

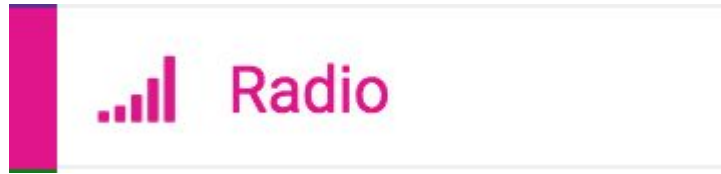
# Micro:bit

Radio, Sensors, & Lights

Grab a Computer, a Microbit and a Cable then go to

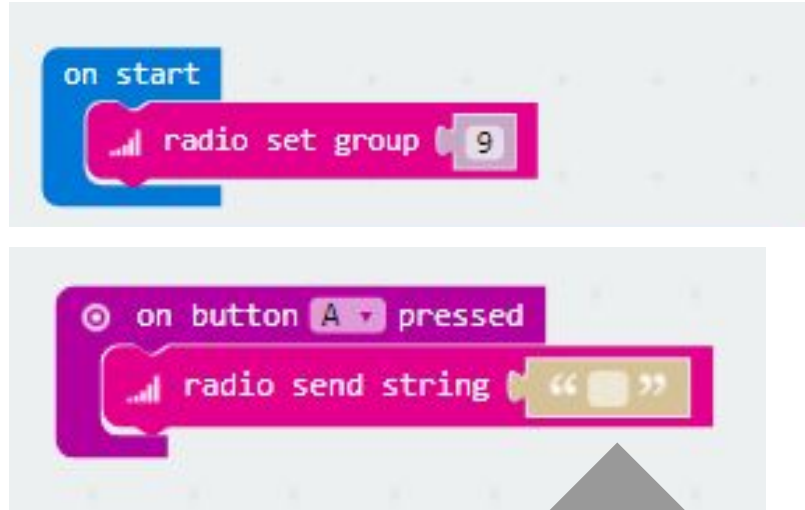
[makecode.microbit.org](https://makecode.microbit.org)

Day 1



# Radio: Challenge #1

## My Code



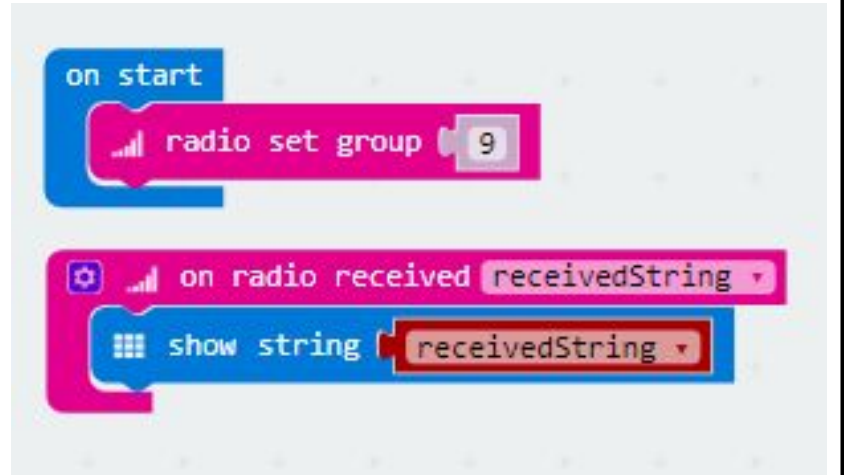
```
on start
  radio set group 9

on button A pressed
  radio send string ""
```

The code consists of two blocks. The first block is an 'on start' block containing a 'radio set group' block with the value '9'. The second block is an 'on button A pressed' block containing a 'radio send string' block with an empty string '""'. A grey arrow points upwards from the text 'Secret word here' to the 'radio send string' block.

**Secret word here**

## Your Code



```
on start
  radio set group 9

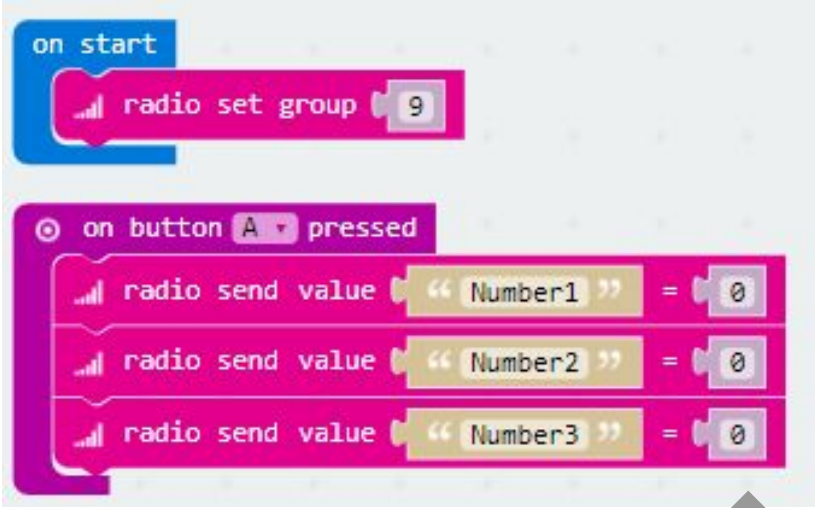
on radio received receivedString
  show string receivedString
```

The code consists of two blocks. The first block is an 'on start' block containing a 'radio set group' block with the value '9'. The second block is an 'on radio received receivedString' block containing a 'show string' block with the parameter 'receivedString'.

Don't forget to flash the micro:bit after downloading your code!

# Radio Challenge #2 - unlock the combination

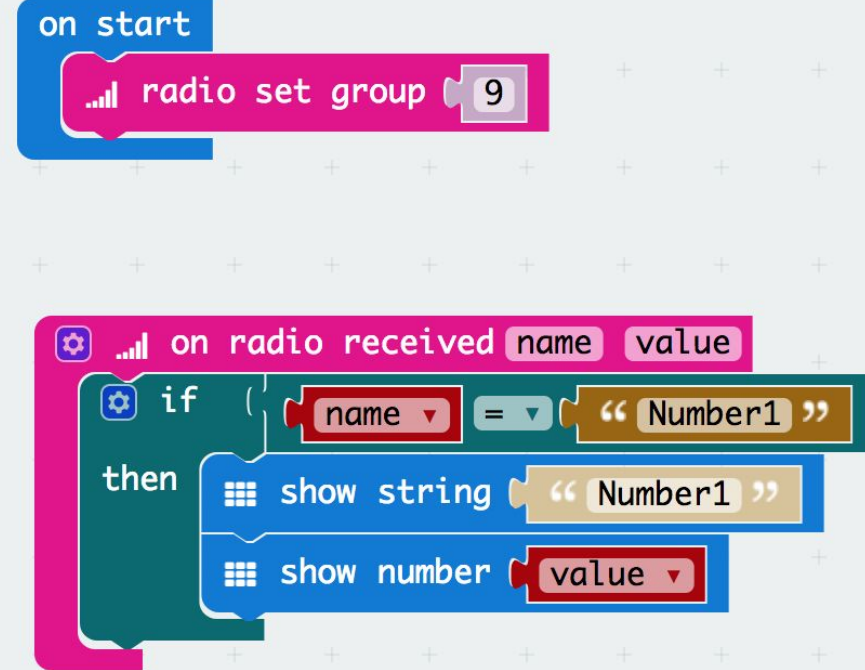
My Code



```
on start
  radio set group 9

on button A pressed
  radio send value "Number1" = 0
  radio send value "Number2" = 0
  radio send value "Number3" = 0
```

Your Code

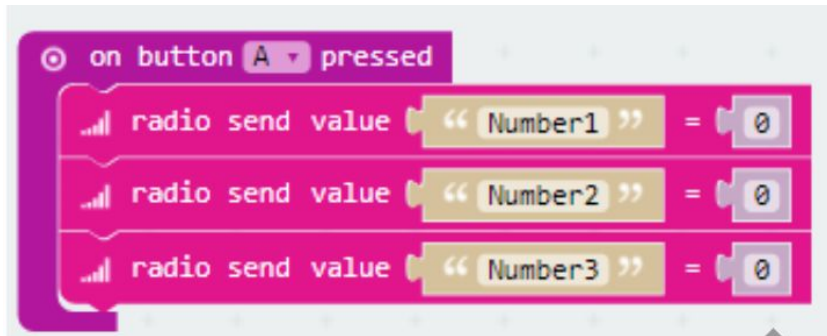


```
on start
  radio set group 9

on radio received name value
  if (name = "Number1")
    then
      show string "Number1"
      show number value
```

