table lamp

Instruction

The eye is a very important organ for us. We can see the colorful world by them. so protecting eyes is very important to us! Reading in a too bright or too dark environment can cause damage to our eyes, so when we read a book, we must pay attention to the brightness of the light to be appropriate. Maybe what we need at this time is a table lamp with adjustable brightness, let's start to make it!

Target

- Learn how linear potentiometer works and use it to make an adjustable table lamp with LED.
- Learn how to change the brightness of the LED.
- Learn how to create variables to store value.
- Learn how to remap numbers from one range to another.

Required Parts

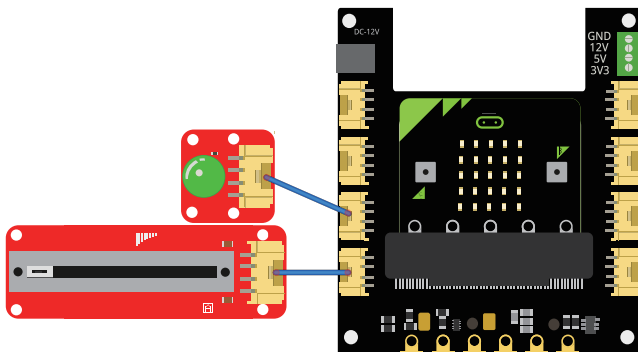
- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-Linear Potentiometer x1
- Crowtail-LED(Green) x1
- Crowtail-Cable x2
- USB cable x1

Hardware learning and connection

The Crowtail- Linear Potentiometer module uses a linear variable resistor with a maximum resistance of 10KΩ. When you move the slider from one side to the other, its output voltage will range from 0 V to the VCC you applied. The panel mount design makes this module easy to be installed in projects.



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



Programming and note

```
forever
  set potentiometerValue to analog read pin P0
  analog write pin P1 to map potentiometerValue from low 0 high 1023 to low 0 high 255
```

- **Set to:** You can use this **set to** block to set the value for a variable that selected from a list of the variables you create. You can create variables by click Variables → Make a Variable and then naming it.
- **Analog write:** Just like the **analog read** block, the **analog write** block deals with a range of values, but instead of reading a pin as an input, the **analog write** block outputs an analog value to a pin. In this case, we use this block to set the brightness of the LED, but it could be a tone for a buzzer, a motor speed and etc.
- **Map:** This is a mathematical method used to map values from one interval to values from another interval. The mapped value range will be reduced or enlarged by a certain proportion. Here we need to map the potentiometerValue (value between 0 and 1023) to value (between 0 and 255) and write it to P1 port. For example, if the value of potentiometerValue is 240, then after mapping, we get the value is 60 and write it to port P1. The reason we use this mapping function is that the LED has a brightness level of 256 (0-255). When the brightness level is greater than 255, the brightness of the LED will not change.
- **Download the program to micro:bit to see what happens.**

Result

Toggle the potentiometer and you will find that the LED light will become brighter or dimmer depending on the position of the potentiometer. When you turn the potentiometer to the far left, the LED light goes out, and when you turn to the right, the LED light is the brightest.



Since the brightness of the lamp can be adjusted in this way, is the sound volume of the stereo adjusted in this way?

Experiment 10 – Automatic control

Instruction

How do we safely use high-power appliances, do you know how to control the elevator? In fact, it needs an automatic control system, in which the relay plays an important role in helping us. This system guarantees our electricity safety by controlling large currents with small currents. In this lab, we will learn how to use relays to build an automatic control system for controlling motors.

Note: Do not connect high-power electrical appliances, otherwise accidents will occur due to improper use!

Target

- Learn how relay works and use it to control the DC motor with switch.

Required Parts

- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-Relay x1
- Crowtail-Switch x1
- Crowtail-Cable x2
- DC Motor x1
- USB cable x1
- Jumper Wire x1

Hardware learning and connection

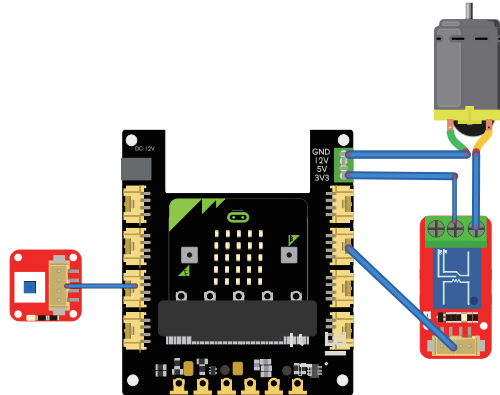


The Relay is the most commonly used module in some actual applications such as home automation, and sometimes you need to control large current such as the air conditioning/water heater. That is what this Large Current Relay Module can help with.

The connector of the relay module has three sockets: common (COM), normally closed (NC), and normally open (NO). COM: common pin. NC (Normally Closed): the normally closed configuration is used when you want the relay to be closed by default, meaning the current is flowing unless you send a signal to the relay module to open the circuit and stop the current. NO (Normally Open): the relay is always open, so the circuit is broken unless you send a signal to close the circuit.

This Large Current Relay Module is a high-quality Single Pole Double Throw Relay (SPST). When the relay not triggered, the 2 outputs are disconnected; when the relay triggered (Logic Low), the 2 outputs connected, the max current can be up to 30A@240VAC or 30A@30VDC, which would be enough for most of the home appliances.

Connect Crowtail-Switch and Crowtail-Relay to P1 and P8 ports of Crowtail-Base shield for Micro:bit. Loosen the screws of the relay, connect one end of the jumper to the 3V3 port of the Base Shield and the other end of the jumper to the COM port of the Relay. Connect the two wires of the motor to the NO port of the relay and the GND port of the Base Shield. **Note that the power consumed by the motor is large, you need to connect a 12V power supply to the Crowtail-Base shield for Micro:bit.** The hardware connections are as follows:



Programming and note

- **Digital write:** When the switch is detected to open, we use the **digital write** module to turn on the relay (1), and then the battery box power can flow through the DC motor so that the battery box and the DC motor could form a closed circuit, then DC motor starts to work. When the switch is detected to be closed (0), the relay will close, then the closed circuit will be blocked, so the DC motor will no longer work.

```

forever
  if digital read pin P1 = 1 then
    digital write pin P8 to 1
  else
    digital write pin P8 to 0
  
```

- **Download the program to micro:bit to see what happens.**

Result

When the switch is pressed for the first time, the relay will open and the DC motor will start working. When the switch is pressed a second time, the relay will close and the DC motor will stop working.



