

KidzCanCode

IGNITE, INSPIRE, INNOVATE

www.kidzcancode.com

Build a Traffic Light

KIDZCANCODE

TREVOR WARREN

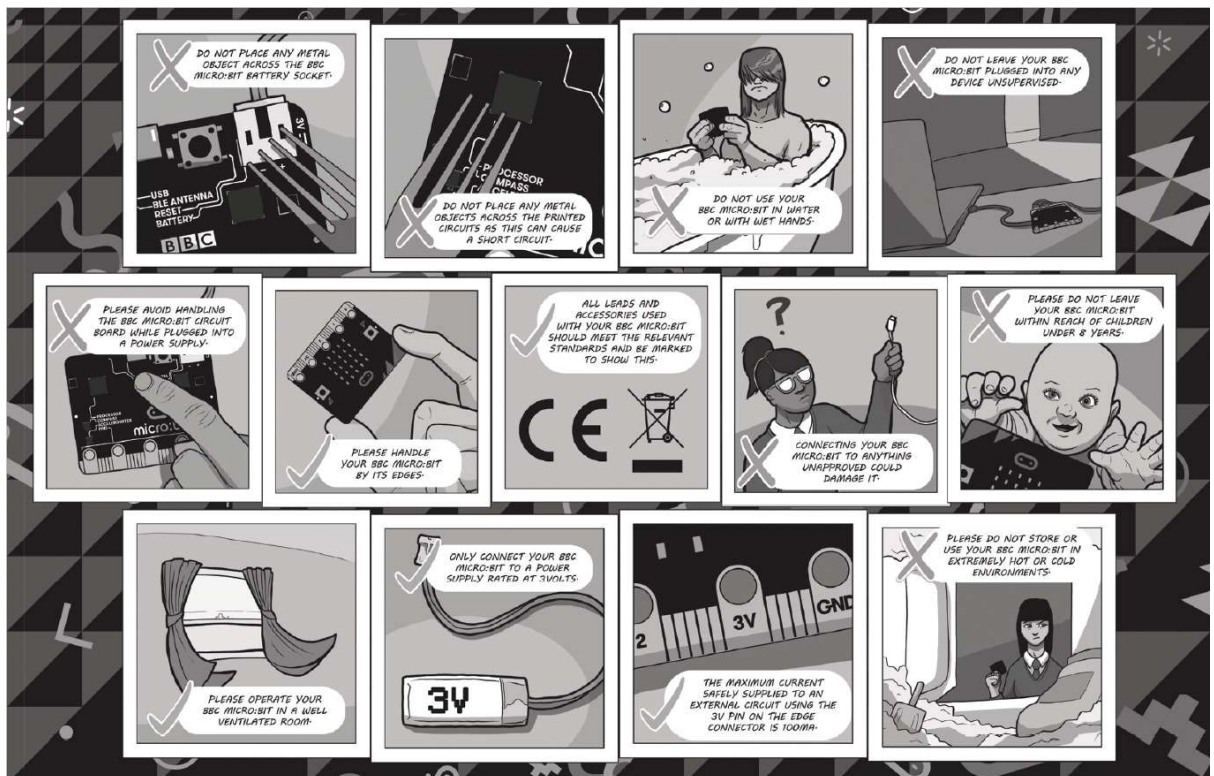
Tutorial 2.1k – Build A Traffic Light

1. Safety Warnings

THE MICRO:BIT IS AN EXPOSED BOARD, TO BE USED WITH CARE PLEASE RETAIN THIS INFORMATION FOR FUTURE REFERENCE. You can read the detailed document at - <http://microbit.org/guide/safety-advice/>

a. General Safety Warnings

Using the BBC micro:bit is easy to use but is designed to have all the electrical parts on display. This does mean there's a small risk that the parts can be damaged and even overheat with a risk of injury but a little bit of care and caution will ensure you and your micro:bit will stay fit and healthy.