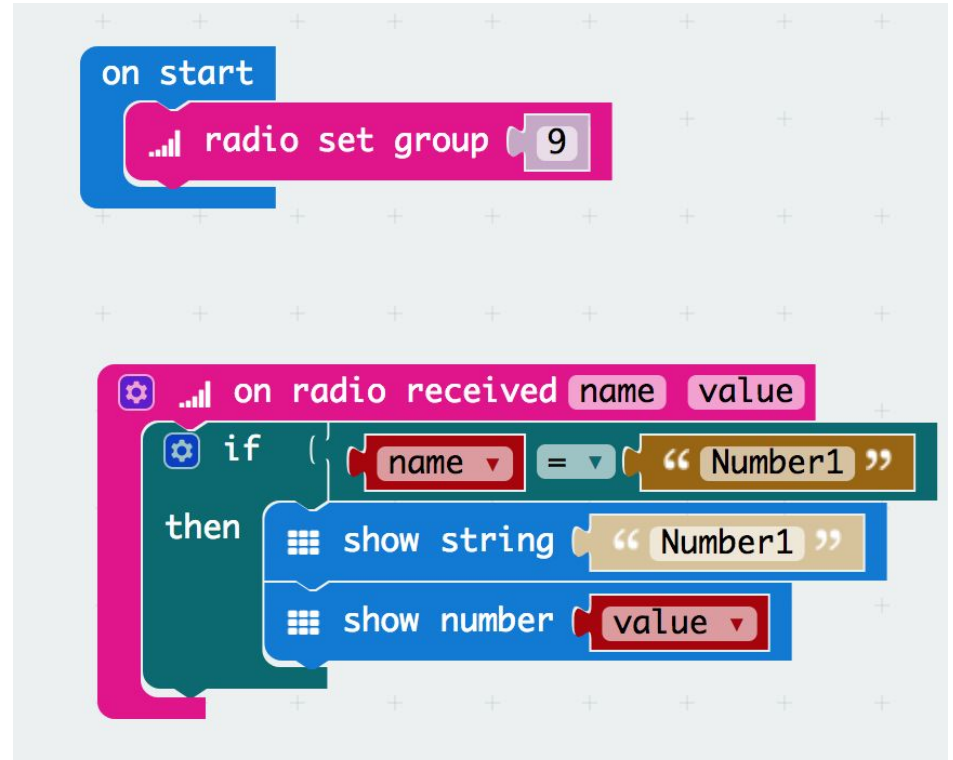
Secret numbers here

Change your code to find out what the 2nd and 3rd numbers are



```
on button A pressed
  radio send value "Number1" = 0
  radio send value "Number2" = 0
  radio send value "Number3" = 0
```

The image shows a Scratch script starting with 'on button A pressed'. It contains three 'radio send value' blocks. Each block has a radio icon, the text 'radio send value', a speech bubble containing the string 'Number1', 'Number2', or 'Number3', an equals sign, and a numeric input field containing the number 0.



```
on start
  radio set group 9

on radio received name value
  if (name = "Number1")
  then
    show string "Number1"
    show number value
```

The image shows a Scratch script starting with 'on start' containing a 'radio set group' block with the value 9. Below it is an 'on radio received' block with 'name' and 'value' variables. Inside this block is an 'if' block with the condition 'name = "Number1"'. The 'then' part of the if block contains two blocks: 'show string' with the value 'Number1' and 'show number' with the value 'value'.

# Find all three numbers at once

## My Code

```
on start
  radio set group 9

on button A pressed
  radio send value "Number1" = 0
  radio send value "Number2" = 0
  radio send value "Number3" = 0
```

The code in the 'My Code' block consists of two event-driven blocks. The first is an 'on start' block containing a 'radio set group' block with the value '9'. The second is an 'on button A pressed' block containing three 'radio send value' blocks. Each 'radio send value' block has a text input field containing a string ('Number1', 'Number2', or 'Number3') and a numeric input field containing the value '0'.

**Secret numbers here**

## Your Code

```
on start
  radio set group 9

on radio received name value
  if (name = "Number1")
  then
  else if
  then
  else
```

The code in the 'Your Code' block consists of two event-driven blocks. The first is an 'on start' block containing a 'radio set group' block with the value '9'. The second is an 'on radio received' block with two dropdown menus, 'name' and 'value'. Inside this block is an 'if' block with a condition 'name = "Number1"'. The 'if' block has three sub-blocks: 'then', 'else if', and 'else', all of which are currently empty.

# Partner Radio Challenge

- 1) Set up a radio group with a partner.
- 2) When you send a message to your partner, their micro:bit should show the message.
- 3) When your partner presses A on their micro:bit, your microbit should make a smiley face.

# Share your code on google doc/assignment

Make a table in your document

Date	Code link	Short description

SHARE your code link on your google doc/assignment with the date and a description.



My Code:

```
on start
  radio set group 9

forever
  radio send value "light" = light level
  pause (ms) 2000
  radio send value "temp" = temperature (°C)
  pause (ms) 2000
```

## Your Challenge:

Choose to show either a light reading or a temperature reading.

For light, show “LIGHT” and then the light level.

For temperature, show “TEMP” and then the temperature.

Show a teacher when completed.

Day 2

Each lantern group needs: a Computer, a light strand  
Each ecosystem group needs: a Weather:bit, Micro:bit,  
and a Cable;  
then go to

[makecode.microbit.org](https://makecode.microbit.org)

and

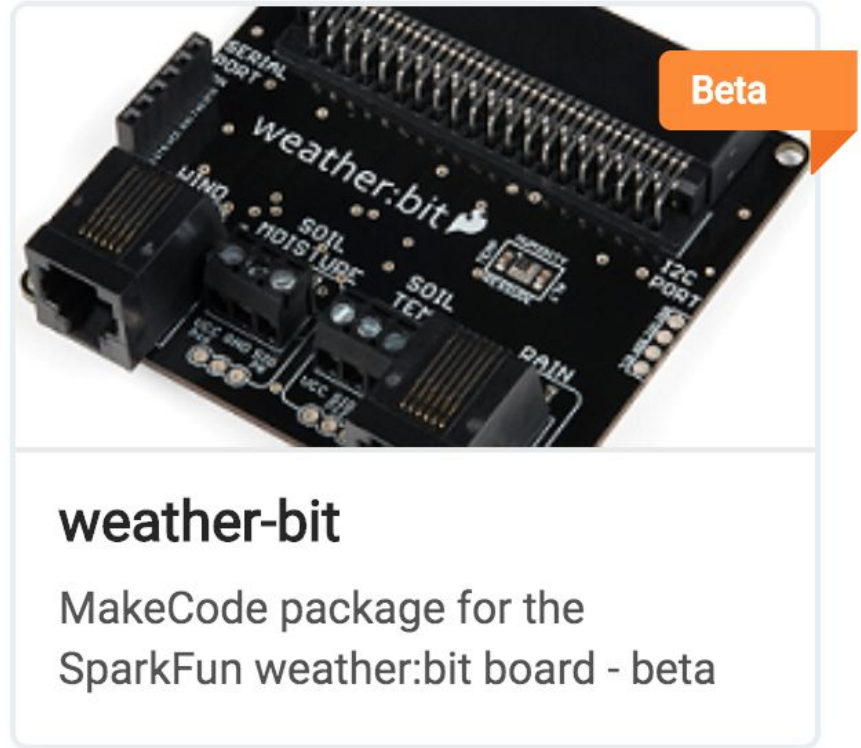
Open your code storage  
document

# Weather:bit

For the project you will get one per ecosystem.

You will need to share in your lantern groups if you have 2 lanterns per ecosystem.

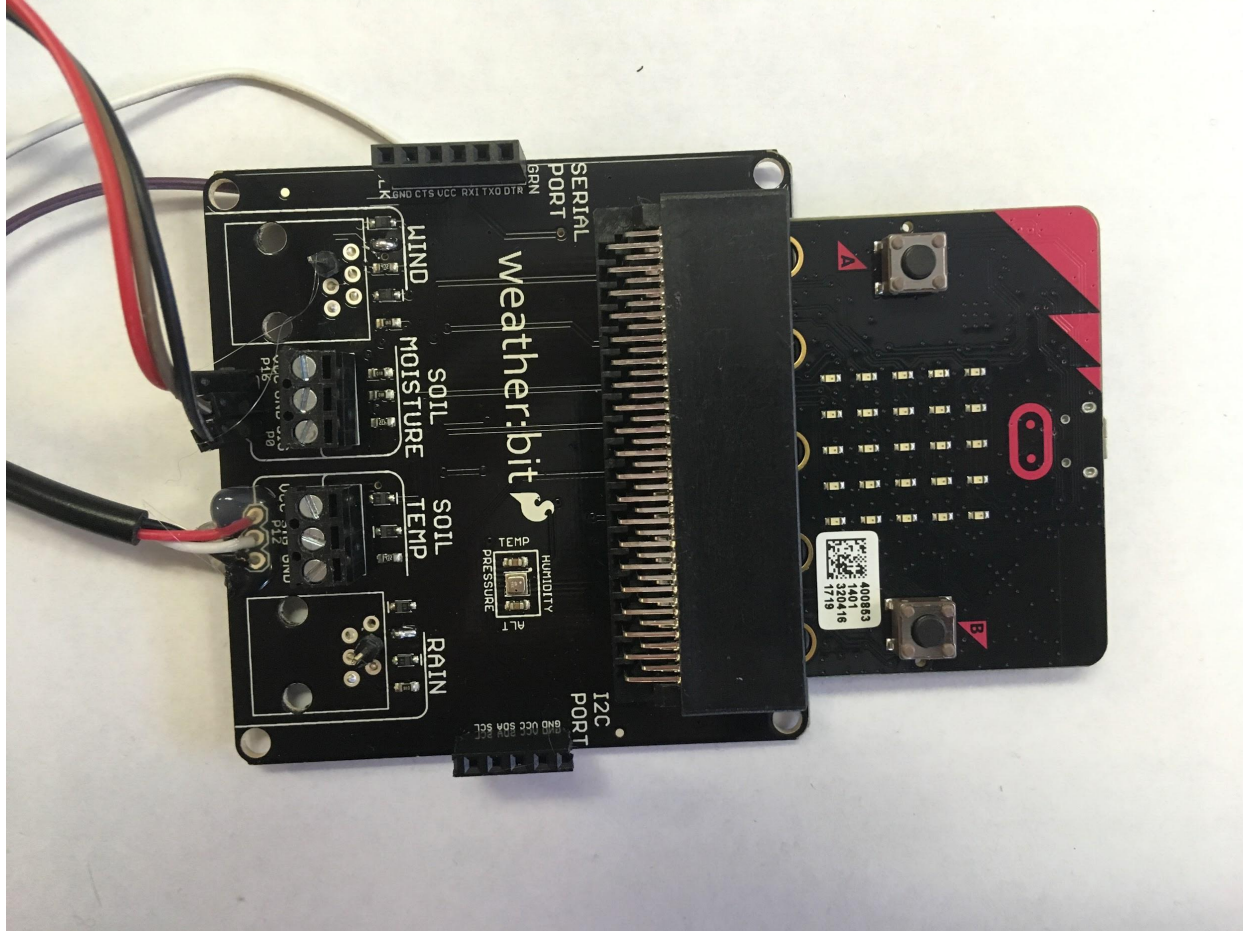
Can program separately but we will show you how to combine the code together later.