What role do relay plays in automatic control systems? What are the appliances that also use relays in our lives?

Experiment 11 – Automatic door

Instruction

It's so convenient when we enter and exit the gate of the community, that the door will open and closes automatically after we press the switch of the gate. How about making a mini automatic door?

Target

- Learn how servo work and use it to make a mini automatic door with touch sensor.
- Learn how to rotate the servo.

Required Parts

- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-9G Servo x1
- Crowtail-Touch Sensor x1
- Crowtail-Cable x1
- USB cable x1

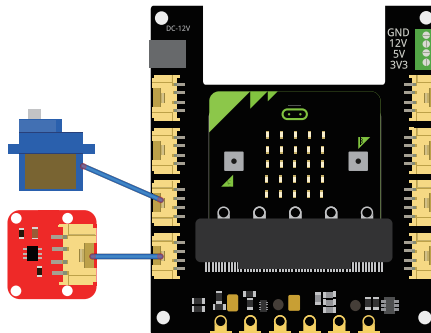
Hardware learning and connection

Tower Pro SG90 is a high quality, low-cost servo for all your mechatronic needs. It comes with a 4-pin power and control cable, mounting hardware. Servo is used in many intelligent situations, such as automatic doors, robots, aerial models, etc.

It can be said that the servo is almost an indispensable module in the field of intelligent control. Have you thought about adding a rotatable part to your smart product, such as automation field, robot's head and hand. **Note: For all experiments using the servo module, you need to connect a 12V power supply for the Crowtail-Base shield for Microbit.**