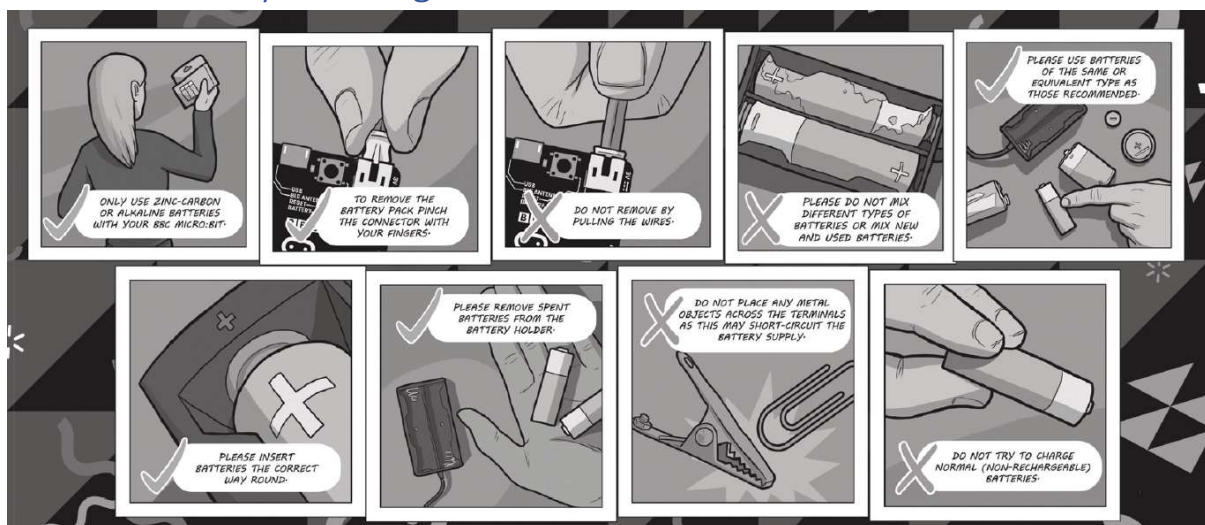


1. Always keep your BBC micro:bit in the anti-static bag when not in use. It's good practice for students to earth themselves before handling it.
2. Please handle your BBC micro:bit by its edges. This minimises the risk of damage through an electrostatic discharge.
3. Please use the battery pack and the USB lead provided to power your micro:bit. Do not use portable battery chargers or USB charging ports (often marked with a lightning bolt or 'SS'), to power your micro:bit. Using these may damage your micro:bit and stop it working properly.
4. Please avoid handling the BBC micro:bit circuit board while plugged into a power supply.
5. All peripherals (for example: USB cable, battery holder, sensors) used with your BBC micro:bit should comply with the relevant standards and should be marked accordingly.
6. Connecting your BBC micro:bit to any unapproved peripherals could damage your BBC micro:bit
7. Please do not attempt to keep using faulty micro:bits. If a school-issued micro:bit develops a fault, contact the vendor immediately.

Tutorial 2.1k – Build A Traffic Light

8. The maximum current safely supplied to an external circuit using the 3V pin on the edge connector is 100mA. Please make sure this limit is not exceeded.
9. Please do not store or use your BBC micro:bit in extremely hot or cold environments.
10. Do not place any metal objects across the printed circuits on the board as this can cause a short circuit damaging your BBC micro:bit. This can cause risk of burn or fire.
11. Do not use your BBC micro:bit in water or with wet hands.
12. Do not leave your BBC micro:bit plugged into a computer or any other device unsupervised.
13. Please do not leave your BBC micro:bit within reach of children under 8 years of age.
14. Please operate your BBC micro:bit in a well ventilated room To remove the battery pack, pinch the connector with your fingers. Do not remove by pulling the wires.

b. Battery Warnings



1. Do not try to charge normal (non-rechargeable) batteries
2. Please do not mix different types of batteries or mix new and used batteries.
3. Please use batteries of the same or equivalent type as those recommended.
4. Please insert batteries the correct way round (with the correct polarity).
5. Please remove spent batteries from the battery holder.
6. Do not short-circuit the battery supply terminals, for example by placing a metal object across the terminals.
7. Only use Zinc or Alkaline batteries with your BBC micro:bit.
8. Please do not use rechargeable batteries

