

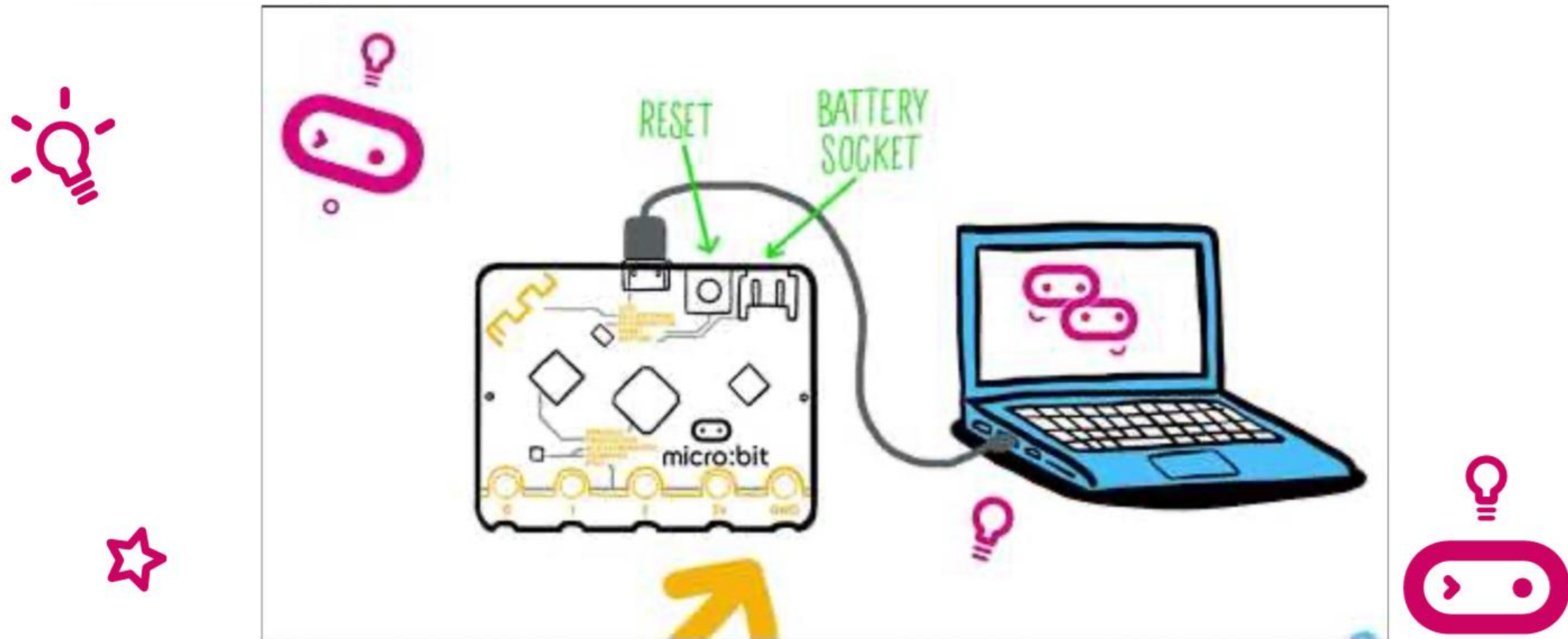


 micro:bit

Kitronik Stop Bit Lessons

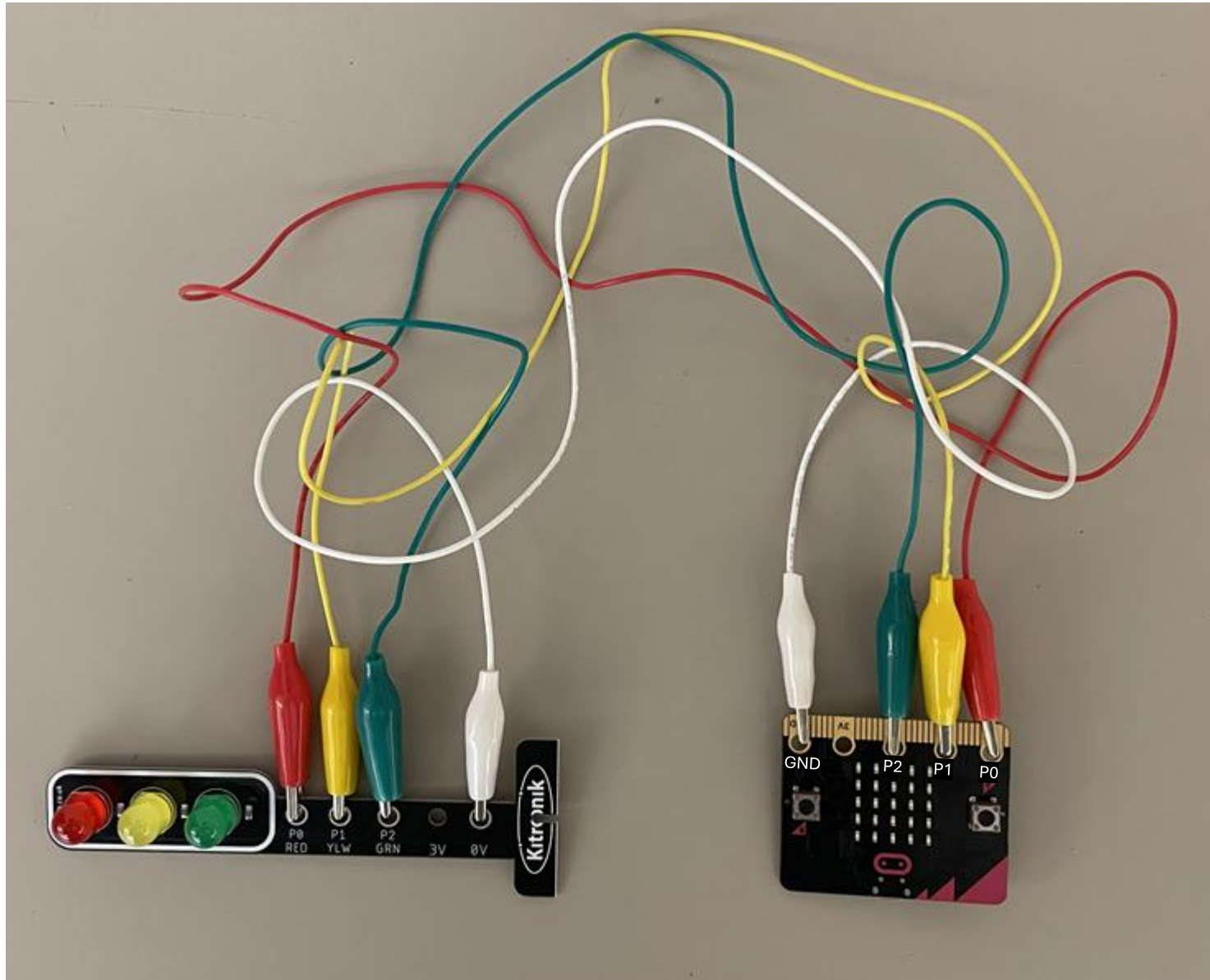


Introducing The Micro:bit



Watch this video. <https://www.youtube.com/watch?v=FfVinxVyzYo> (1:29)

Ask Mr. Desmond for the micro:bit device, a set of wires with alligator clips, and the Kitronik Stop Bit.



Connect the ports on the micro:bit to the ports on the Stop Bit using the provided wires with alligator clips.

Connect: P0 to P0, P1 to P1, P2 to P2, and 0V to GND.

Check in with Mr. Desmond.

Ask Mr. Desmond for the micro:bit USB cable and the lamp post support base.

Setup the traffic light on the support base.



Plug the USB cable into the micro:bit and your Chromebook.

Wait a moment for the Chromebook to recognize and connect to the micro:bit.

Check to make sure that the micro:bit is accessible when viewing your Google Drive or your Files.

Check in with Mr. Desmond.

Ready to code? Go to the next page to begin.

Open a new MakeCode project.