Connect Crowtail-9G Servo and Crowtail-Touch Sensor to P1 and P0 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
    servo write pin P1 to 180
    pause (ms) 2000
  else
    servo write pin P1 to 0
```

- **Servo write:** Use this **servo write** block to write a value to the servo on the specified pin and control the shaft. This function will move the shaft of a standard servo(Crowtail-9G Servo) to the specified angle, or set the speed of a continuous rotation servo (0 specifies full speed in one direction, 180 specifies full speed in the other, and approximately 90 specifies no movement). In this case, we will move the servo's shaft to 180 degrees and 0 degrees these two specified angles.
- **Pause:** After we moved the servo's shaft to 180 degrees (opening the door), we used a "pause" block to pause the code so that the door would not close immediately and there was enough time to enter or exit the door.
- **Download the program to micro:bit to see what happens.**

Result

Press the touch sensor, the shaft of the servo will move to 180 degrees. After two seconds, if the touch sensor is not touched, the servo's shaft will move back to 0 degrees.



How to make automatic doors smarter? When we stand in front of the door, how do we make it open automatically without any keys or switches?

Experiment 12 – Weather station

Instruction

The changing weather is so annoying, I even got a cold because of it! No, I have to think of a way to remind myself of changes in the weather at all times. I have to pay attention to keep warm and never catch a cold again. Less gossip, let's make a weather station to ensure we can obtain temperature and humidity anytime and anywhere to solve this problem!

Target

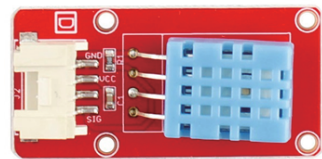
- Learn how Temperature&Humidity sensor work and use it to make a weather station.
- Learn how to add an extension package.
- Learn how to use two onboard buttons of micro:bit.

Required Parts

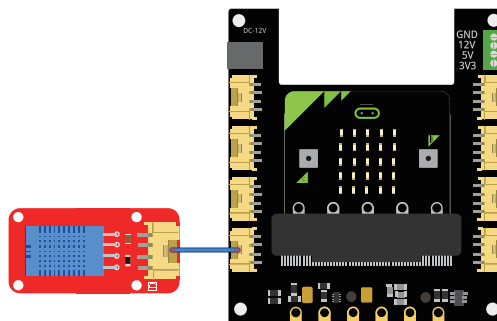
- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-Temperature&Humidity Sensor x1
- Crowtail-Cable x1
- USB cable x1

Hardware learning and connection

This module can help you detect the temperature and humidity of the environment of your house. The module contains a DHT11 temperature & humidity sensor that is a complex sensor with a calibrated digital signal out. It uses digital module acquisition technology and temperature&humidity sensor technology. The sensor consists of a resistance type moisture element and an NTC temperature measuring element. Because of the single-wire serial interface, it is easy to use the module.



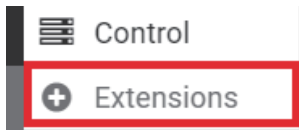
Connect Crowtail-Temperature&Humidity Sensor to P0 port of Crowtail-Base shield for Micro:bit. The hardware connections are as follows:



Programming and note

To using the Crowtail-Temperature&Humidity Sensor, we need to add the extension package. Follow the steps below to add extension package:

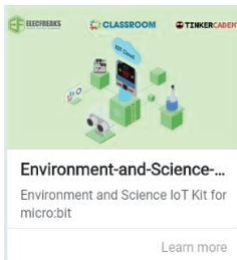
- **Step 1:** Click on the Extensions



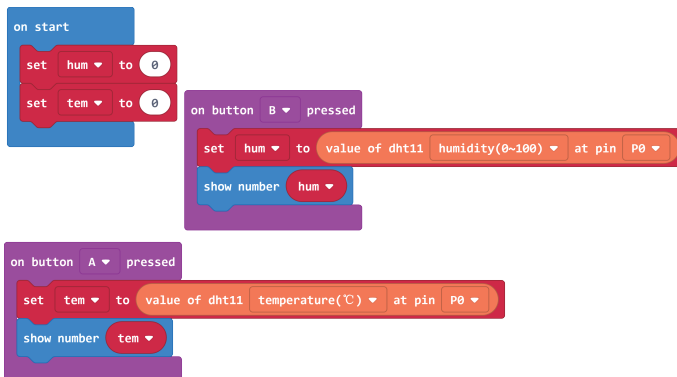
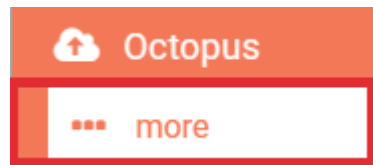
- **Step 2:** Enter the address in the search bar and search: <https://github.com/Tinkertanker/pxt-iot-environment-kit>



- **Step 3:** You will view the search results and click on it.



- **Step 4:** Then you can see the extensions package you added in Makecode and click to use the DHT11 block.



- **On button pressed:** Start an event handler (part of the program that will run when something happens, like when a button is pressed). This handler works when button A or B is pressed, or A and B together. When you are using this function in a web browser, click the buttons on the screen instead of the ones on the micro:bit. In this experiment, when button A is pressed, show the temperature on micro:bit's matrix LEDs; when button B is pressed, show humidity on micro:bit's matrix LEDs.
- **Value of dht11...at pin...:** Use this block in the extension package we added to set the pin of Crowtail-Temperature&Humidity Sensor and get data from it. Available data includes Fahrenheit, Celsius and Humidity. In here, we get Celsius and Humidity from the sensor.

- Download the program to micro:bit to see what happens.

Result

When you press button A on the micro:bit, the matrix LEDs will show you the current temperature. When you press button B on the micro:bit, the matrix LEDs will show you the current humidity.



How to build a more comprehensive weather station so that we can get air quality, temperature, humidity, ultraviolet information and more?

Experiment 13 – Ultrasonic ranging smart door

Instruction

Do you remember the thinking problem in experiment 11 ? Did you find the answer? If you don't, rest assured that we will lead you to build such a smart, automatic door in this experiment. The only difference is that in this section we will use delicate and magical ultrasonic sensors.

Target

- Learn how ultrasonic ranging sensor works and use it to make a smart door with servo and buzzer.
- Learn how to get the distance from the ultrasonic ranging sensor.

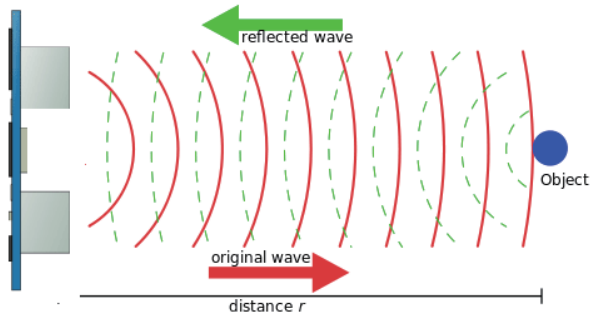
Required Parts

- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-Ultrasonic Ranging Sensor x1
- Crowtail-9G Servo x1
- Crowtail-Buzzer x1
- Crowtail-Cable x2
- USB cable x1

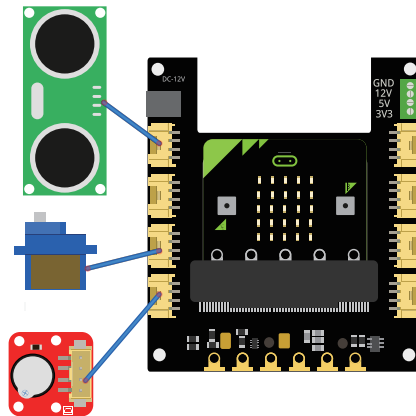
Hardware learning and connection

This HC-SR04 has stable performance and high ranging accuracy. The process of ultrasonic ranging: Ultrasonic pulses travel outward until they encounter an object, The object causes the wave to be reflected back towards the unit. The ultrasonic receiver would detect the reflected wave and stop the timer. Now read the time of the counter, which is the ultrasonic propagation time in the air. According to the formula: Distance = ECHO high level time X ultrasonic velocity (Speed of Sound in air 340m/sec) / 2, you can calculate the distance to the obstacle.





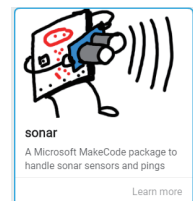
Connect Crowtail-Ultrasonic Ranging Sensor, Crowtail-Buzzer and Crowtail-9G Servo to P13&P15, P0 and P1 ports of Crowtail-Base shield for Micro:bit. **Note: You need to connect 12V power supply for the Crowtail-Base shield for Micro:bit.** The hardware connections are as follows:



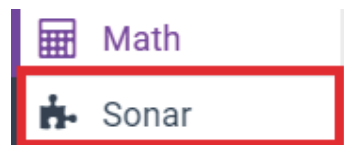
Programming and note

Just as Crowtail-Temperature&Humidity Sensor, we need to add the extension package for Crowtail-Ultrasonic Ranging Sensor before we use it. Follow the steps below to add extension package of ultrasonic ranging sensor:

- **Step 1:** Click on the Extensions, then search “sonar” in the search bar.



- **Step 2:** Add sonar package to your Makecode and you can see it.



- **Ping trig...echo...unit:** This block is inside the sonar package we added. We can use this block to get the time, distance in cm and distance in inches detected from the ultrasonic ranging sensor. In this experiment, we set the trig pin and echo pin of the sensor are P13 and P15 port of micro:bit. And the data we want to get is the distance in cm.