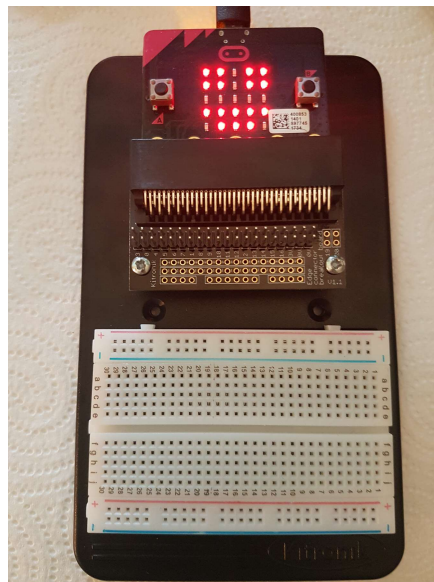
Tutorial 2.1k – Build A Traffic Light

2. Pre-requisites

If you have questions with the assembly of the micro:bit, edge connector breakout board, mounting board and the breadboard please drop us a note at help@kidzcancode.com. The edge connector board, mounting board and the breadboard are part of the Kitronix Inventors kit which needs to be purchased separately.

To be able to perform this tutorial you will need the following components –

1. Parts required –
 - a. 1 x BBC Micro:bit
 - b. 1 x Mounting Plate
 - c. 1 x Edge connector breakout board
 - d. 1 x Bread board
2. Assembly required –
 - a. Bread board mounted on top of the mounting plate
 - b. BBC Micro:bit inserted into the Edge Connector breakout board



Before proceeding please check your setup and confirm that all the required parts are configured as demonstrated in the above picture.

Tutorial 2.1k – Build A Traffic Light

3. Learning Objectives

The objectives of this tutorial are to introduce the student to the following concepts –

- Coding for use of the buttons on the micro:bit, perform a given action when a button is pressed.
- Understand how to write a digital on or off signal to a particular pin on the micro:bit to turn devices on/off
- Learn the use of loops to run code repeatedly

The BBC micro:bit is a powerful little computer. Through programming these games kids explore more advanced computer science concepts. Along the way kids are encouraged to share, create and extend the games using their own imagination and creativity.

This tutorial builds upon concepts introduced in previous tutorials so please make sure you have covered the previous tutorials before you dive into this one. So overall this tutorial intends to build upon concepts learnt in previous tutorials while exploring new concepts.

In future tutorials we will continue to build upon the concepts learned here and will build more complex interactive games using the functionality provided by the micro:bit.



Tutorial 2.1k – Build A Traffic Light

4. Activity

a. Activity

This activity involves designing a traffic light circuit using a few easy to use electronic components and combining them with the micro:bit. In addition to wiring up the electronic components on a circuit you will also need to write code using the block editor which will tell the micro:bit to activate and de-activate the different electronic components making our traffic light work.

The traffic light project is at the intersection of electronics and programming and is based around the micro:bit. This project also makes use of components available within the Kitronix Inventors kit.



The activity is designed to have additional challenges which allow the developer to keep pushing the boundaries. The challenges in this activity include –

- As part of the initial activity we will wire up all the electronic components and integrate them with the micro:bit to function as a simple traffic light
- The next phase will involve a bit more work requiring you to insert timers into the code and extend the project to make it more efficient and realistic in the way it works

