University of BRISTOL

DiGi MaKERS
Create the digital world

Merchant Venturers School of Engineering

Outreach Programme

Micro:bit Virtual Circuits

1 hour edition

Created by

Ed Nutting

Organised by

Caroline.Higgins@bristol.ac.uk

Published on June 2, 2017

Notes to Teachers & Helpers

- This workshop is intended to last 1 to $1\frac{1}{2}$ hours.

- This workshop is intended for ages 9^+ (years 5^+).

- The content is intended to be learnt through group work (groups of 3 to 6), using this worksheet as a guide.

- The learning platform is the Micro:bit. Each child should have 1 Micro:bit.

- There are a number of versions of the Micro:bit editor. We will be using the most up to date editor called the "Microsoft PXT Beta editor".

- This workshop teaches the following skills:

  Items marked with an asterisk are directly relatable to the National Curriculum.

  * Using the Micro:bit block editor
  - Creating images
  * Variables (boolean and numeric)
  * Comparison logic
  * Conditional (if/elseif/else) blocks
  - Setting LED brightness
  * Example of using radio (bluetooth) communication

1 Introduction

Hi! In this short workshop we're going to try to introduce some of the concepts that computer science and electronic engineers use every day to design everything from your mobile phones to your car entertainment systems.

Let's get started. Each section is made up of four parts:

Actions Stuff for you to do. They are highlighted in blue.

Notes Notes about important stuff you need to be aware of (and possibly remember!). They are highlighted in red.

Questions Questions you should try to answer. Sometimes you'll need to write things down; other times you'll need to build something in the game. They are highlighted in yellow.

Ask a helper or the teacher to check your answers.

Goals Stuff you should have completed at the end of each section. They are highlighted in green.

We'll also write some information between parts and include plenty of screenshots to help you out.

2 What will we be making?

This workshop is all about programming and circuits. Using our Micro:bits, in groups on each table, we will each be programming our Micro:bits to be a single component in a virtual circuit. Each group will then connect their Micro:bits together wirelessly to simulate a complete circuit!

Some of the code has been provided for you because it's a little tricky. The rest we will show you how to write yourself - you'll be using a wide range of the blocks available.

3 Download the template file

Firstly, we need to download the template file for the code that will help you get started.

Actions

1. Your teacher will show you the URL of the website to go to.
2. Download the file to your “My Desktop” or “My Documents” folder.
3. Done!

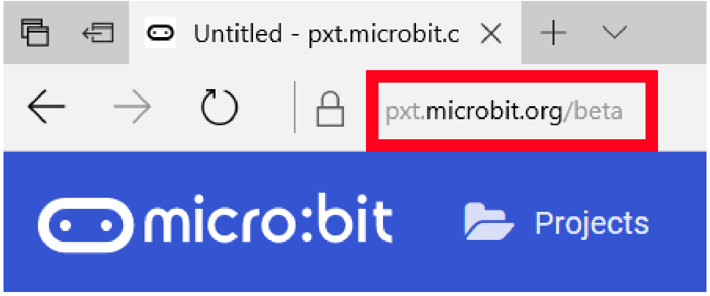
4 Creating your project

We're going to be using Microsoft's Block Editor for the Micro:bit. This is freely available and works entirely in the web browser. An "editor" is a thing which lets you write code.

Once we've opened the editor, we're then going to create a new project from our template code.

Actions

1. In your web browser
2. Go to https://pxt.microbit.org/beta - this is the latest version of the block editor for Micro:bit
3. Click the "Projects" button
4. Click "Import file"
5. Open the file you downloaded in the previous section (probably called "Microbit Virtual Circuits Starter.hex")



The URL of the beta Micro:bit block editor

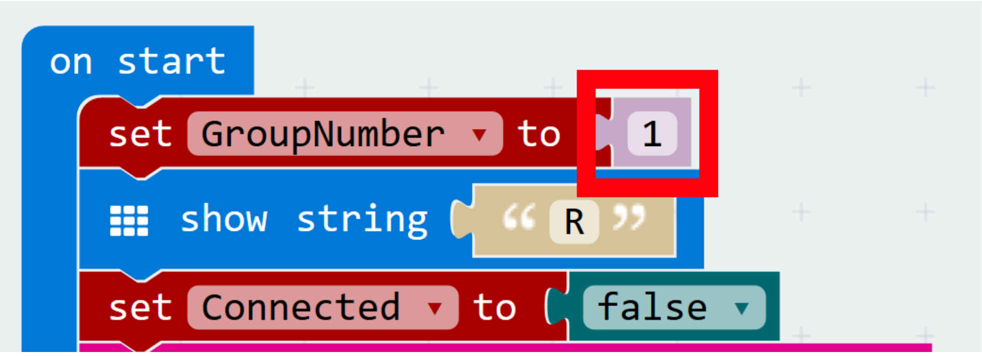
Notes
We use the Beta version of the editor because it has the latest features and bug fixes from Microsoft. The normal version doesn't support the Bluetooth communication used in this workshop.