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

Microsoft | micro:bit

Introduction to the BBC micro:bit

Show Instructions

My Projects [View All](#) [Import](#)

New Project

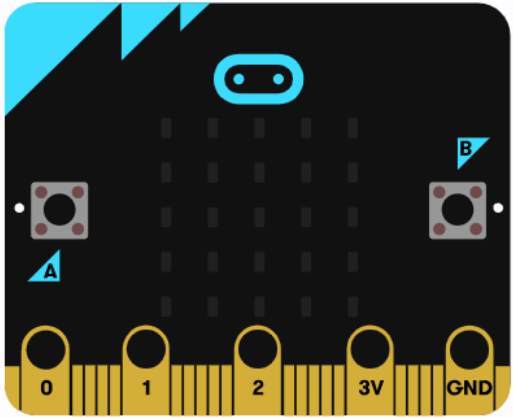
← Click here.
Name your new project Traffic Light.

Tutorials

- New? Start Here
Flashing Heart
- Name Tag
- Smiley Buttons
- Dice
- Love Meter
- Micro Chat

Follow all the steps of this lesson. Do not skip any pages.

The screenshot displays the Microsoft MakeCode IDE for the micro:bit. The top navigation bar includes the Microsoft logo, the text "micro:bit", and tabs for "Blocks" and "JavaScript". On the right side of the top bar are icons for home, share, help, settings, and a "Sign In" button. The main workspace is divided into three sections: a micro:bit board on the left, a block palette in the center, and a workspace on the right. The block palette lists categories: Basic, Input, Music, Led, Radio, Loops, Logic, Variables, Math, Extensions, and Advanced. The workspace contains two blue blocks: "on start" and "forever". The bottom bar features a "Download" button, a project name "Traffic Light", and several utility icons.



Search...

- Basic
- Input
- Music
- Led
- Radio
- Loops
- Logic
- Variables
- Math
- Extensions
- Advanced



Click on Extensions.