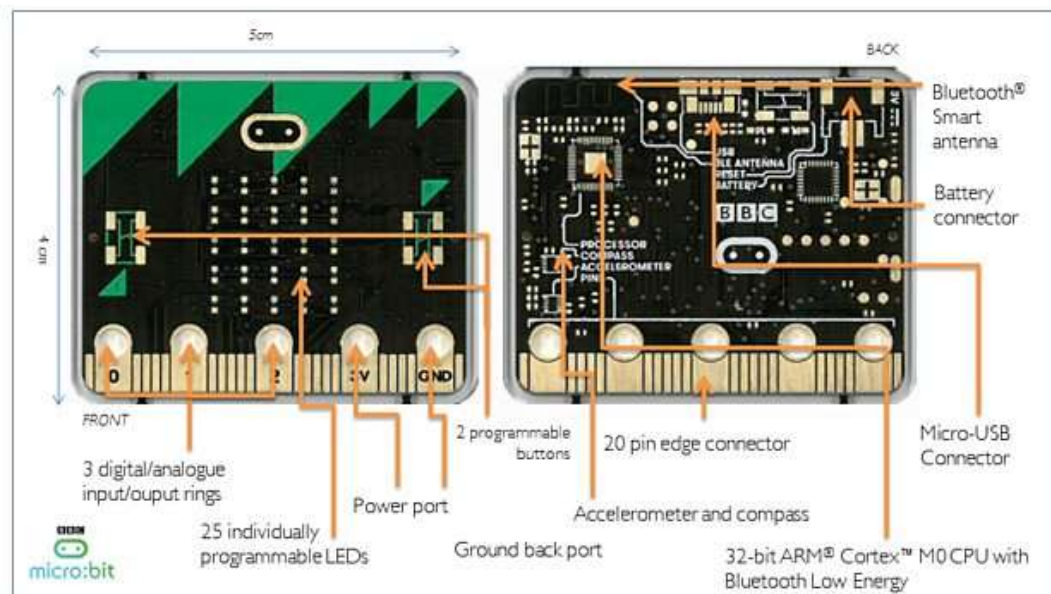
Tutorial 2.1k – Build A Traffic Light

b. How Does It Work

This section talks about the BBC micro:bit and what it's made up of. If you have already read through this section then feel free to skip directly to the next section. The BBC micro:bit is a powerful little board and has various types of sensors on board. Here's what makes up the BBC micro:bit.

1. Size: approx. 5cm x 4cm.
2. Weight: 8g.
3. Processor: 32-bit ARM Cortex M0 CPU.
4. Bluetooth Low Energy.
5. Digital Compass.
6. Accelerometer.
7. Micro-USB controller.
8. 5x5 LED matrix with 25 red LEDs.
9. Pins for connecting external sensors, LED's, etc.

Here's what the micro:bit looks like –



Front of the board (left hand side)

1. Button A (left button with edge connector at the bottom) – labelled A on the board
2. Button B (right button with edge connector at the bottom) – labelled B on the board
3. P0 (left large pin (crocodile clip port) with edge connector at the bottom) - labelled 0 on the board
4. P1 (middle large pin (crocodile clip port) with edge connector at the bottom) - labelled 1 on the board
5. P2 (right large pin (crocodile clip port) with edge connector at the bottom) - labelled 2 on the board
6. +3V - labelled 3V on the board. This is 3V PWR OUT
7. GND
8. P3 – P22 pins from left to right with edge connector at the bottom. Referred to as Pins when referencing that part of the board. Text will talk about 'pins' when referring to

Tutorial 2.1k – Build A Traffic Light

individual connections or the general way of connecting to the board – not labelled on the front of the board

9. LED matrix referred as the 'screen' - not labelled on the board
10. LED coordinates starting at 0,0 top left corner and ending at 4,4 at the bottom corner - not labelled on the board

The order of the large pins as follows: P0 P1 P2 3V GND labelled 0, 1, 2, 3V GND on the board

Rear of the board (Right hand side)

1. 1. USB Plug (Micro-USB plug) – labelled USB on the board
2. Button R (reset button) – labelled Reset on the board
3. Status LED – not labelled on the board
4. Battery socket – labelled Battery on the board

Other components on the board include

1. Accelerometer
2. Compass
3. Bluetooth Smart Technology Antenna
4. AAA Battery Holder - not labelled on the board
5. Processor (Cortex M0)

The BBC micro:bit is programmable in a few different languages. You can write code for the micro:bit using the Makecode block coding interface, Javascript, Python or even in C. Most of our tutorials will cover the use of the Makecode block coding interface built by Microsoft for the micro:bit.



Tutorial 2.1k – Build A Traffic Light

5. Let's Get Started

It's time to write some code and get going with coding our game. So let's head over to the micro:bit block code editor page (<https://makecode.microbit.org/>) and get coding!!!

a. How Does A Traffic Light Work

We encounter traffic lights almost every day of our lives but it's likely that we've never stopped to think how these interesting devices function, or how do multiple of these traffic lights remain co-ordinated across a large intersection. It probably takes a lot of planning, programming and testing to ensure that when one traffic light is signalling (green) for traffic on one side to turn, the traffic light on the other side signals traffic to stop going straight ensuing that there is safe movement of cars/pedestrians all around.

It seems all so mundane and common but the technology and work that goes into making it is quite interesting and has been developed, perfected over many decades. While we will not be really dealing with real world traffic systems, our tutorial is going to build a simple replica of a real world traffic system.