## weather-bit

MakeCode package for the SparkFun weather:bit board - beta

Slide the micro:bit into the weather:bit



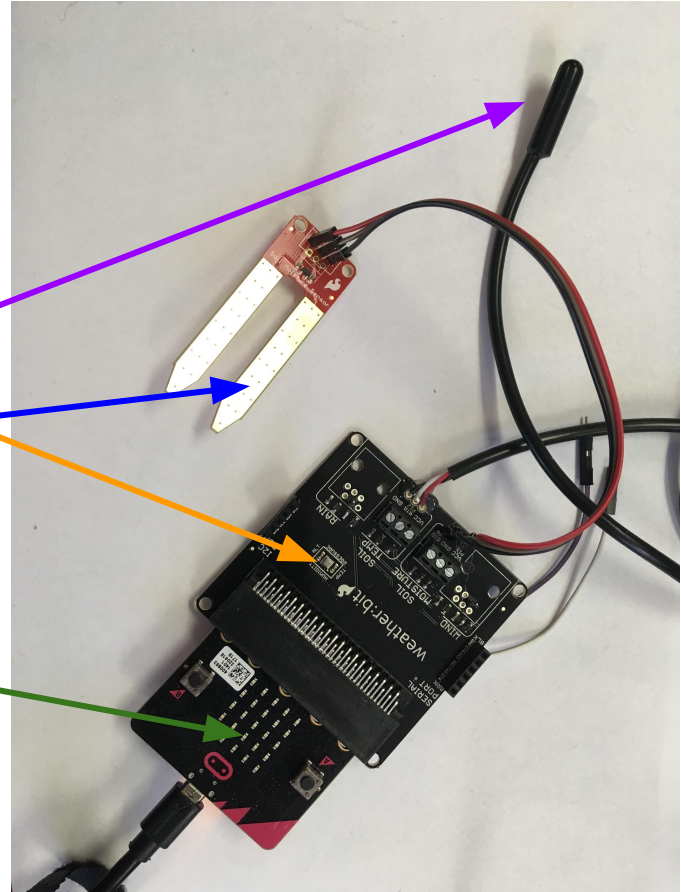
# Find the following sensors

- Air temperature
- Humidity
- Pressure
- Soil OR water temperature
- Soil moisture
- Light?

**\*\* This setup is NOT waterproof!**

# Find the following sensors

- Air temperature
- Humidity
- Pressure
- Soil OR water temperature
- Soil moisture
- Light



**\*\* This setup is NOT waterproof!**

# The weather:bit on makecode

The screenshot displays the MakeCode IDE interface. On the left is the Microsoft menu with options: Project Settings, Add Package..., Delete Project, Language, High Contrast On, Reset, Privacy & Cookies, Terms Of Use, About..., and Give Feedback. An orange arrow points from the 'Add Package...' option to the 'Add Package... ?' dialog box. The dialog box has a search bar containing 'weatherbit'. Below the search bar are two categories: 'devices' (Camera, remote control and other Bluetooth services) and 'bluetooth' (Bluetooth services). An orange arrow points from the 'devices' category to the 'weather-bit' package card. The package card features an image of the weather:bit board, a 'Beta' badge, and the text: 'weather-bit', 'MakeCode package for the SparkFun weather:bit board - beta'. A final orange arrow points from the package card to a code block in the workspace. The code block contains the following code:

```
on start
  start weather monitoring
```



# Weather:bit challenge

Show the two data values on your micro:bit screen for your lantern.

Be sure to indicate which is which value.

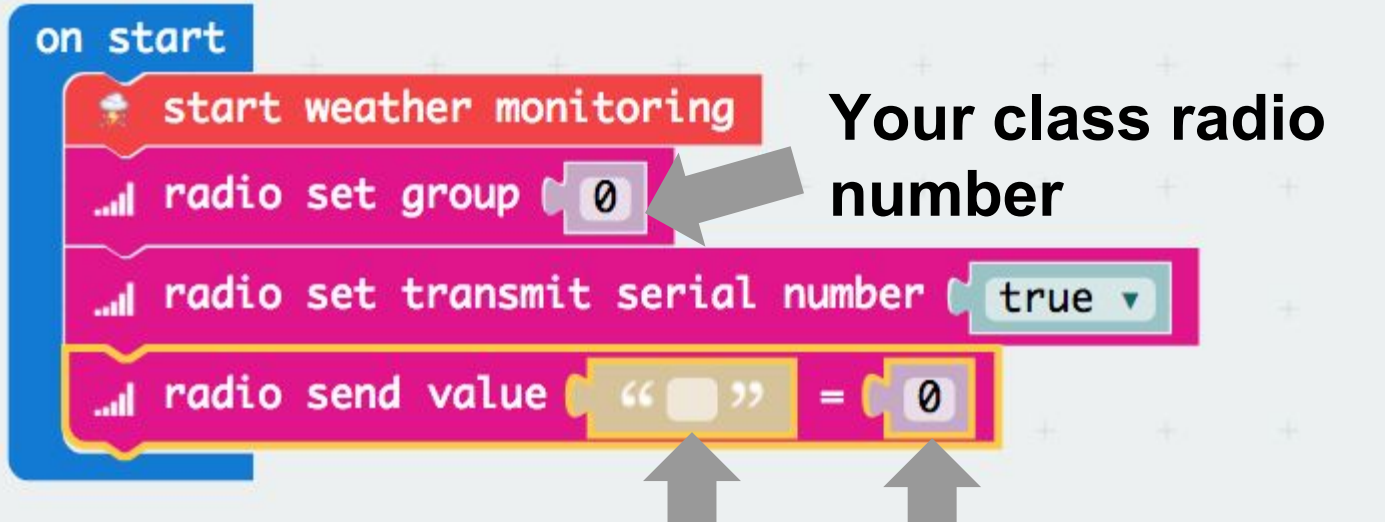
-----

Challenge: Can you find a way to change the value of this data?

Hint: Think about the environment the sensor is in.



# Weather:bit hourly readings for ecosystems



The image shows a Scratch code editor with four blocks under an 'on start' event:

- start weather monitoring** (red block)
- radio set group** (pink block) with a numeric input field containing '0'. A grey arrow points from the text 'Your class radio number' to this field.
- radio set transmit serial number** (pink block) with a dropdown menu set to 'true'.
- radio send value** (pink block) with a text input field containing a grey box and a numeric input field containing '0'. A grey arrow points from the text 'Your group's secret number' to this field.

Below the code, two grey arrows point upwards to the text labels: 'Your group name (unique, short)' points to the text input field in the 'radio send value' block, and 'Your group's secret number' points to the numeric input field in the 'radio send value' block.