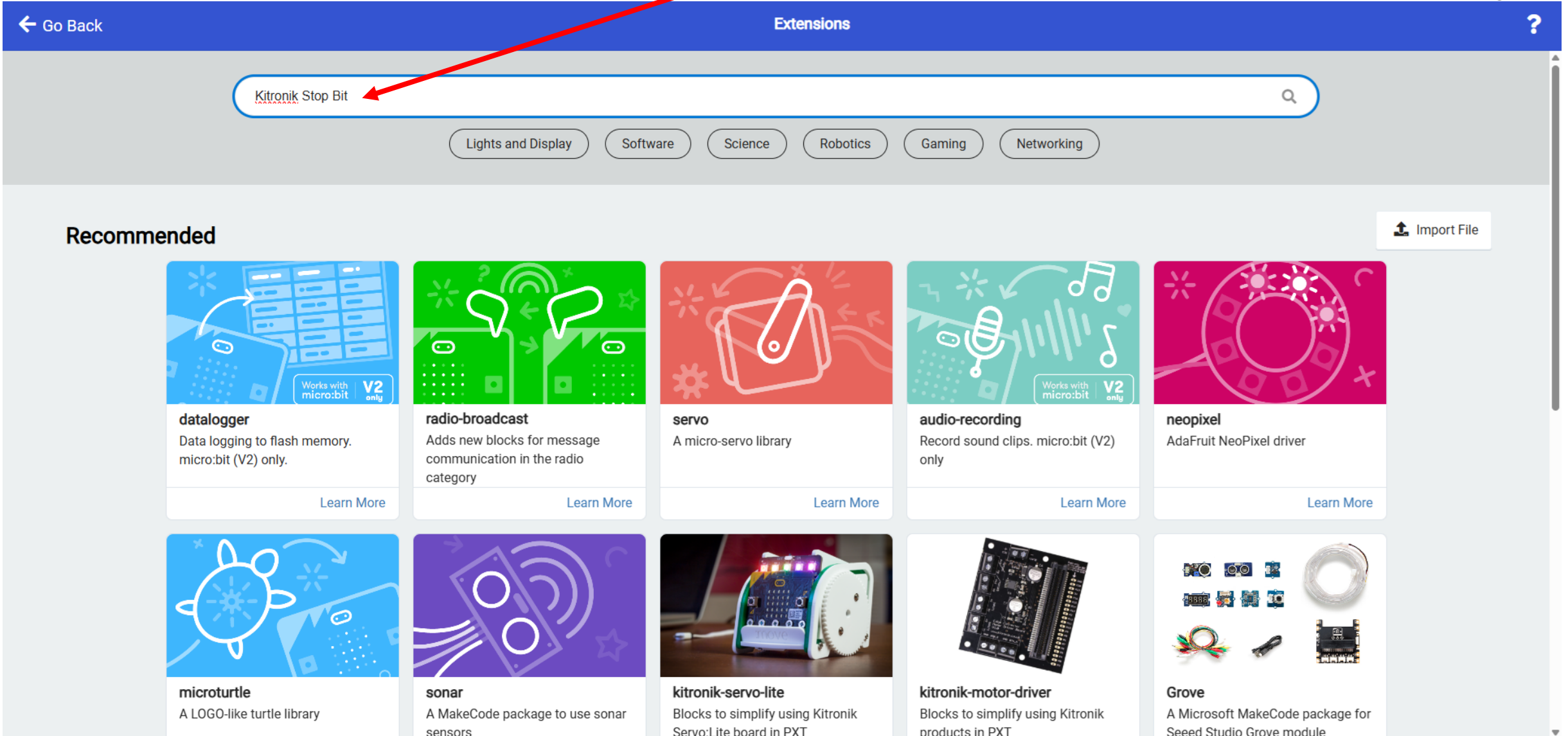


Download

Traffic Light



In the search box type Kitronik Stop Bit and then click the .



← Go Back Extensions ?

Kitronik Stop Bit

Lights and Display Software Science Robotics Gaming Networking

Recommended [Import File](#)

- datalogger**
Data logging to flash memory. micro:bit (V2) only.
[Learn More](#)
- radio-broadcast**
Adds new blocks for message communication in the radio category.
[Learn More](#)
- servo**
A micro-servo library.
[Learn More](#)
- audio-recording**
Record sound clips. micro:bit (V2) only.
[Learn More](#)
- neopixel**
AdaFruit NeoPixel driver.
[Learn More](#)
- microturtle**
A LOGO-like turtle library.
- sonar**
A MakeCode package to use sonar sensors.
- kitronik-servo-lite**
Blocks to simplify using Kitronik Servo:Lite board in PXT.
- kitronik-motor-driver**
Blocks to simplify using Kitronik products in PXT.
- Grove**
A Microsoft MakeCode package for Seeed Studio Grove module.


Click on the kitronik-stopbit box.

← Go Back Extensions ?

Kitronik Stop Bit


Lights and Display Software Science Robotics Gaming Networking

Home Import File



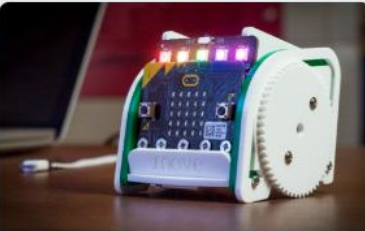
kitronik-stopbit
Custom blocks for STOP:bit for BBC micro:bit

[Learn More](#)




microbit-robot
Universal Robot blocks for MakeCode micro:bit - beta

[Learn More](#)




kitronik-servo-lite
Blocks for driving Kitronik micro:bit boards

[Learn More](#)



kitronik-motor-driver
Blocks for driving the Kitronik micro:bit motor driver board

[Learn More](#)



kitronik-move-motor
Custom blocks for the :MOVE Motor

[Learn More](#)

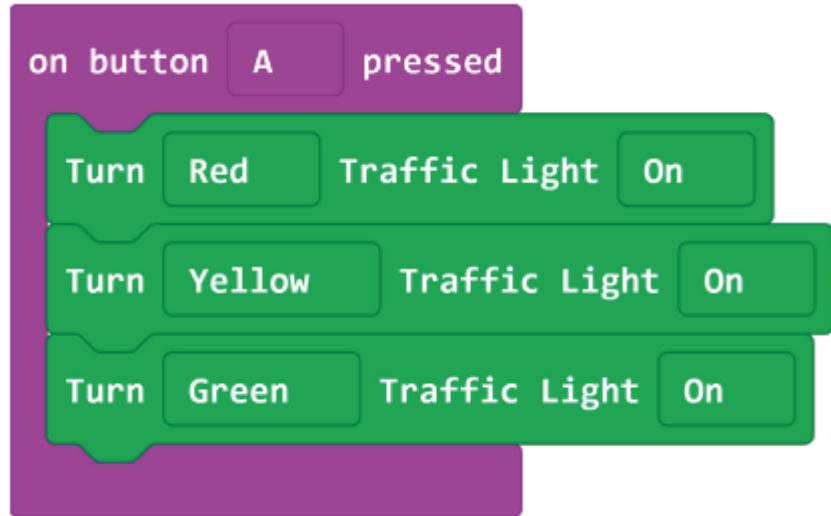
Click here.