In a real world traffic system there are three known states i.e. stop (red), get read to stop or start (amber) and go (green). These are the three states that we need to be aware of and signal to those driving on the street. For the traffic lights to work and signal to drivers different actions, they have to be triggered off in sequence i.e. if you have shown traffic a green light, you maintain a timer (let it stay on for a fixed period of time), then trigger of the amber light (for a short period of time) finally switching to a red light. See the bullet points below which highlight the same point –

1. Stop to Start
 - a. Red light currently on
 - b. Amber goes on (on some traffic light systems) for a short time
 - c. Green light comes on signalling traffic to start moving
2. Come to a stop
 - a. Green light currently on
 - b. Amber light goes on for a short time
 - c. Red light comes on signalling traffic to halt

Tutorial 2.1k – Build A Traffic Light

The reason for the amber light is to give traffic driving along a heads up and let them know that very soon the lights are going to change to red requiring them to come to a complete stop. This avoids drivers having to slam on their brakes causing accidents.

In some cities the reverse is also true i.e. when transitioning from red to a green the lights switch to amber for a few seconds and then switch over to green. This is not something that we've seen happen everywhere but it's definitely also how some cities implement their traffic light systems. In our case we will be using three different colour LED's for our traffic light with colours green, red and yellow.

b. Components Required For the Circuit

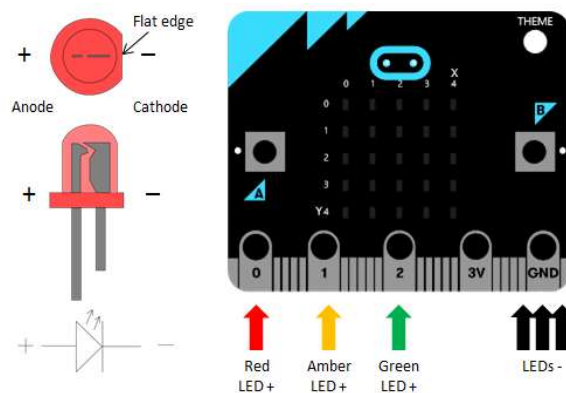
To build this tutorial you will need the following components –

1. 1 Red LED
2. 1 Orange LED
3. 1 Green LED
4. 1 Bread board
5. 6 x male to female DuPont wires

You should find all of the above components in the “Kitronik Inventors kit”. LED's are small active electronic devices which we will be using extensively in the micro:bit electronics track. LED's only work when energised in a particular way i.e. when their Anode (positive terminal) is connected to the positive terminal and Cathode (Negative terminal) of the LED is connected to the negative/ground terminal. So please use care and if you are unsure please speak with your parent or some else who understands how to wire electronic devices before you make any connections.

c. Wiring Up The Components

Before you begin please make sure you have disconnected the micro:bit from all power supplies i.e. from the USB that's plugged into your computer or the battery box if that's what you are using to drive your micro:bit. Keeping the micro:bit disconnected from the power supply ensures that you can build your circuit and validate it, fixing any issues before you power it on. This approach will save you a lot of time and possibly also some \$\$\$ in the longer run. So, let's get started with wiring up the components.



Setup the components as follows –

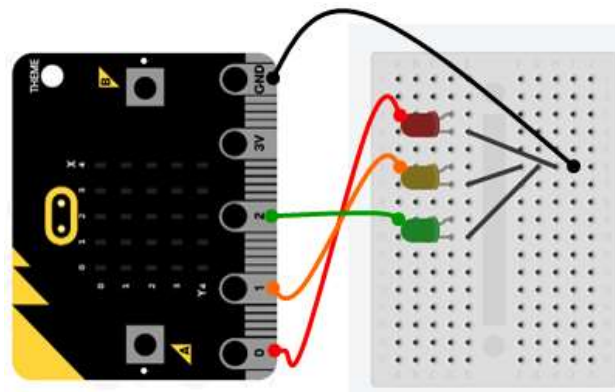
1. Push each of the 3 LED's into the bread board

Tutorial 2.1k – Build A Traffic Light

2. Red LED –
 - Connect up the Anode (longer leg) of the Red LED to Pin 0 of the micro:bit. Use a male-female DuPont wire and make the connection.
 - Connect up the Cathode (shorter leg) of the Red LED to the Ground pin on the micro:bit. Use a male-female DuPont wire and make the connection.
3. Orange LED –
 - Connect up the Anode (longer leg) of the Orange LED to Pin 1 of the micro:bit. Use a male-female DuPont wire and make the connection.
 - Connect up the Cathode (shorter leg) of the Orange LED to the Ground pin on the micro:bit. Use a male-female DuPont wire and make the connection.
4. Green LED –
 - Connect up the Anode (longer leg) of the Green LED to Pin 2 of the micro:bit. Use a male-female DuPont wire and make the connection.
 - Connect up the Cathode (shorter leg) of the Green LED to the Ground pin on the micro:bit. Use a male-female DuPont wire and make the connection.