**Your class radio number**

**Your group name  
(unique, short)**

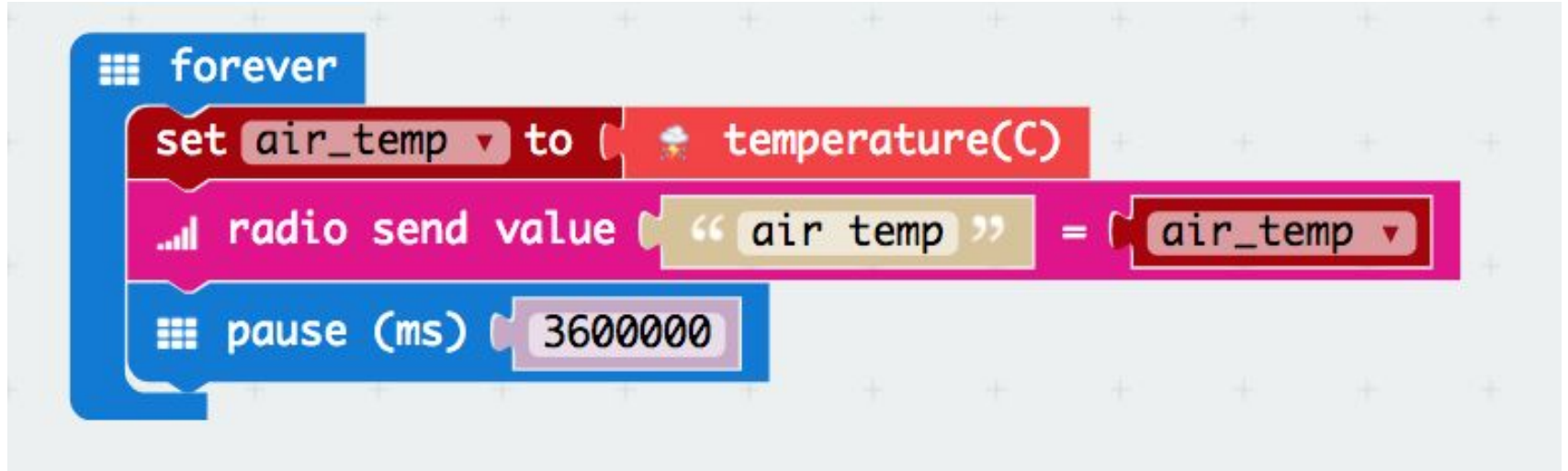
**Your group's  
secret number**

Make this code

Download the program and flash the micro:bit.

Check in with the teacher.

# Weather:bit hourly readings for ecosystems



```
forever
  set air_temperature to temperature(C)
  radio send value "air temp" = air_temperature
  pause (ms) 2000
  set soil_moisture to soil moisture
  radio send value "moisture" = soil_moisture
  pause (ms) 3600000
```

Repeat the first two lines for all other data with 2 sec pauses:

- Humidity
- Pressure
- Soil (or water) temperature
- Soil moisture
- Light

# Share your code on google doc/assignment

Choose one person to be the recorder for your lantern group.

That person should **SHARE** the code link in thier doc/assignment with the date, a description of what the code does **AND** includes who is in the group.

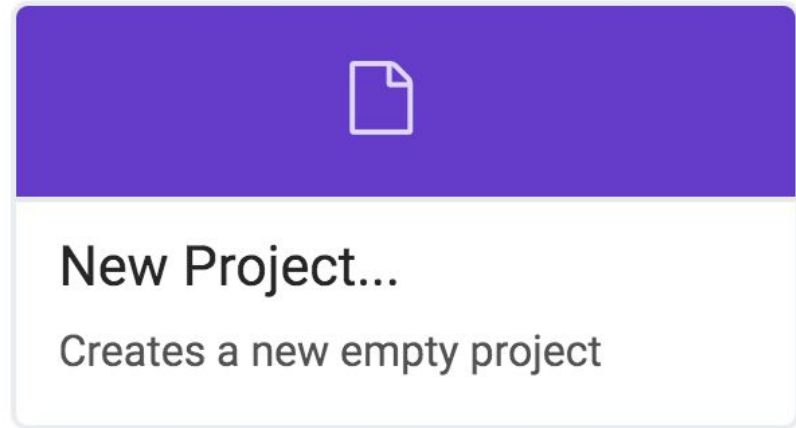
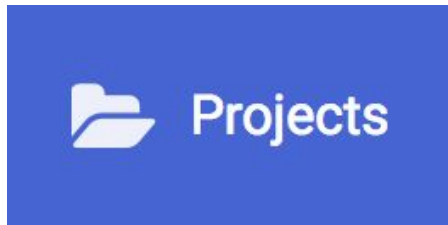
Date	Code link	Short description

## Day 3

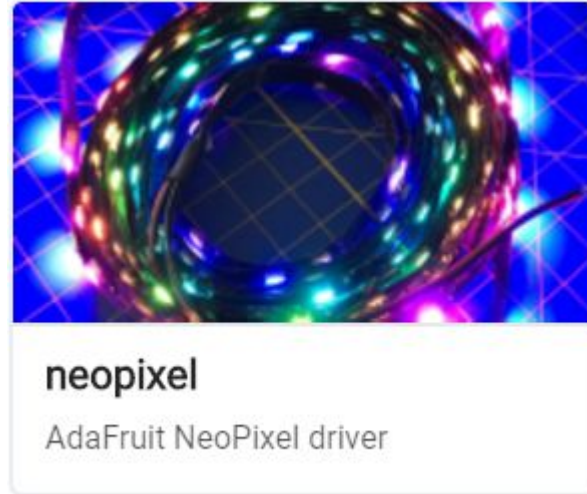
Grab a Computer per Lantern, a Microbit and a  
Cable then go to

[makecode.microbit.org](https://makecode.microbit.org)

# Make a new project



# Download the neopixel package





# To setup the neopixels

on start

set neopixels to  NeoPixel at pin P0 with 8 leds as RGB (GRB format)

What does this do to the simulator?

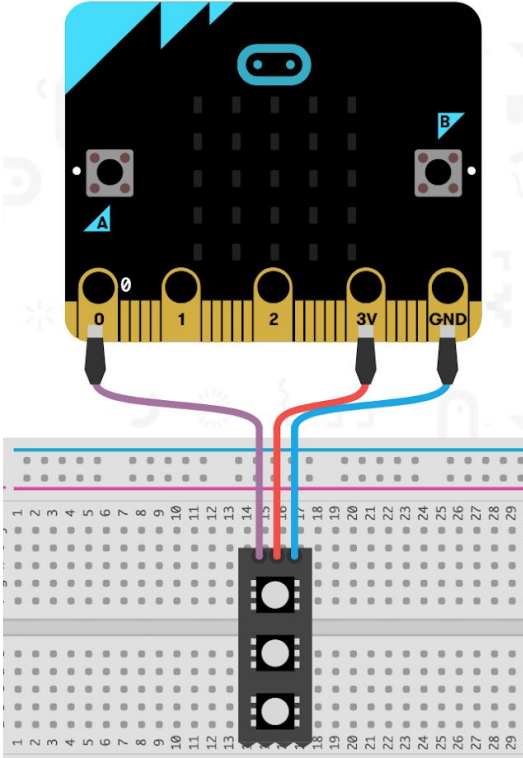
What do you think P0 is?

What does the 8 mean in this code?

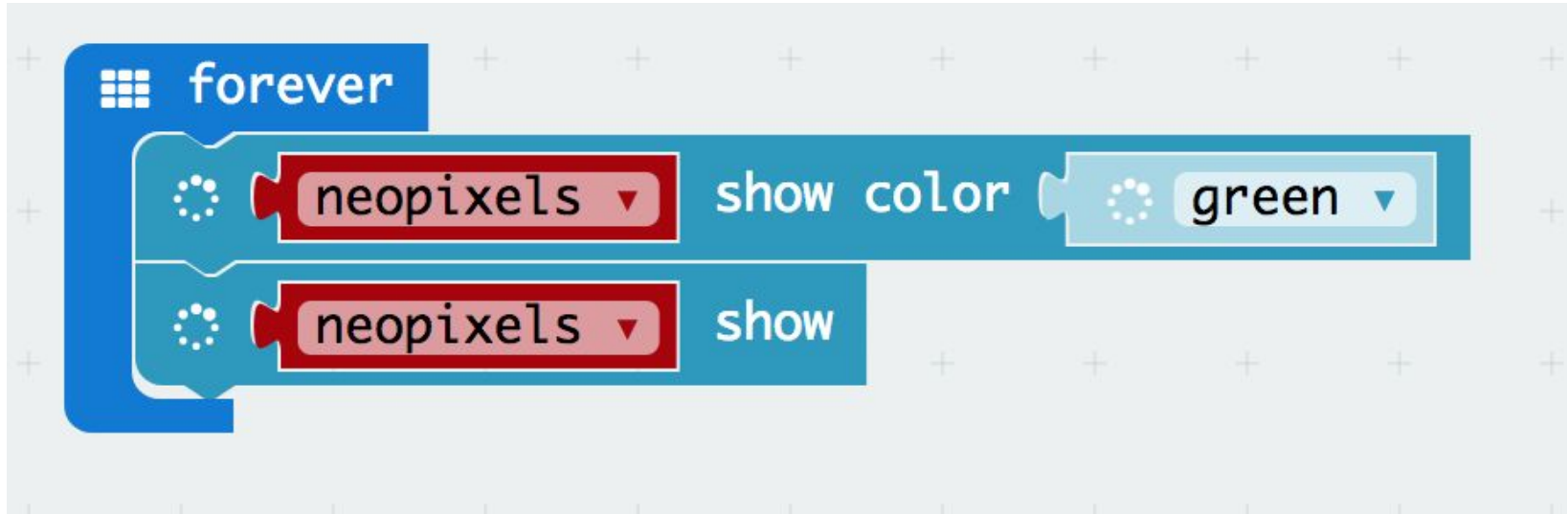
What is “neopixels” in this code?