```

on start
  set distance to 0

forever
  ping trig P13
  set distance to echo P15
  unit cm
  if distance < 20 then
    servo write pin P1 to 180
    show string "Welcome!"
    pause (ms) 2000
    start melody power down repeating once
  else
    servo write pin P1 to 0
  pause (ms) 1000
  
```

- **Start melody repeating:** We use the **Start Melody Repeat** block to play a melody to remind the door is about to close, please be careful! When the melody ends, the door begins to close.

- **Download the program to micro:bit to see what happens.**

Result

When the distance detected by the ultrasonic sensor is less than 20 cm, open the door (the servo's shaft rotates 180 degrees), and the matrix LEDs of the micro: bit will display "Welcome". After 2 seconds, the buzzer will play a melody to warn that the door is about to close. Then, when the distance detected by the ultrasonic sensor more than 20 cm, the door start closing (the servo's shaft rotates 0 degrees).



How to apply ultrasonic ranging sensors to the car to make the car have an obstacle avoidance function?

Experiment 14 – Make an accurate clock

Instruction

Have you ever been late for school because you forgot to set the alarm clock? This is really embarrassing! Let's make an alarm clock with a RTC module together today, and say goodbye to being late!

Target

- Learn how RTC work and use it to make a clock with buzzer.
- Learn how to set time for RTC.

Required Parts

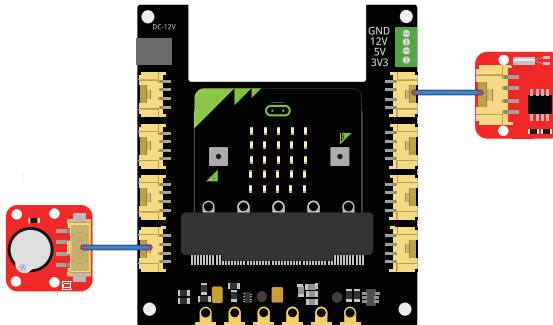
- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-RTC x1
- Crowtail-Buzzer x1
- Crowtail-Cable x2
- USB cable x1

Hardware learning and connection

This tiny RTC module is based on the clock chip DS1307 which communicates with microcontrollers with I2C protocol. The clock/calendar provides seconds, minutes, hours, day, date, month and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. Besides, this module is really low power consumption, it can serve you more than a month with a CR1220 battery.



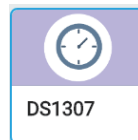
Connect Crowtail-RTC and Crowtail-Buzzer to I2C and P0 port of Crowtail-Base shield for Micro:bit. The hardware connections are as follows:



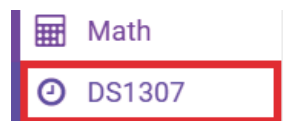
Programming and note

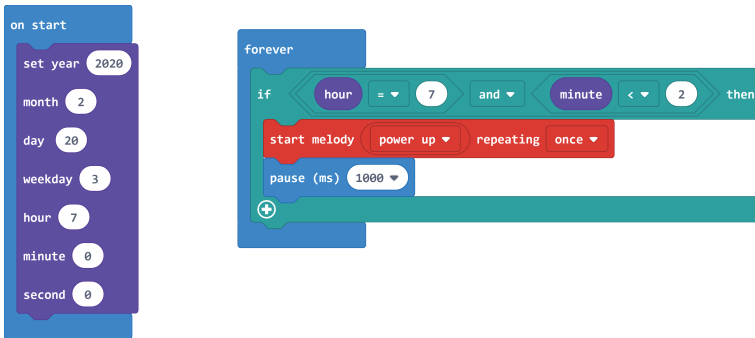
Before we using the RTC, we need to add an extension package for RTC(DS1307). Follow the steps below to add an extension package of RTC.

- **Step1:** Click on the Extensions, then search “ds1307” in the search bar.



- **Step2:** Add the DS1397 package to your Makecode and you can see it.





- **Set date and time:** We use this block to set the current date and time. In this case, we start the time from the beginning of the program, so the time needs to be set exactly the same as your local time, or you can set it faster because you need time to upload the program, otherwise the clock may be inaccurate.
- **And:** It is a logic **and** block. This combines two logical statements into one statement that returns true when both of the other statements are true and only true. In this experiment, when hour equal to 7 and minute less than 2, then if run the code inside if statement. Otherwise, the code inside if statement will not run.
- **Download the program to micro:bit to see what happens.**

Result

After uploading the program, the buzzer will start playing music for 2 minutes. Then, if you don't stop powering the micro: bit, the buzzer will play the "alarm" for 2 minutes every day starting from 7 am.



Can you display the time on the micro: bit's matrix LEDs in the format we see every day?

Experiment 15 – Distance display

Instruction

It takes too much time to scroll the messages on the micro: bit's matrix LEDs, which makes viewing very inconvenient. But don't worry, you will learn a better way to display messages or data from this experiment. It does not require you to spend time waiting for the information to scroll. You can see all the messages and data at a glance. This method is to use OLED for display.

Target

- Learn how OLED works and use it to make a monitor with ultrasonic ranging sensor to show distance that detected by the ultrasonic ranging sensor.
- Learn how to show number and string on OLED.

Required Parts

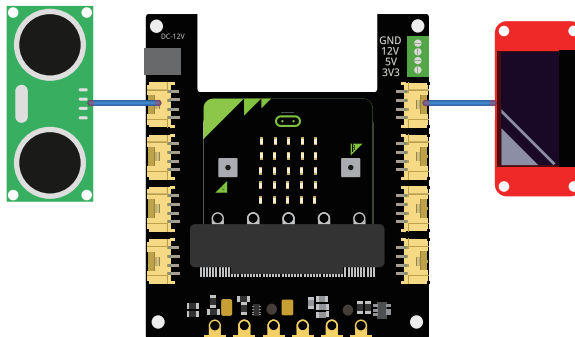
- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-Ultrasonic Ranging Sensor x1
- Crowtail-OLED x1
- Crowtail-Cable x2
- USB cable x1

Hardware learning and connection

Crowtail- OLED is constructed from the 128 x 64 dot-matrix OLED module. The display offers high brightness, self-emission, high contrast ratio, slim/thin outline, wide viewing angle, wide temperature range and low power consumption. You can almost display anything with this OLED module, such as text and pictures! This kind of OLED is very suitable to use in your project which needs a small display to show data or pictures.



Connect Crowtail-Ultrasonic Ranging Sensor and Crowtail-OLED to P13&P15 and I2C ports of Crowtail-Base shield for Micro:bit. The hardware connections are as follows:




Programming and note

We need to add two extensions package before we use OLED(SSD1306) and ultrasonic ranging sensor, Follow the steps below to add extension packages of them.


- **Step1:** Click on the Extensions, then search "OLED" and "sonar" in the search bar.





sonar
A Microsoft MakeCode package to handle sonar sensors and pings

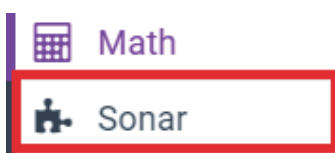
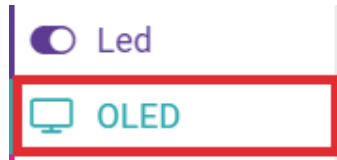
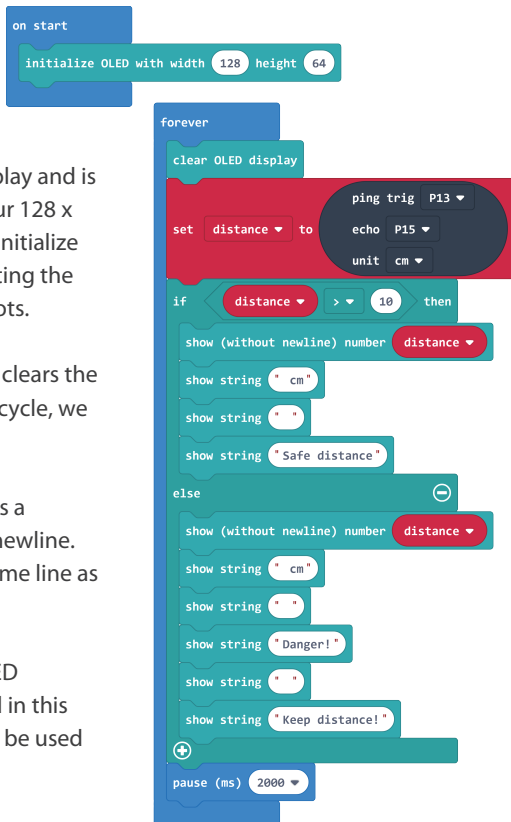
[Learn more](#)



oled-ssd1306
Tinkercademy MakeCode package for using the SSD1306 OLED

[Learn more](#)

- **Step2:** Add sonar and OLED packages to your Makecode and you can see it.