Notice the STOP:bit code blocks have been added.


The screenshot displays the Microsoft MakeCode editor interface for a micro:bit project. The top navigation bar includes the Microsoft logo, the micro:bit logo, and tabs for 'Blocks' and 'JavaScript'. On the right side of the navigation bar are icons for home, share, help, settings, and a 'Sign In' button. The left sidebar features a virtual micro:bit board and a category menu with options: Basic, Input, Music, Led, Kitronik STOP:bit (highlighted in green), Radio, Loops, Logic, Variables, Math, Extensions, and Advanced. The central workspace shows a script with two green blocks from the Kitronik STOP:bit extension: 'Make Traffic Light state to Stop' and 'Turn Red Traffic Light On'. A red arrow points to the 'Kitronik STOP:bit' category in the sidebar. The right workspace contains two blue blocks: 'on start' and 'forever'. The bottom toolbar includes a 'Download' button, a search bar containing 'Traffic Light', and various editing tools like undo, redo, and zoom.

Traffic Lights - Intro

Create the code sequences below.



```
on button A pressed
  Turn Red Traffic Light On
  Turn Yellow Traffic Light On
  Turn Green Traffic Light On
```



```
on button B pressed
  Turn Red Traffic Light Off
  Turn Yellow Traffic Light Off
  Turn Green Traffic Light Off
```

Download your code.

Carefully follow the instructions on the next page.

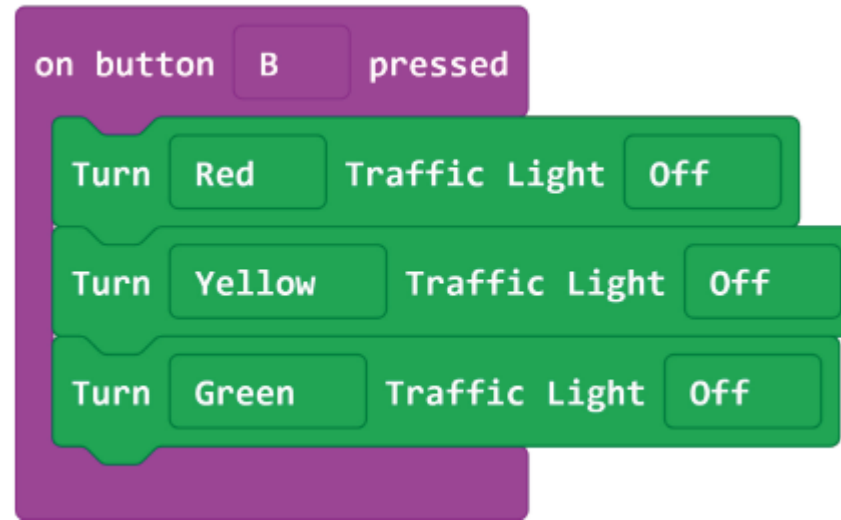
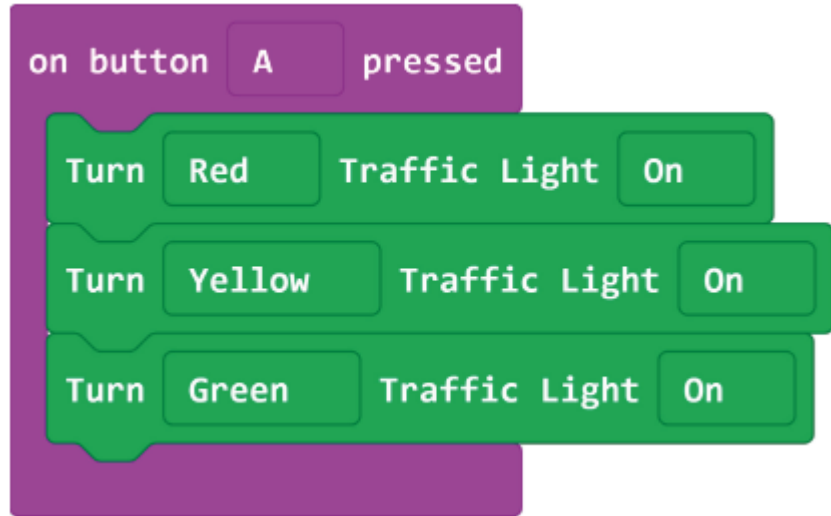
Step-by-step Guide To Download Your Code To Micro:Bit