on start

set neopixels to NeoPixel at pin P0 with 8 leds as RGB (GRB format)



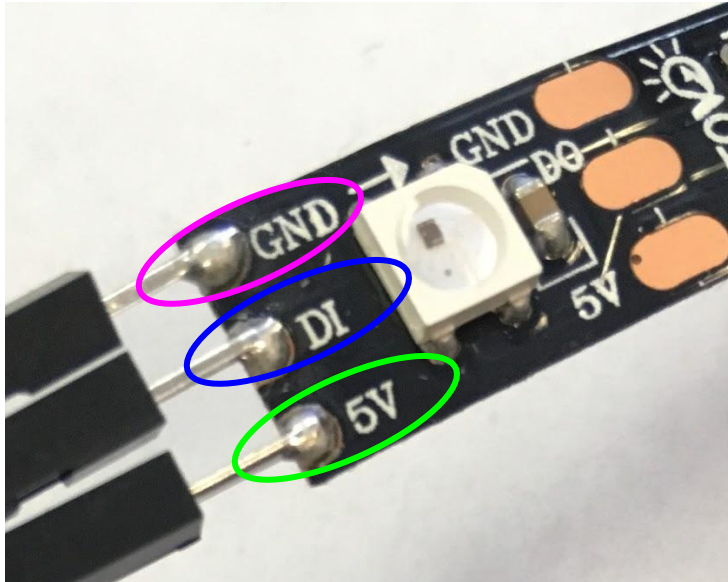
# Example:



**5 minutes: play with coding the micro:bit simulator using the neopixel package. See what you can learn!**

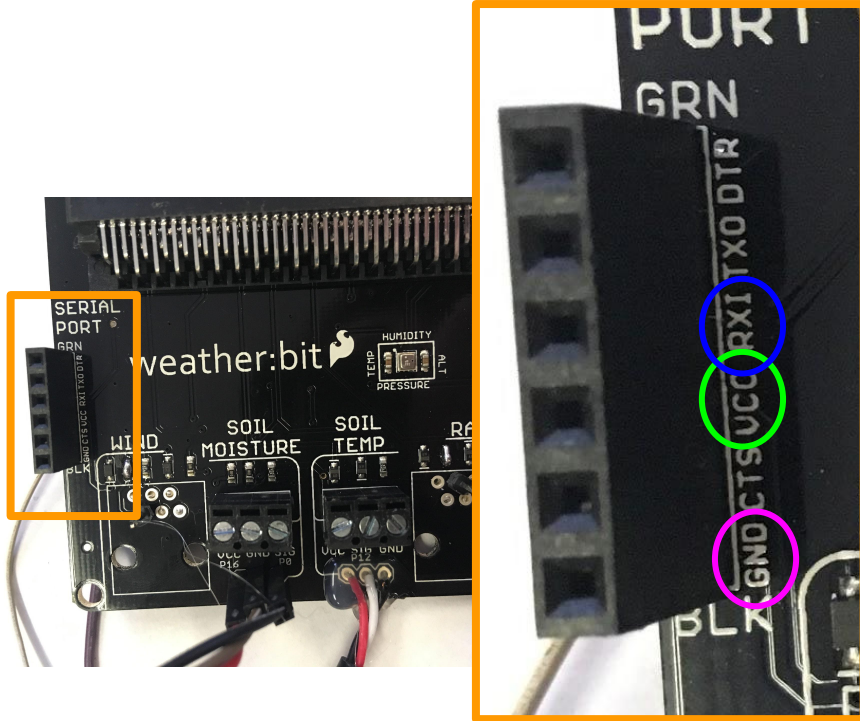
Unplug the micro:bit from your computer

# How to setup w:b, m:b, and neopixels



DI (LEDs) → signal  
GND (LEDs) → ground  
5V (LEDs) → power

# How to setup w:b, m:b, and neopixels



**RX1** (w:b) → signal  
**GND** (w:b) → ground  
**VCC** (w:b) → power

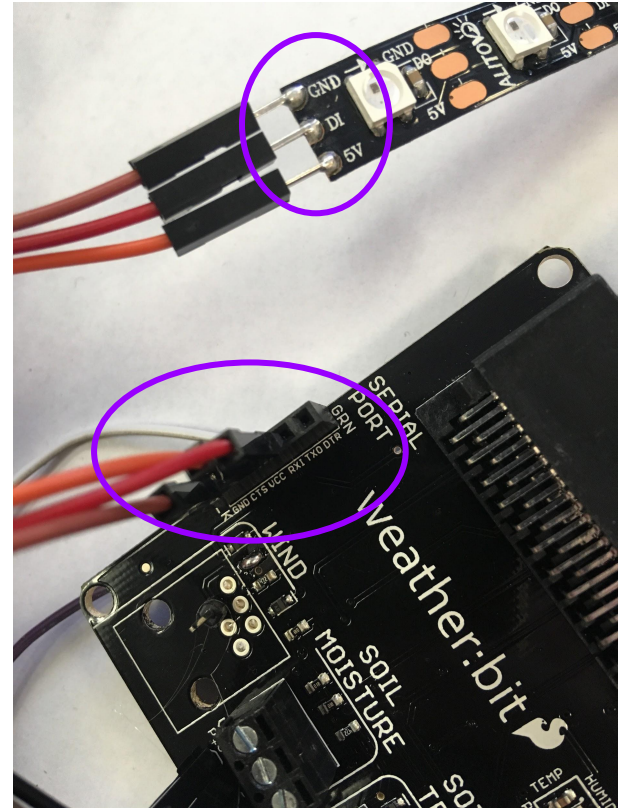
# How to setup w:b, m:b, and neopixels - 1st set of lights

LEDs → w:b

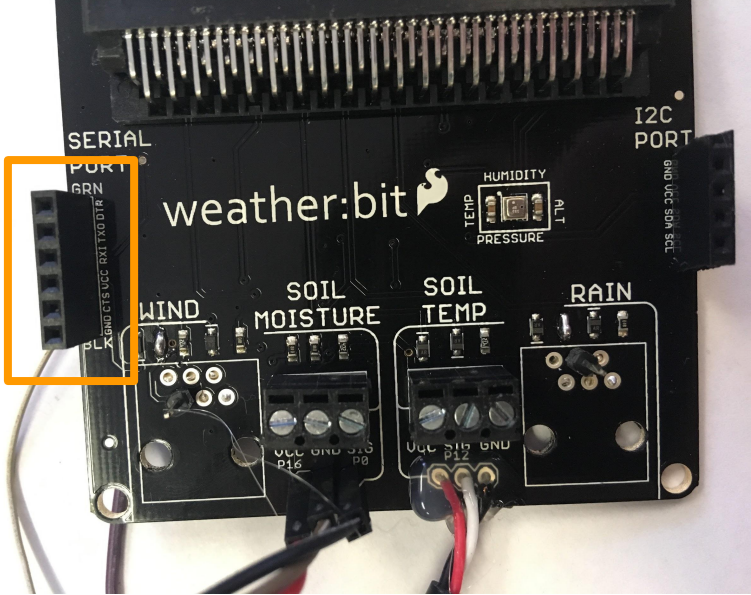
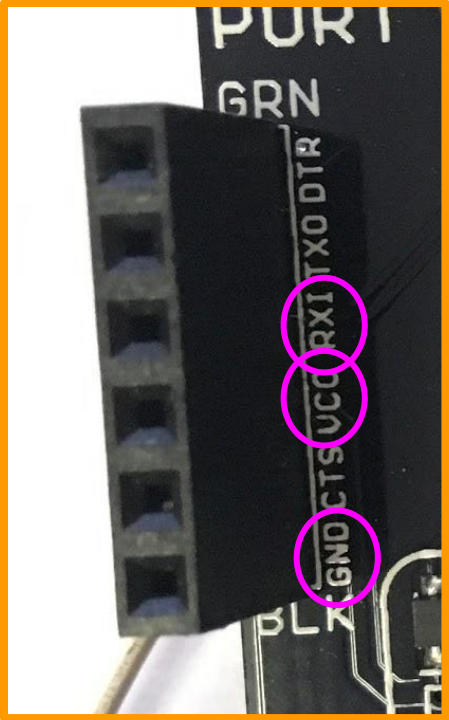
DI (LEDs) → RX1 (w:b)

GND (LEDs) → GND (w:b)

5V (LEDs) → VCC (w:b)