```

on start
  initialize OLED with width 128 height 64

forever
  clear OLED display
  ping trig P13
  set distance to echo P15
  unit cm
  if distance > 10 then
    show (without newline) number distance
    show string "cm"
    show string "**"
    show string "Safe distance"
  else
    show (without newline) number distance
    show string "cm"
    show string "**"
    show string "Danger!"
    show string "**"
    show string "Keep distance!"
  pause (ms) 2000
  
```

- **Initialize OLED with width...height...**
This block is used to set up the OLED display and is ready for micro: bit use. In order to use our 128 x 64 dot matrix OLED module, we need to initialize the OLED according to the hardware, setting the width to 128 dots and the height to 64 dots.
- **Clear OLED display:** This block is used to clears the OLED display. In this case, every time we cycle, we will display on a clear OLED screen.
- **Show(without newline) number:**Displays a number on the OLED module without a newline. Note that the next line(" cm") is on the same line as this block display.
- **Show string:** Displays a string on the OLED module with a newline. If nothing is filled in this block, nothing will be displayed. This can be used for a blank line.

- Download the program to micro:bit and see what happens.

Result

When the distance detected by the ultrasonic ranging sensor is less than 10 cm, the distance and warning("Danger!" and "Keep distance!") will be displayed on OLED. Otherwise, the distance and Security Information("Safe distance") will be displayed on OLED.



Can OLED be used to make a more comprehensive display, showing all the data and information we need on the OLED, such as time, light, temperature and humidity?

Experiment 16 – Wireless communication

Instruction

Wireless communications are everywhere in our daily life, such as mobile phones we most often see and use. Wireless communication brings us a lot of conveniences. It enables us to communicate with each other even thousands of miles apart. Thanks to the radio/Bluetooth module on the micro: bit board, wireless communication can also be achieved between our two micro: bits or communicate with the phone.

Target

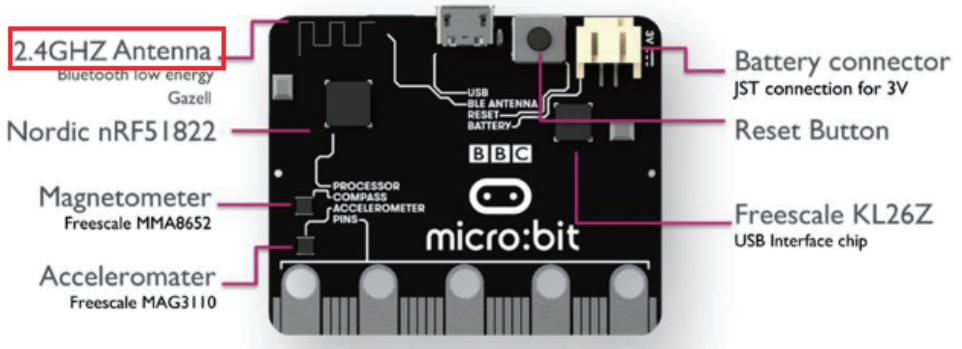
- Learn how radio work on micro:bit and use it to communicate between 2 micro:bits.
- Learn how to send and receive numbers between 2 micro:bits.

Required Parts

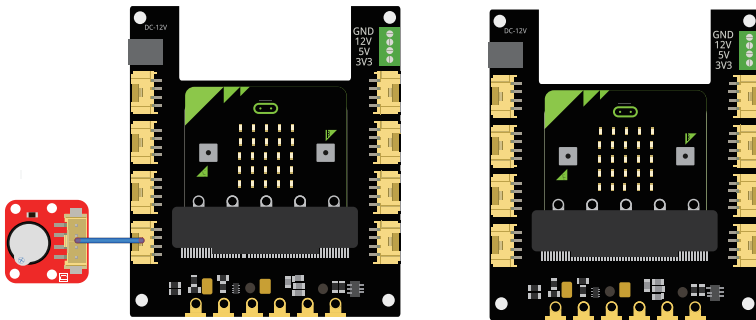
- Micro:bit x2
- Crowtail-Base shield for Micro:bit x2
- Crowtail-Buzzer x1
- Crowtail-Cable x1
- USB cable x1

Hardware learning and connection

Micro: bit has two wireless communication methods, one is radio communication and the other is Bluetooth communication. These two hardware modules are embedded in the upper left position behind the micro: bit, as shown in the figure. In this experiment, we will learn the standard radio on micro:bit which allows you to communicate between two or more micro:bits.

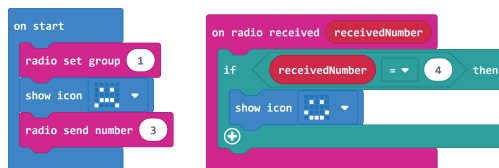


Connect Crowtail-Buzzer to P0 port of Crowtail-Base shield for Micro:bit(named bit1). Plug another micro:bit into Crowtail-Base shield for Micro:bit(named bit2, if you don't have two Crowtail-Base shield for Micro:bit, you don't have to plug this micro:bit into Crowtail-Base shield for Micro:bit). The hardware connections are as follows:

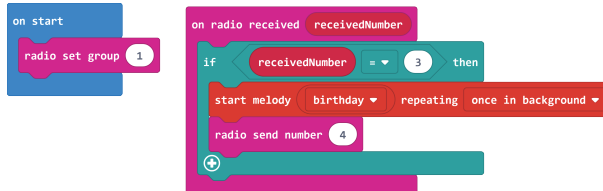


Programming and note

Send_code:



Receiver_code:



- **Radio set group:** Make a program have the group ID you tell it for sending and receiving with radio. A group is like a cable channel (a micro:bit can only send or receive in one group at a time). A group ID is like the cable channel number. If you do not tell your program which group ID to use with this function, it will figure out its own group ID by itself. If you load the very same program onto two different micro:bits, they will be able to talk to each other because they will have the same group ID. In this experiment, we set the group of two micro:bits to 1 so that they can communicate with each other.
- **Radio send number:** This block is to broadcast a number to other micro:bits connected via radio. You need to enter a parameter for this block which is the number you want to send.
- **On radio received receivedNumber:** Run part of a program when the micro:bit receives a number over the radio. "receivedNumber" is a parameter, it's the number that was sent in this packet or 0 if this packet did not contain a number. If the number received by bit1 (receiver_code) is 3, the code inside the if statement of bit1 will be run. If the number received by bit2 (send_code) is 4, the code inside the if statement of bit2 will be run.
- **Download the send_code program to bit2 and receiver_code program to bit1 and see what happens.**

Result

At first, the micro: bit matrix LEDs named bit2 will display a crying face, and then the micro: bit of bit1 will immediately play a melody (comfort bit2), and the crying face of bit2 will become a smiling face.



Does wireless communication require two or more micro:bits? Then if there are no two micro:bits, can the micro: bits be controlled wirelessly? For example, the module is used to wirelessly control the micro: bit.

Experiment 17 – Remotely control your micro:bit

Instruction

When you're too sleepy, you definitely don't want to get up from your bed and turn off the lights. When you are busy with other things, you don't feel like being interrupted by knocking on the door. Well, in this experiment we are ready to help you solve this problem. We will use the remote control to remotely control these things, instead of having to leave for them.

Target

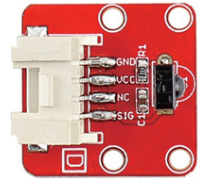
- Learn how IR receiver work and use it to control the LED with infrared remote control and servo connected to micro:bit.
- Learn how to detect the keys pressed by the infrared remote control.

Required Parts

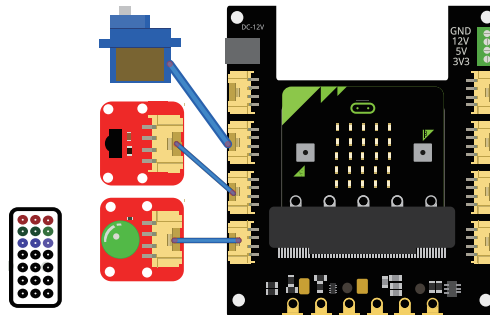
- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-IR Receiver x1
- Crowtail-LED(Green) x1
- Crowtail-9G Servo x1
- Crowtail-Cable x2
- Infrared Remote Control x1
- USB cable x1

Hardware learning and connection

The Crowtail- IR Receiver module uses the HS0038B which is miniaturized receivers for infrared remote control systems and it is the standard IR remote control receiver series, supporting all major transmission codes. The IR detector has a demodulator inside that looks for modulated IR at 38 kHz, we can use our infrared remote control to control this module well. The Infrared Receiver can receive signals well within 10 meters. If more than 10 meters, the receiver may not get the signals.



Connect Crowtail-LED, Crowtail-IR Receiver and Crowtail-9G Servo to P0, P1 and P2 ports of Crowtail-Base shield for Micro:bit. **Note: You need to connect 12V power supply for Crowtail-Base shield for Micro:bit.** The hardware connections are as follows:



Programming and note

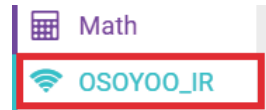
Before using the IR receiver, we need to add an extension package for this module. Follow the steps below to add an extension package for it.

- **Step1:** Click on the Extensions, then enter the following link "https://github.com/osoyoo/OSOYOO_IR" in the search bar.

https://github.com/osoyoo/OSOYOO_IR



- **Step2:** Add the package to your Makecode and you can see it