- 1) Connect. Plug your micro:bit device into the Chromebook with a USB cable; it will appear as a new drive named MICROBIT in your Files app.
- 2) Code & Download. In MakeCode, build your program and click the purple Download button. Do not PAIR. Click the blue Download as File button. This saves a “_____.hex” file (e.g., microbit-program.hex) to your Chromebook’s Google Drive or Google Files.
- 3) Locate File. Open the Files app (the blue folder icon) and find your downloaded “_____.hex” with the files in your Google Drive or with the files in your Google Files.
- 4) Drag & Drop. Click and drag the “_____.hex” file directly onto the MICROBIT drive.
- 5) Transfer. The MICROBIT drive will momentarily disappear as the code is transferred to the micro:bit device. Look for the blinking lights on the micro:bit and an on-screen message showing you the status of your download.
- 6) Run. Your program will start automatically on the micro:bit once the transfer is complete.

Need help? Watch a video to help you with these steps to download your code to the micro:bit from a Chromebook.

<https://www.youtube.com/watch?v=9V2oAsB3F8s>

Traffic Lights - Intro



Test your code.

Press Button A. What happens?

Press Button B. What happens?

Show Mr. Desmond your working traffic lights.




Traffic Lights – Light Sequencing

Create the code sequences below.



```
on button A pressed
  Turn Green Traffic Light Off
  Turn Yellow Traffic Light Off
  Turn Red Traffic Light On
  pause (ms) 1000
  Turn Red Traffic Light Off
  Turn Yellow Traffic Light On
  pause (ms) 1000
  Turn Yellow Traffic Light Off
  Turn Green Traffic Light On
```



```
on button B pressed
  Turn Green Traffic Light On
  Turn Yellow Traffic Light Off
  Turn Red Traffic Light Off
  pause (ms) 1000
  Turn Green Traffic Light Off
  Turn Yellow Traffic Light On
  pause (ms) 1000
  Turn Yellow Traffic Light Off
  Turn Red Traffic Light On
```

Download your code.

Test your code.

Press Button A.
What happens?

Press Button B.
What happens?

Show Mr. Desmond your working traffic lights.

Traffic Lights – Light Sequencing

Write your answers to the following questions on a piece of paper.

- 1) What is the need for traffic lights?
- 2) What is the sequence of traffic lights?
- 3) Imagine you are at a real traffic light.
What is the usual timing for each of the lights?
(How long for green, how long for yellow, how long for red?)

Be prepared to discuss your answers with Mr. Desmond.

Simulating A Real Traffic Light System

Create a program for a traffic light system to control all three lamps in such a manner that the sequence of lamp colour changes simulates the action of a real traffic light.

Demonstrate your program and your working traffic light system to Mr. Desmond.

Save your code, you will be using it again in the future.



Traffic Light System Including Pedestrian Crossing Signals

1) Open your existing or create a new program for a traffic light system to control all three lamps in such a manner that the sequence of lamp colour changes simulates the action of a real traffic light.

2) In a new Makecode app window complete micro:bit Lesson 2 – Heart.

3) Return to your traffic light system program. Use the show LED block to create two distinct pedestrian crossing signals.

One for when to walk / cross.

One for when to not walk / not cross.



Traffic Light System Including Pedestrian Crossing Signals

4) Create the necessary code sequences for the micro:bit to display pedestrian crossing signals that are in sync with the colour light changes of the traffic light system.

The pedestrian crossing signals must be carefully timed to allow for pedestrians to cross safely, this includes using the signals to show when it is safe to cross and when it is not safe to cross or begin an attempt to cross.

5) Demonstrate your program and your working traffic light system with pedestrian crossing signals to Mr. Desmond.