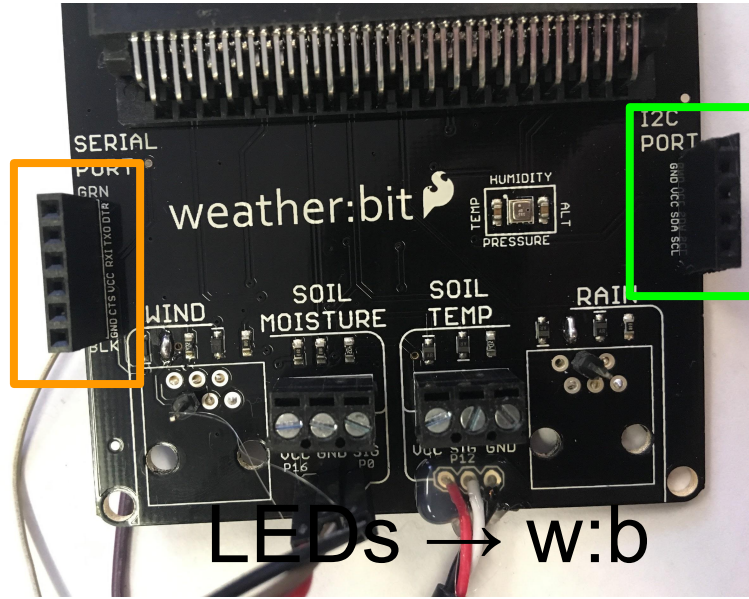
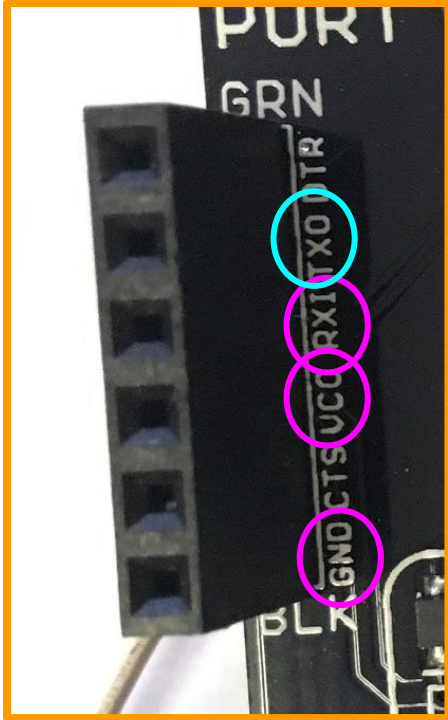


# Two lanterns per weather:bit?





## 2nd set of lights



DI (LEDs) → TXO (w:b)  
GND (LEDs) → GND (w:b)  
5V (LEDs) → VCC (w:b)

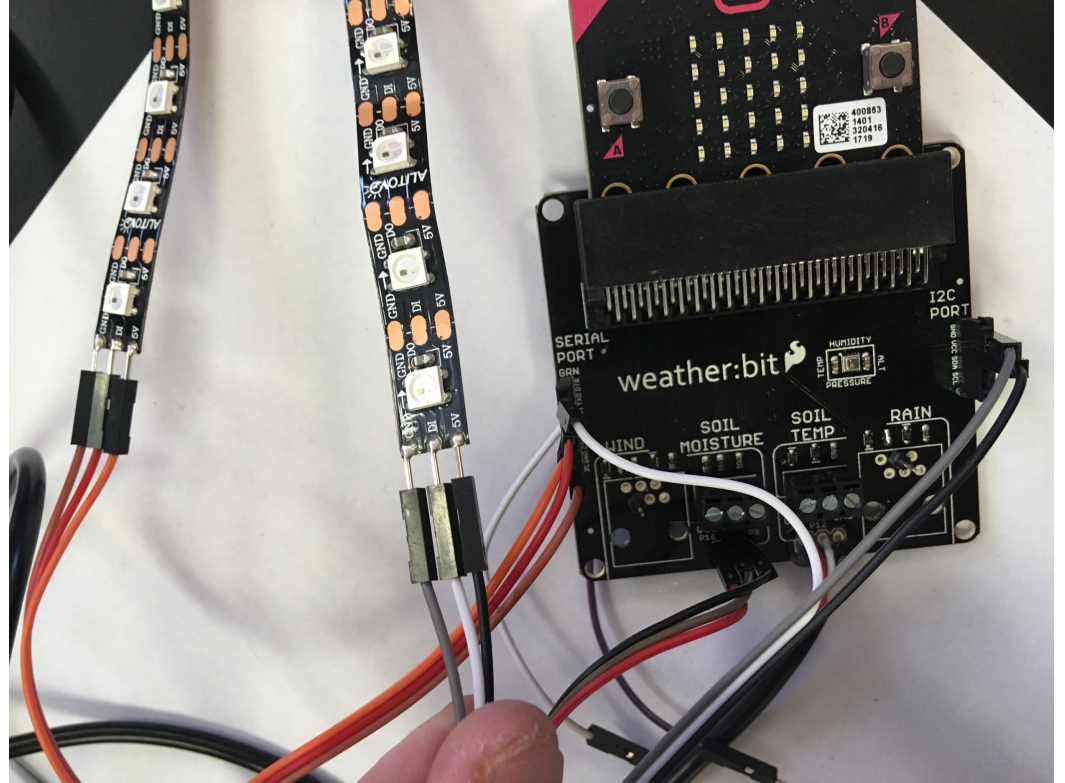
# Two lanterns per weather:bit?

RX1 is Pin P14

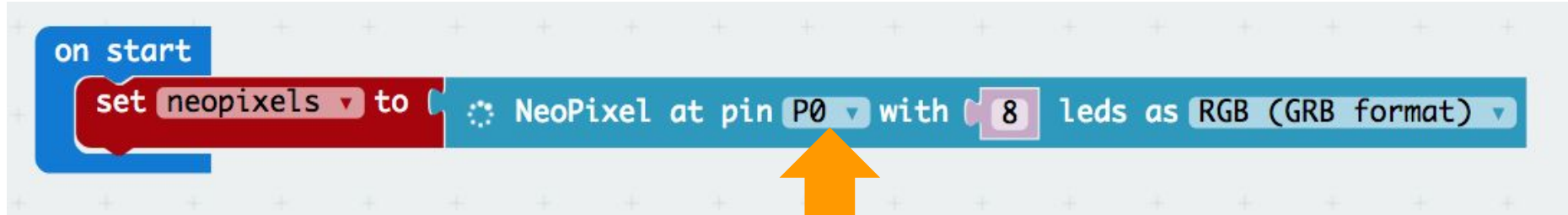
TXO is Pin P15

-----

Choose which half of the group will use Pin P14 (RXI) and which will use Pin P15 (TXO)



Make sure the signal pin is correct



RXI on weather:bit is Pin P14 on micro:bit

TXO on weather:bit is Pin P15 on micro:bit

Test your LEDs with your code!

Click Download and flash your micro:bit.

Share the weather:bit in your group.

If you didn't get a code to work you can use this one to test:

<http://bit.ly/2CDVKh3>

Rename your program

Then **SHARE** the link in your google assignment/doc  
(don't forget to add who is in your group)

# To combine your neopixel codes

You should have 2 neopixel codes - one for each half of your group (P14 and P15).

- Choose one person to be the overall recorder for your ecosystem group.
- Share the links for both codes with that person.
- The recorder should open both codes in separate tabs on their computer (click the link and then click edit).

**\*\*IF** you are making only a single lantern then open the w:b code and neopixel code in two different tabs.

# Combine your codes into one program

Click  on both programs.

Copy any different lines on the Pin 14 neopixel code.

Paste them into the Pin 15 code.

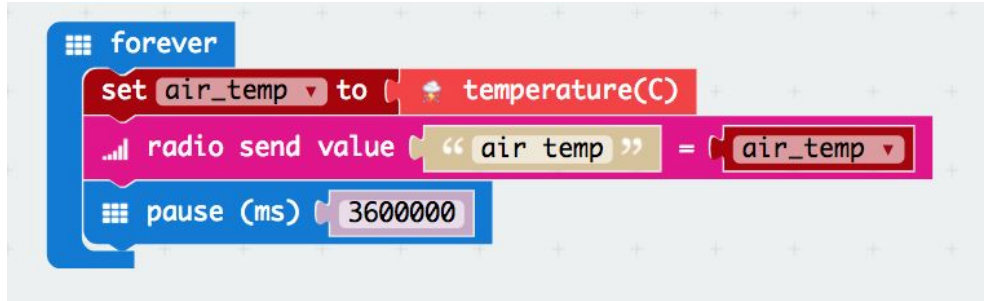
Click  and make sure everything looks right.

Rename, test and share this code.



# Combine your w:b and neopixel codes into one program

Tuesday you made a weather:bit code. Open that (use your link). Looks something like this program.



```
forever loop
  set air_temp to temperature(C)
  radio send value "air temp" = air_temp
  pause (ms) 3600000
```

The image shows a Scratch code block for a weather:bit program. It is a 'forever' loop containing three blocks: 'set air\_temp to temperature(C)', 'radio send value "air temp" = air\_temp', and 'pause (ms) 3600000'.

Tuesday you also made a neopixel code - combined P14 and P15 codes. Open that in a different tab. It started like this.



```
on start
  set neopixels to NeoPixel at pin P0 with 8 leds as RGB (GRB format)
```

The image shows a Scratch code block for a neopixel program. It is an 'on start' block containing one block: 'set neopixels to NeoPixel at pin P0 with 8 leds as RGB (GRB format)'.

