

# Investigating the world around us

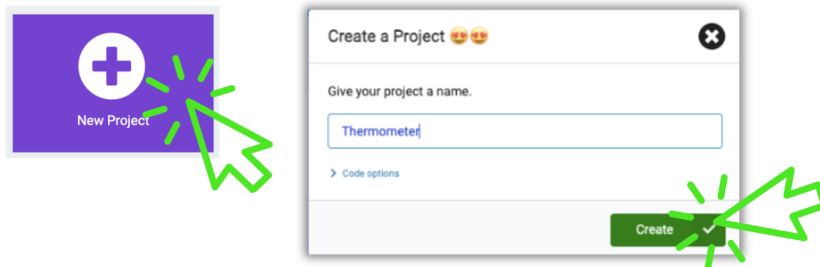
Connect the micro:bit to your computer.

In this project we are using the micro:bit's built in temperature sensor, microphone and light sensor. The thermometer is in the micro:bit's processor or CPU (central processing unit) to measure how hot or cold it is. The sound sensor is the microphone and the LEDs are used as light sensors

Let's get coding!

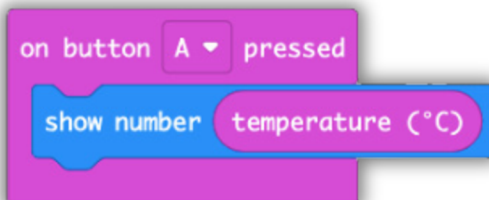
<https://makecode.microbit.org/>

Click on 'New Project' and give your project a name.

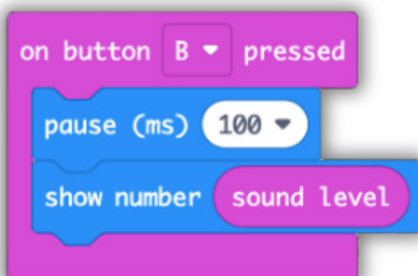


Processor / CPU

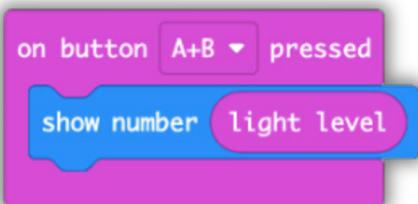
Add the following code:



When you press button A, the temperature will show on the LED display.



When you press button B, the sound level will display. The 'pause' block ensures that the results are not skewed by the sound made by clicking the button.



When you press button A and B together, the micro:bit will display the light level.

Connect your micro:bit to the computer using the USB cable and click 'download'. Test it out.

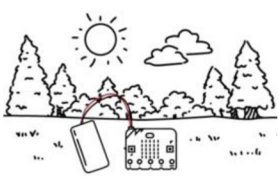

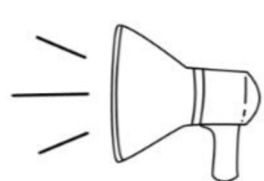
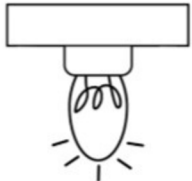
Decide on your locations, then, attach your battery pack and take your micro:bit outside!

# Investigating the world around us



## DATA RECORDING SHEET

Complete the following table using your micro:bit to collect data from different locations.

			
LOCATION / TIME	TEMPERATURE (BUTTON A)	SOUND LEVEL (BUTTON B)	LIGHT LEVEL (BUTTONS A & B)



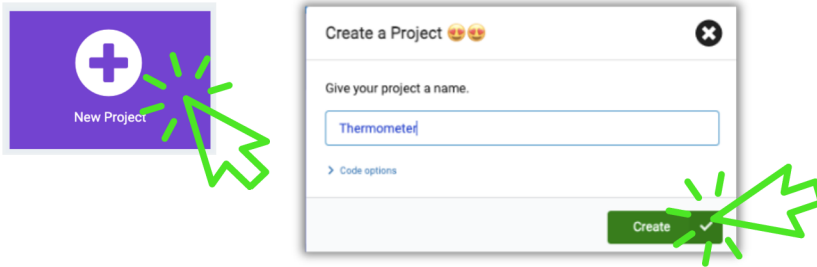
# Investigating the world around us

## Level up - thermometer

Here is a more advanced thermometer program that will display a bar graph and also give you instructions for what to do in different temperature conditions!