```

on start
  connect ir receiver to P1

on NUM1 button pressed
  digital write pin P0 to 1

on NUM2 button pressed
  digital write pin P0 to 0

on NUM3 button pressed
  servo write pin P2 to 180
  pause (ms) 1000

on NUM4 button pressed
  servo write pin P2 to 0
  pause (ms) 1000
  
```

- **Connect ir receiver to:** This block is used to tell the micro:bit that which port the IR receiver is connected.
- **On button pressed:** This block is to detect whether the key on the infrared remote control is pressed and which key is pressed. In this block, detecting which key was pressed is optional. When the key we selected is pressed, the IR receiver will run the code we need to process when the key is pressed.
- **Download the program to micro:bit to see what happens.**

Result

When you press the "1" key on the controller, the LED will turn on; when you press the "2" key, the LED will turn off. When you press the "3" key, the shaft of the servo will rotate to 180 degrees(open door); when you press the "4" key, the shaft of the servo will rotate to 0 degrees(close door).



Note: Before using the infrared remote control, please install a coin cell battery for it. The type of the coin cell battery is CR2025.



Can we use remote knowledge to make a remote control car, and if so, what other components and modules do we need?

Experiment 18 – Smart music box

Instruction

Have you ever seen a robot that says hello when someone approaches? That is a very interesting thing, so let's do a similar project in this lesson, put it in a box and make a fun and smart music box!

Target

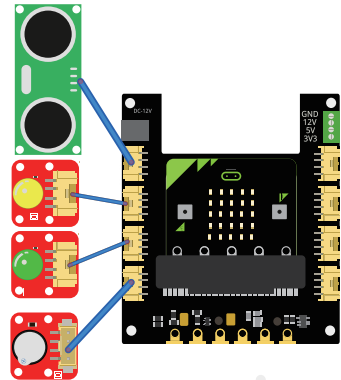
- Use the modules and programming knowledge we have learned to make a smart music box.
- Learn how to stop the buzzer melody.

Required Parts

- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-Ultrasonic Ranging Sensor x1
- Crowtail-Buzzer x1
- Crowtail-LED(Green) x1
- Crowtail-LED(Yellow) x1
- Crowtail-Cable x4
- USB cable x1

Hardware learning and connection

Connect Crowtail-Ultrasonic Ranging Sensor to P13&P15 port of Crowtail-Base shield for Micro:bit. Connect Crowtail-Buzzer, Crowtail-LED(Green) and Crowtail-LED (Yellow) to P0, P1 and P2 ports of Crowtail-Base shield for Micro:bit. The hardware connection are as follows:



Programming and note

- **Stop melody all:** Stop playing a musical melody. We can use this block to stop melodies that are played either in the foreground or background. In this case, if the value of dis more than 10, we use this block to stop melody immediately.

- **Blocks overview**

1. Add extension package for the ultrasonic ranging sensor.
2. Set a variable named dis to store the distance detected by the ultrasonic ranging sensor.
3. Read the distance in cm from ultrasonic and store in dis.
4. If the value of "dis" less than 10, show "heart" icon on matrix LEDs, play a melody and turn two LEDs on.

```
on start
  set dis to 0

forever
  set dis to ping trig P13
  echo P15
  unit cm
  if dis < 10 then
    show icon heart
    digital write pin P1 to 1
    digital write pin P2 to 1
    start melody entertainer repeating once
    pause (ms) 2000
  else
    clear screen
    stop melody all
    digital write pin P1 to 0
    digital write pin P2 to 0
    pause (ms) 1000
```

5. If the value of dis more than 10, clear the matrix LEDs, stop melody and turn two LEDs off.
6. Loop in forever block.

- Download the program to micro:bit to see what happens.

Result

When the distance detected by the ultrasonic sensor is less than 10cm, the matrix LEDs on the micro: bit will display a heart icon, the buzzer will play the melody, and two LEDs will light up. When the ultrasonic sensor detects a distance greater than 10cm, the micro: bit will clear the matrix LEDs display, immediately stop the buzzer to play the melody and turn off the two LEDs.