# Combine your codes into one program

Click

A blue rectangular button with a white curly brace icon on the left and the text "JavaScript" in white on the right.

on both programs.

Copy any different lines on the w:b code.

Paste them into the neopixel code.

Click

A blue rectangular button with a white puzzle piece icon on the left and the text "Blocks" in white on the right.

and make sure everything looks right.

Goal: To have the w:b sending code with your 2 neopixel codes.

If your program includes the following:

- Hourly readings for all data
- A program for lights using Pin 14 (RXI)
- A program for lights using Pin 15 (TXO)

Then:

- Download your code
- Flash your micro:bit with the new code
- Share this code in the recorder's document - be sure to include names of the rest of the team.

# Try some new programs with your neopixels

<http://www.playfulcomputation.group/luminous-science-programs-for-lanterns.html>

- 1) Each group choose a different example on this site.
- 2) Download the program, flash the m:b (in your ecosystem) and see what it does. Try to understand the code.
- 3) Edit the program. Change the program in two ways at least.
- 4) Redownload the program, flash the m:b, and see how the changes you made affected the lights.
- 5) Share your program on your google assignment/doc. In your document, describe what you changed in the program.

Show and tell?

# Program your lights using data from the ecosystem

You should already have a plan for how your lantern will react to different data. But you can still redesign the lights plan!

Keep in mind the following when thinking about how the lights will react to the data:

- What is important for your ecosystem?
- What are you curious about when you aren't around for your ecosystem?
- How often you want to check the data?
- How often do you want to update the lights?
- Do you want to show only live data or past data as well? (see this example for past data)
- Could you use the lantern to predict when your ecosystem might start to be unhealthy?

Day 4

# Keep programming your lights to respond to data in the ecosystem

Keep in mind the following:

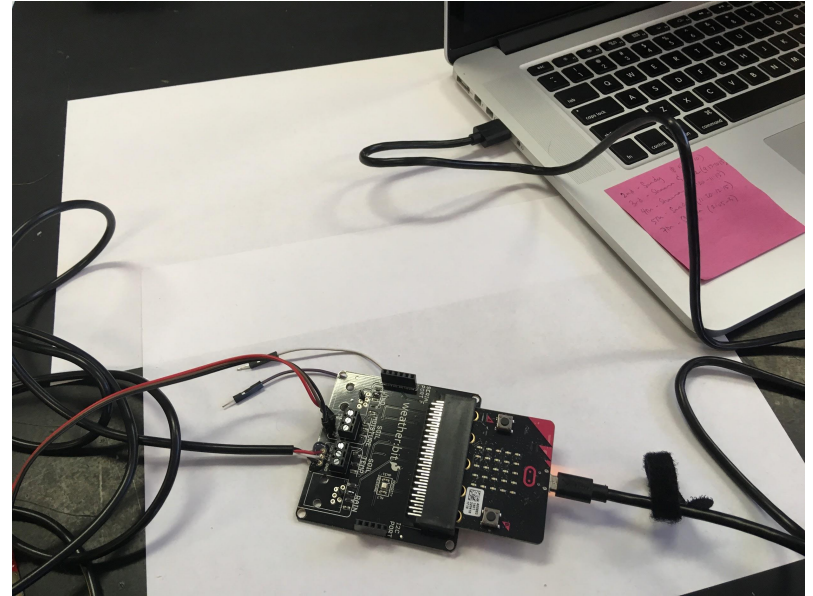
- What is important for your ecosystem?
- What are you curious about when you aren't around for your ecosystem?
- How often you want to check the data?
- How often do you want to update the lights?
- Do you want to show only live data or past data as well? (see this example for past data)
- Could you use the lantern to predict when your ecosystem might start to be unhealthy?



# How to setup weather:bit and micro:bit with ecosystem

- 1) Place the sensors into non-invasive areas of your ecosystem.\*\*
- 2) Tape your weather:bit to the wall of the ecosystem.
- 3) Make sure you can access the usb cable outside of the ecosystem and it can reach power.
- 4) Add jumper cable extensions so lights can be attached outside of the ecosystem.

\*\*Note: If you have a large animal (turtle) that might eat the wires protect them!



Be sure your micro:bit has power. Use a usb port



Even through your micro:bit is in the ecosystem you can still program it since you have access to the usb.

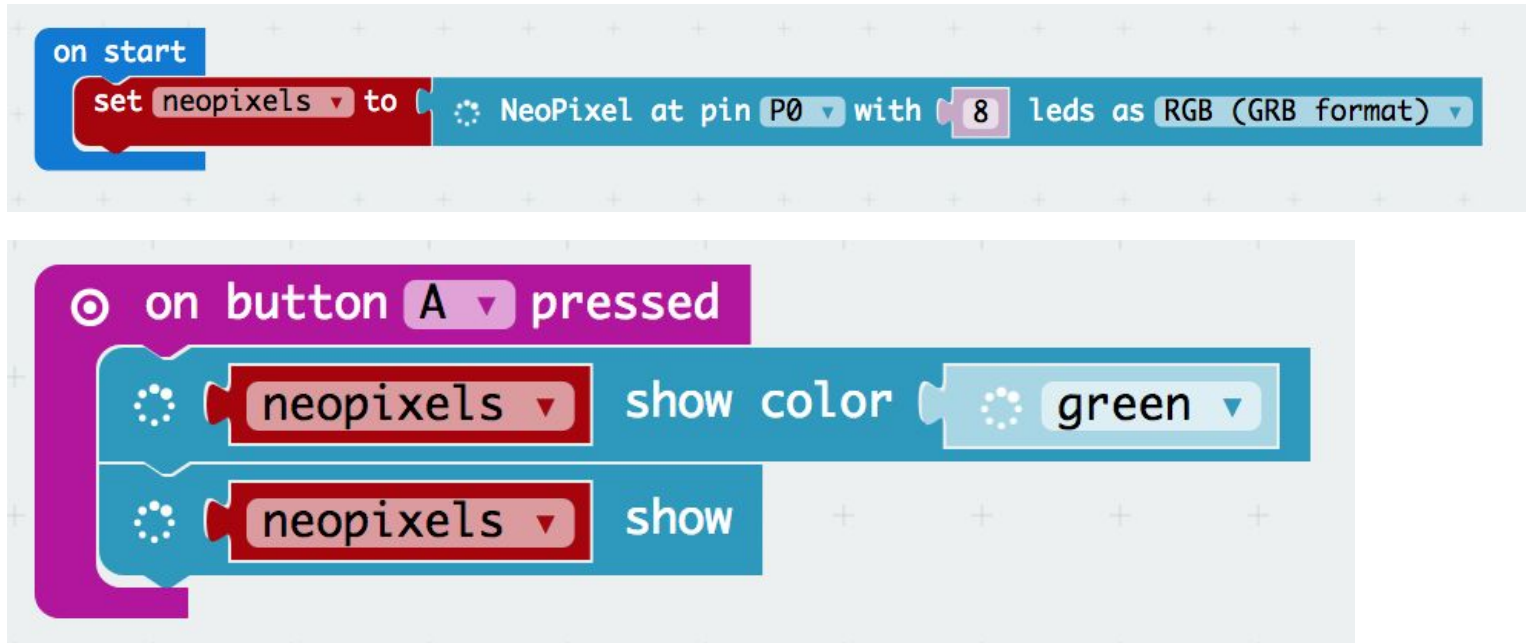
We will check the data on Monday!

# Try some new programs with your neopixels

- Program in your lantern pairs.
- If you are the recorder, share the w:b/m:b setup that is in your ecosystem with the other lantern pair in your group.
  
- You can program at your desk. Only plug in the micro:bit when you are ready to try out a program and download (flash) it to the micro:bit.
- At the end you can combine your two codes to make them both work on a single micro:bit.

Extras

# Make a program where the neopixels respond to a button press



The image shows two sections of Scratch code. The first section is an 'on start' block containing a 'set neopixels to' block. The 'neopixels' dropdown is set to 'NeoPixel at pin P0', the 'with' field is '8', and the 'leds as' dropdown is 'RGB (GRB format)'. The second section is an 'on button A pressed' block containing two 'neopixels' blocks: 'show color' set to 'green' and 'show'.

```
on start
  set neopixels to NeoPixel at pin P0 with 8 leds as RGB (GRB format)

on button A pressed
  neopixels show color green
  neopixels show
```

Do something more exciting than this example!!

```
on start
  set LEDs to NeoPixel at pin P0 with 8 leds as RGB (GRB format)

on button A pressed
  call function blinky

function blinky
  repeat 4 times
    do
      LEDs show rainbow from 1 to 360
      LEDs show
      pause (ms) 100
      LEDs clear
      LEDs show
      pause (ms) 100
```

What is the function doing for this code?

Modify your neopixel code to use a function!

Re-SHARE the link in your google assignment/doc

# Answer the following questions

```
on start
  set count to 0
  show icon [LEDs]

on button B pressed
  set count to 0
  show icon [LEDs]

on button A pressed
  change count by 1
  if (count > 2)
  then
    show string "medium"
  else if (count > 4)
  then
    show string "big"
  else
    show string "?"
```

The image shows a Scratch script for a microbit. It starts with an 'on start' block containing 'set count to 0' and 'show icon [LEDs]'. There are two 'on button B pressed' blocks, each containing 'set count to 0' and 'show icon [LEDs]'. The main logic is in an 'on button A pressed' block: it increments 'count' by 1, then uses an 'if' statement to check 'count > 2' (showing 'medium'), 'count > 4' (showing 'big'), or otherwise showing '?'. The background is a light blue grid.