Please do not power up the micro:bit until you've got the circuit reviewed by someone who understands electronics, is able to validate the circuit and confirm everything is as per the requirement. It's important that you get the connections validated before you power on your circuit.

See the image below to work out what your wired circuit might look like.

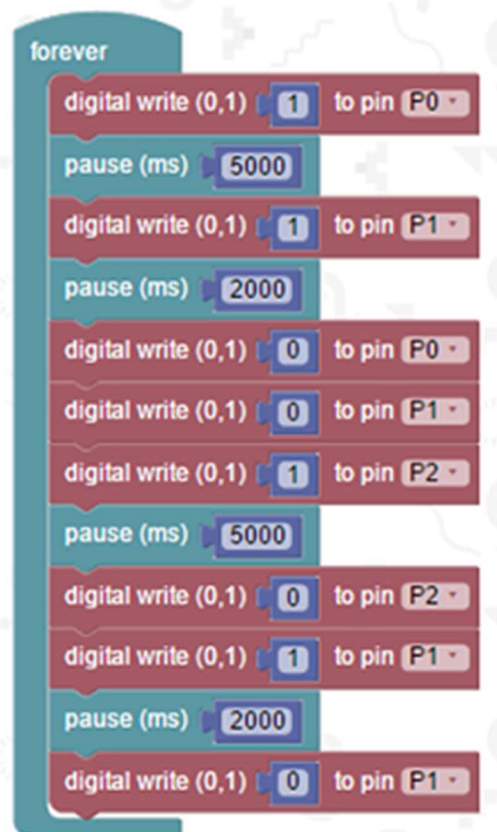
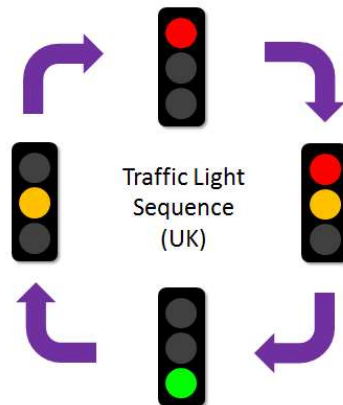


Tutorial 2.1k – Build A Traffic Light

d. Wiring Up the Components

Now that you have wired up your circuit let's open up the micro:bit block editor and write code so as to create the following sequence with the micro:bit.

1. Red lights up initially
2. Red transitions to Amber signalling green coming up soon
3. Green lights up
4. Green transitions to Amber signalling red coming up soon
5. Red lights up



Now power on the micro:bit and you should now have your LED's lighting up in the above sequence.