How can we use this project to make our own songs to welcome others or pranks?

Experiment 19 – Remote control fan

Instruction

When summer comes, you will find it great and convenient to have a remote control fan. With it, we no longer need to manually turn on and off the fans, especially when you are too hot to want to move. When you finish this experiment, you will get your own mini micro: bit remote fan! Let's get started.

Target

- Make your own mini remote control fan.

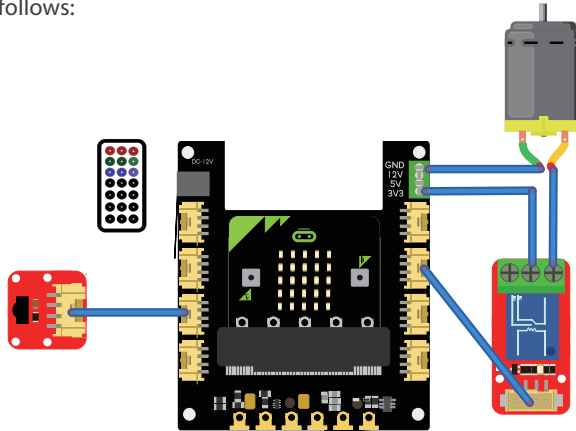
Required Parts

- Micro:bit x1
- Crowtail-Base shield for Micro:bit x1
- Crowtail-IR Receiver x1
- Crowtail-Relay x1
- Crowtail-Cable x2
- Infrared Remote Control x1
- DC Motor x1
- USB cable x1
- Jumper Wire x1

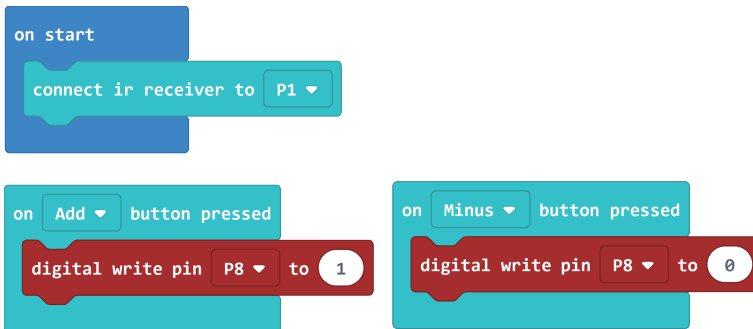
Hardware learning and connection

Connect Crowtail-IR Receiver to P1 port of Crowtail-Base shield for Micro:bit. Connect Crowtail-Relay to P8 port of Crowtail-Base shield for Micro:bit. Loosen the screws of the relay, connect one end of the jumper to the 3V3 port of the Base Shield and the other end of the jumper to the COM port of the Relay. Connect the two wires of the motor to the NO port of the relay and the GND port of the Base Shield. **Note that the power consumed by the motor is large,**

you need to connect a 12V power supply to the Crowtail-Base shield for Micro:bit. The hardware connection are as follows:



Programming and note



- **On button pressed:** This block is to detect whether the key on the infrared remote control is pressed and which key is pressed. As soon as the key is pressed, this block detects and runs the code we want to run when the button is pressed. When the "+" key is pressed, turn the relay on; when the "-" key is pressed, turn the relay off.
- **Download the program to micro:bit and see what happens.**

Result

When you press "+" key, the fan will be turned on; when you press "-" key, the fan will be turned off.



The fan can now only be turned on or off, but the wind speed cannot be adjusted. Can we make a fan that can turn on and off and regulate the wind speed?

Experiment 20 – Car remote control

Instruction

Just like our brains, controllers play a very important role in remotely controlled cars. It is responsible for telling the car what to do and how to do it. Why not try to use a micro:bit as a controller to simulate how the car is controlled, such as lights, horns, steering, etc. This is what we will do in this experiment.

Target

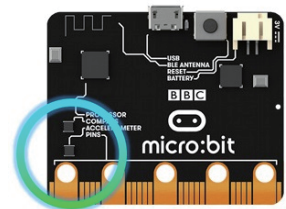
- Make a remote controller and simulate controlling the car.
- Learn how to use gesture blocks to detect gestures you take when holding a micro: bit (such as shaking a micro: bit)

Required Parts

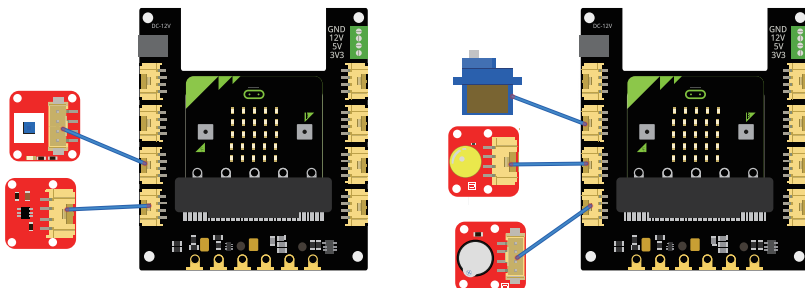
- Micro:bit x2
- Crowtail-Base shield for Micro:bit x2
- Crowtail-Touch Sensor x1
- Crowtail-Switch x1
- Crowtail-LED(Yellow) x1
- Crowtail-Buzzer x1
- Crowtail-9G Servo x1
- Crowtail-Cable x4
- USB cable x1

Hardware learning and connection

The circled position in the picture is the accelerometer on the micro: bit. We can use this accelerometer to measure micro: bit acceleration and detect micro: bit movement. It can also detect other micro: bit gestures, such as shaking, tilting, and free fall.



Connect Crowtail-Touch Sensor and Crowtail-Switch to P0 and P1 ports of Crowtail-Base shield for Micro:bit(named bit1). Connect Crowtail-Buzzer, Crowtail-LED and Crowtail-9G Servo to P0, P1 and P2 ports of Crowtail-Base shield for Micro:bit(named bit2). **Note: You need to connect 12V power supply for bit2.** The hardware connection are as follows:



Programming and note

Send_code(bit1):