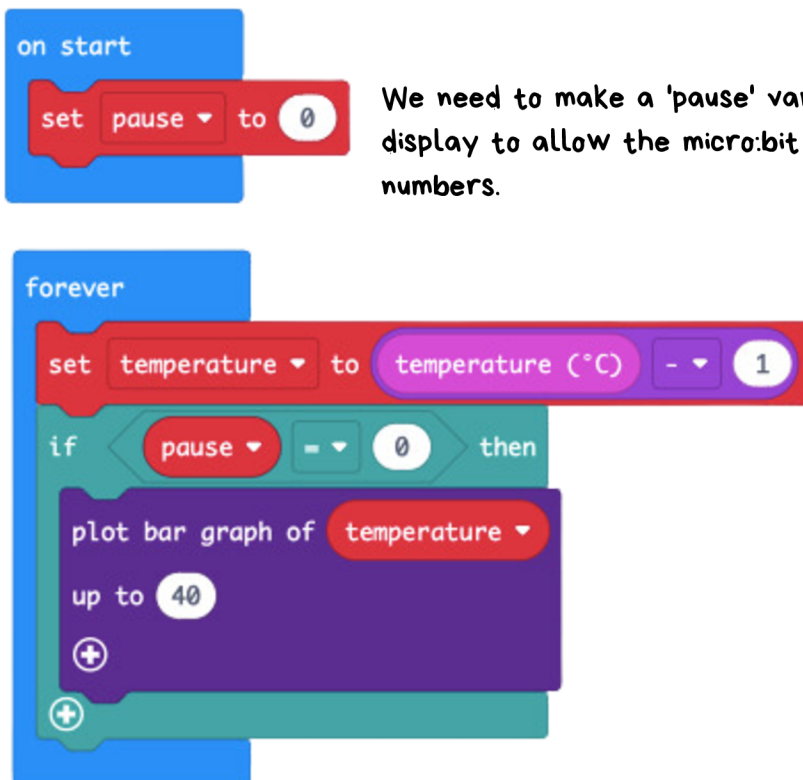
<https://makecode.microbit.org/>

Start a new project and give it a name.



If you have a thermometer, take a reading and compare it to the reading you get from your micro:bit. The longer your micro:bit has been running, the hotter it will be. Mine was 1° higher than room temperature so I added a maths block to subtract 1° to give a more accurate reading.

Add the following code:



We need to make a 'pause' variable so that we can pause the graph display to allow the micro:bit to tell us what the temperature is in numbers.

The processor's temperature is a fairly good approximation of the temperature around you in °C (Celsius) but it might get a bit warmer after it's been used for a few hours!

Continues on the next page...



## Investigating the world around us cont.

Continue adding code to your thermometer.

```

on button A pressed
  set pause to 1
  clear screen
  if temperature ≥ 25 then
    show string "Hot! Shorts"
  else if temperature ≥ 20 then
    show string "Nice weather. T shirt"
  else if temperature < 20 then
    show string "Brrr, cold! Warm jumper"
  set pause to 0
  
```

```

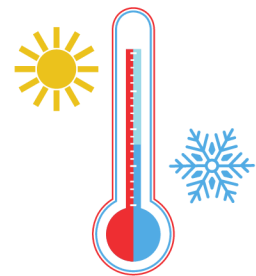
on button B pressed
  set pause to 1
  clear screen
  show string join temperature "°C"
  set pause to 0
  
```

Decide what temperature you consider 'hot', 'just right' and 'cold' and what message you would like to display! I've chosen the clothes I would like to wear.

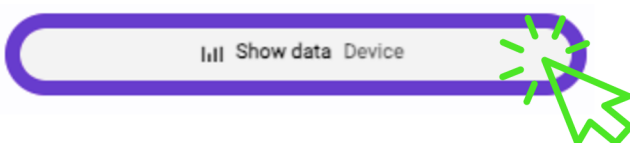
Add this code so that you can monitor the temperature data over time.

```

forever
  serial write value temperature - temperature
  
```



Click on 'Show data' in the MakeCode editor to see real-time data for the temperature. You can track how your environment changes over time.



How about monitoring and comparing the temperature in light and share on different days of the week.

## Investigating the world around us cont.

### Level up - thermometer 2

Here is an alternative thermometer that keeps track of the lowest and highest temperatures recorded by using 3 variables.

```

on start
  set current-temperature to temperature (°C)
  set max to current-temperature
  set min to current-temperature
  
```

The 'current-temperature variable' is set to the current temperature reading, max is set to the maximum and min is set to the minimum temperature reading.

```

forever
  show string .
  set current-temperature to temperature (°C)
  if current-temperature < min then
    set min to current-temperature
  if current-temperature > max then
    set max to current-temperature
  pause (ms) 1000
  clear screen
  pause (ms) 1000
  
```

This infinite (forever) loop ensures that every two seconds it takes a new reading. It also flashes a dot on the LED display so that you know it's working.

If the current temperature is less than (<) than the value stored in the min variable, it changes the min variable to be the same as the current temperature.

If the current temperature is greater than (>) the max variable's value, it changes the max variable to be the same as the current temperature.

How about monitoring the temperature throughout the day. If you have more than one micro:bit, you can compare different locations at the same time.