Tutorial 2.1k – Build A Traffic Light

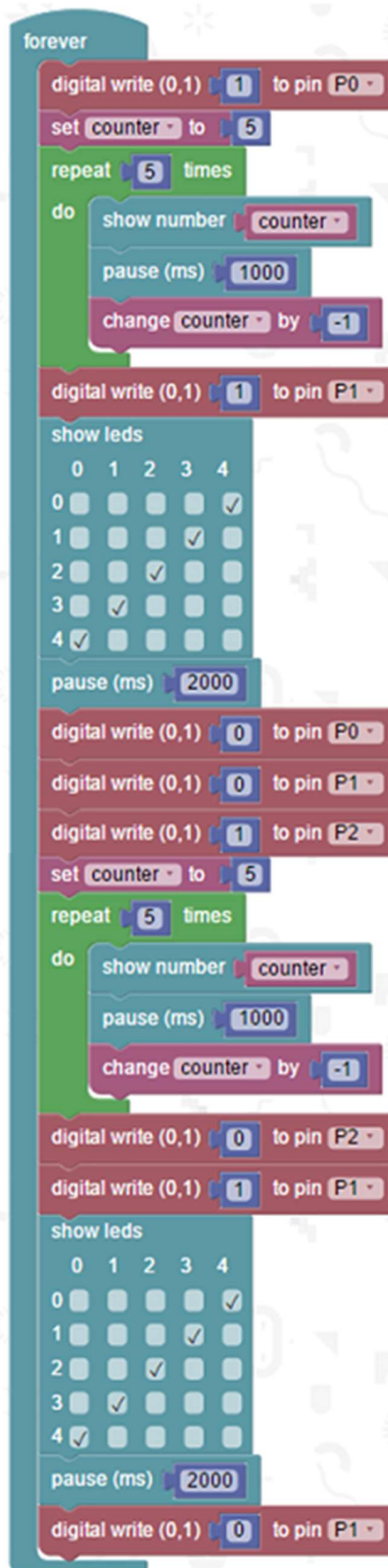
e. A More Advanced Solution

In the previous solution we put together we intentionally kept things simple and avoided using timers to display the amount of time remaining (between change of traffic lights) on the micro:bit. In our earlier section we spoke about timers and the need for timers to allow drivers to understand the change in state i.e. from green to amber and then red, this ensures that drivers have time to stop avoiding having to slam on their car brakes causing accidents. You might have seen traffic signals which display a countdown timer for pedestrians or even motorists counting down the transition (change) time between lights.

In this solution we are adding a countdown timer on the micro:bit when the green or the red light is on to inform the drivers who are waiting for the number of seconds remaining before the traffic light changes. The countdown timer will provide drivers with a visual message on the micro:bit reading out the amount of time remaining between transition of the lights.



Tutorial 2.1k – Build A Traffic Light

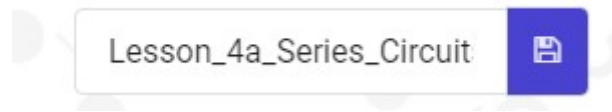


You are encouraged to make changes, improvise and customize the game using your own ideas.

Tutorial 2.1k – Build A Traffic Light

6. Downloading Your Code To The micro:bit

Once you have completed the program, enter a name for your program using the option provided below i.e. in the text box adjacent to the small save button.



Now that you have given your program a name and saved it you can download it your micro:bit. But before we do that let's confirm what drive your micro:bit shows up as. On most machines the micro:bit will show up as an additional USB drive. So head over into windows explorer and confirm what drive name (D:, E:, F:, G:, etc.) the micro:bit shows up as. You need to absolutely be sure what drive the micro:bit shows up as. Once you've confirmed what drive the micro:bit shows up as on your machine you can select the right drive when downloading the code to the micro:bit. If in doubt please ask the volunteer/mentor/session facilitator helping out.



To download the newly written code to the micro:bit, hit the download button shown above. You should now see a dialog box open up and you will be asked to save the file somewhere on your machine. Please choose the drive your micro:bit shows up as i.e. D: or E: or F: or whatever it shows up as on your machine. A sign of success is when you see the lights on the rear (orange) lighting up in quick succession suggesting that the code is being written to the micro:bit. On completion the micro:bit reboots and you should now see the code in action on the micro:bit.

If hitting the download button shown above does not open up a dialog box asking you to save to the micro:bit please save the file (you will have a <filename.hex> file) to your desktop. Then open up windows explorer and drag that file onto the drive which is your micro:bit. A sign of success is when you see the lights on the rear (orange) lighting up in quick succession suggesting that the code is being written to the micro:bit. On completion the micro:bit reboots and you should now see the code in action on the micro:bit.

Please feel free to customize the code blocks, have a play. Add your own custom code and re-download the code to the micro:bit. Give yourself a tap on the back, you've just completed your first circuit!!!!

Tutorial 2.1k – Build A Traffic Light

7. Challenges

Well done for completing the tutorial. There's a lot of ground we have covered in this tutorial so please feel free to make notes, come back to the tutorial at some point down the line and ask your learning facilitator any questions or doubts you might have on the concepts covered this far.

Let us now stretch it a bit further -

1. Write additional code to detect that key A is pressed and transition the lights from green to amber and then red
2. Write additional code to detect key B is pressed and transition the lights from red to amber and then green
3. Write additional code to detect A+B is pressed and blink all red, green and amber simultaneously