```
on start
  radio set group 10

on logo down
  radio send number 1

on tilt left
  radio send number 2

on tilt right
  radio send number 3

forever
  if digital read pin P0 = 1 then
    radio send number 4
  if digital read pin P1 = 1 then
    radio send number 5
  else
    radio send number 6
```

Receiver_code(bit2):

```
on start
  radio set group 10
  digital write pin P1 to 0

on radio received receivedNumber
  if receivedNumber = 1 then
    servo write pin P2 to 90
    show leds
  else if receivedNumber = 2 then
    servo write pin P2 to 0
    show leds
  else if receivedNumber = 3 then
    servo write pin P2 to 180
    show leds
  else if receivedNumber = 4 then
    start melody ba ding repeating once in background
  else if receivedNumber = 5 then
    digital write pin P1 to 1
  else if receivedNumber = 6 then
    digital write pin P1 to 0
```

- **On Gesture:** On logo down, on tilt left and on tilt right these three blocks are all called On gesture block. This type of block will start an event handler(part of the program that will run when something happens). This handler works when you do a gesture, for example, shaking the micro:bit. In send_code, when bit1 logo down, we send 1; when bit1 tilts to the left, we send 2; when bit1 tilts to the right, we send 3.
- **Show leds:** This block is to shows a picture on the LED screen. There are 4 * 4 LEDs, you can control any one of the LEDs above to draw your own picture. Here the picture we need to draw is 3 direction arrows.
- **If...else if...:** We use this statement to run the corresponding code based on different numbers received by radio.

Blocks Overview

Send_code(bit1):

1. Set radio group for micro:bit.
2. Send corresponding code based on different gesture detected.
3. If the touch sensor is touched(1), send number 4.
4. If the switch is pressed(1), send number 5.
5. If the switch pops up(0), send number 6.

Receiver_code(bit2):

1. Set radio group of micro:bit.
2. If the number received is 1, the servo shaft rotates to 90 degrees and a forward picture is displayed on the micro:bit matrix LEDs.
3. If the number received is 2, the servo shaft rotates to 0 degrees and a left picture is displayed on the micro:bit matrix LEDs.
4. If the number received is 3, the servo shaft rotates to 180 degrees and a right picture is displayed on the micro:bit matrix LEDs.
5. If the number received is 4, play a melody in background.
6. If the number received is 5, turn LED on.
7. If the number received is 6, turn LED off.

- Download the `send_code` program to bit1 and `receiver_code` program to bit2 and see what happens.

Result

When the bit1 logo tilts down, the matrix LED on bit2 displays the forward picture and the servo rotates to 90 degrees; when bit1 tilts to the left, the matrix LED on bit2 displays the left picture and the servo rotates to 0 degrees; when bit1 tilts to the right, the matrix LED on bit2 displays the right picture and the servo rotates to 180 degrees; when you press the touch sensor connected to bit1, the buzzer connected to bit2 will play the melody. When you press the switch connected to bit1, the LED connected to bit2 will light up. When you pop the switch, the LED will go out.



Try to use what you have learned in this project to make a real remote control car!