When will the microbit say medium?

When will the microbit say big?

When will the microbit say “?”



# Make this code & download it to a micro:bit

```
on start
  set count to 0
  show icon

on button B pressed
  set count to 0
  show icon

on button A pressed
  change count by 1
  if (count > 2)
  then
    show string "medium"
  else if (count > 4)
  then
    show string "big"
  else
    show string "?"
```

Does it behave like you expected? Why or why not?

The image shows a Scratch script for an 'on button A pressed' event. The script starts with an 'on start' block containing a 'set count to 0' block and a 'show icon' block. The main event block contains a 'change count by 1' block, followed by an 'if' block. The 'if' block has three conditions: 'count > 2', 'count > 4', and a default 'else' block. The 'if' block contains three 'show string' blocks: 'medium' for the first condition, 'big' for the second, and '?' for the default. There is also a separate 'on button B pressed' block containing 'set count to 0' and 'show icon'.

```
on start
  set count to 0
  show icon

on button A pressed
  change count by 1
  if (count > 2)
  then
    show string "medium"
  else if (count > 4)
  then
    show string "big"
  else
    show string "?"

on button B pressed
  set count to 0
  show icon
```

DEBUG: This code will never say “big”. Fix the code so that it says big when count is more than 4.

# Vernier sensor and light pins

Pins for vernier sensors:

- P1 - wind
- P2 - rain
- P3 - back wire on end of the row
- P4 - back wire on bottom row

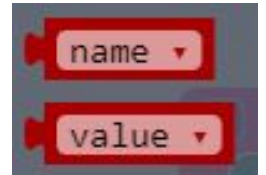
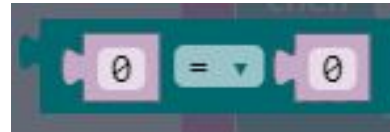
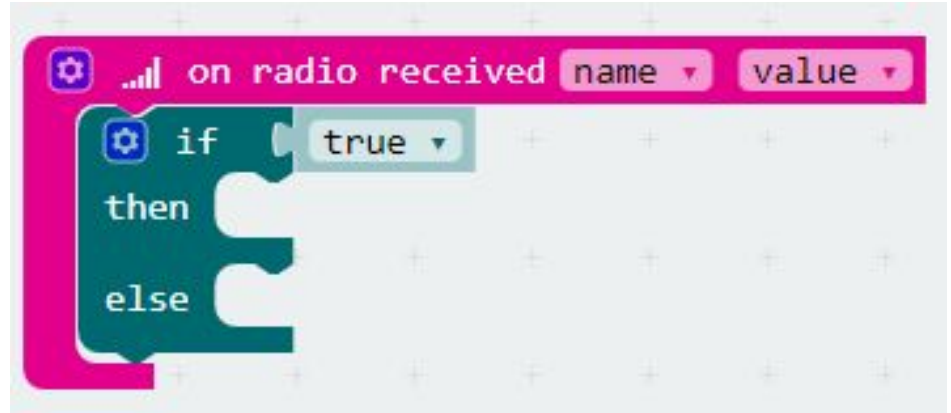
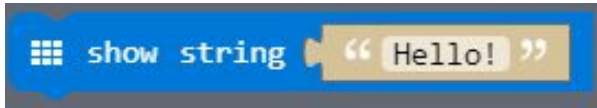
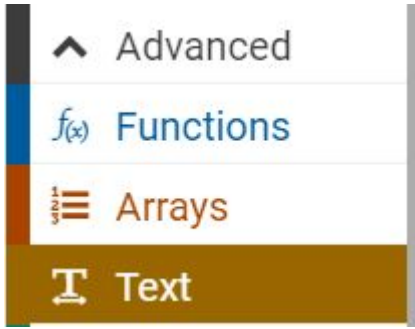
Pins for lights:

- P14 - RXI (lights)
- P15 - TXO (lights)

# Random Other Activities

# Next level - Hints

Blocks you will need:



# Partner Challenge - Complete and share out.

Set up a radio group with a partner.


When you send a message to your partner, their micro:bit should show the message.

When your partner presses A on their micro:bit, your microbit should make a smiley face.

Get creative and take the challenge to the next level. Also trade off whose micro:bit is sending and whose is receiving.

# One step further

- Set your microbit to radio channel 1
- Check for the message that is named “light”
- Have your microbit show

LIGHT = \_\_\_\_\_  


Here is the number of the light level you receive.

# Get creative with your neopixels (in pairs)

## Option A:

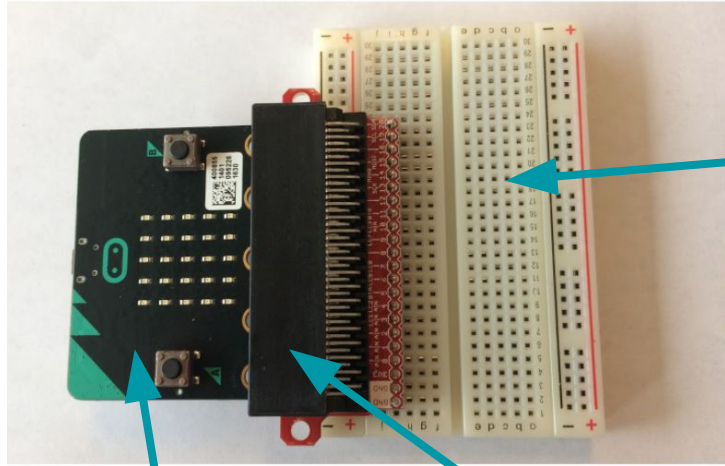
- 1) Set up a radio group with a partner (2 micro:bits).
- 2) When you send different messages to your partner, their neopixels show something different.

## Option B:

- 1) Learn to change the brightness of the neopixels and/or customize your own colors.
- 2) Make some pattern of lights/brightness change, repeat, or rotate based on some kind of input (e.g. light level, button presses).



# Attaching breakout board with neopixel LEDs with bread board

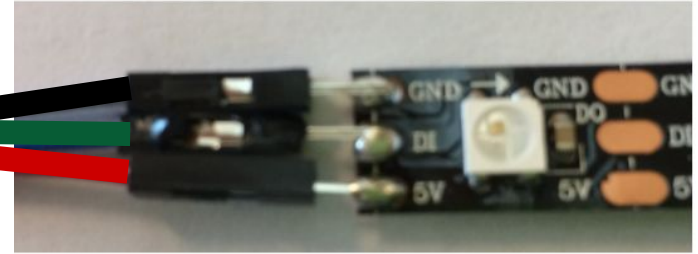
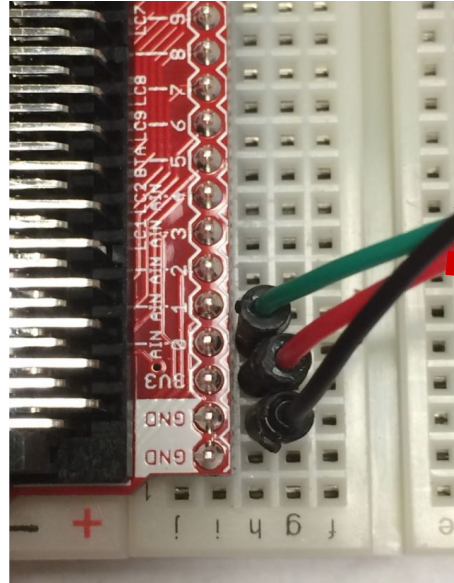
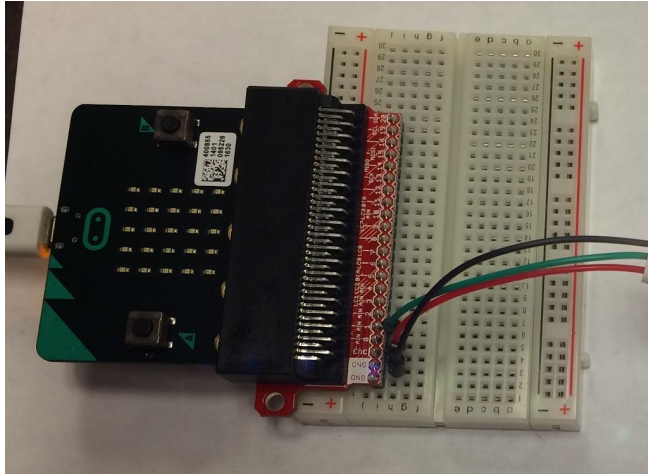


Micro:bit

Breakout board

Bread board

# Wire your LEDs



PIN X (m:b) → DI (LEDs),  
GND (m:b) → GND (LEDs)  
3V3 (m:b) → 5V (LEDs)

RX1 (w:b) → signal  
GND (w:b) → ground  
VCC (w:b) → power