5 Set your group number

You should be working in groups of between 3 and 6 people. Each group will have been given a number (probably your table number).

Actions

1. Look at the screenshot below. In your code, replace the number 0 in the red box with your group number.



```
on start
    set GroupNumber ▾ to 1
    show string "R"
    set Connected ▾ to false ▾
```

Set your group number to your table number

6 Create the component images

Our circuits are going to have 3 possible components: batteries, switches and lightbulbs (specifically, “Light Emitting Diodes” known as LEDs). Each of these components will be displayed on the Micro:bit once the component is connected to the circuit. Also, batteries and switches can be switched on and off, so we need two images for them - one for ‘on’ and one for ‘off’.

Notes

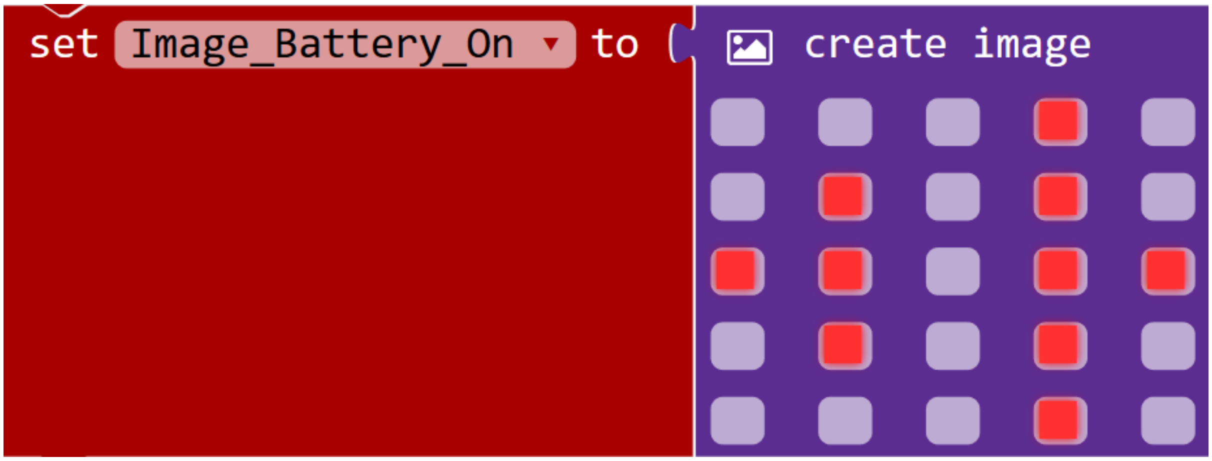
Your template code should already contain the blocks necessary for creating the images. The images haven’t been drawn yet though.

Notes

The code isn’t complete enough yet for these components to be drawn on the Micro:bit yet - we’ll see how to do that later.

Actions

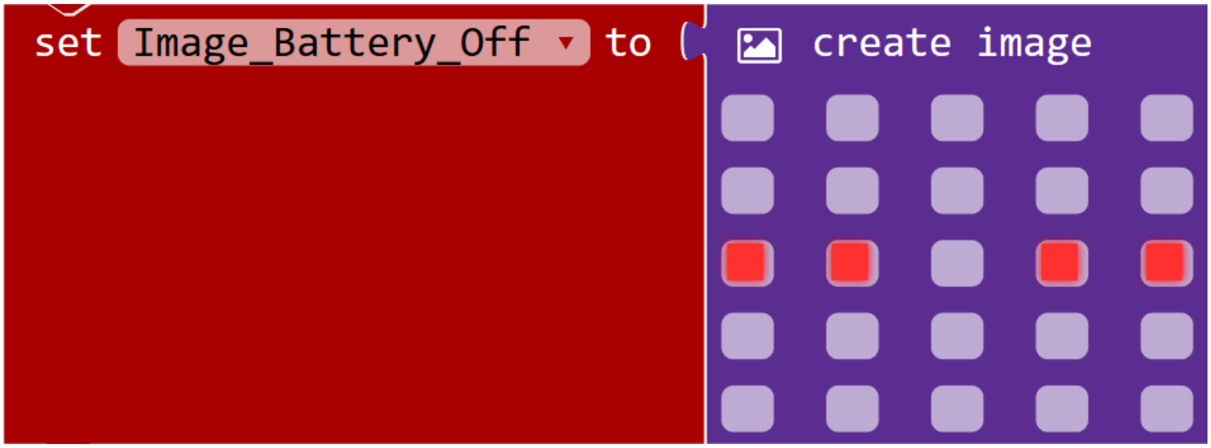
1. Using the screenshots (below and on the next few pages), draw each of the images in the existing “Create Image” blocks in your code.



```
set Image_Battery_On ▾ to create image
```

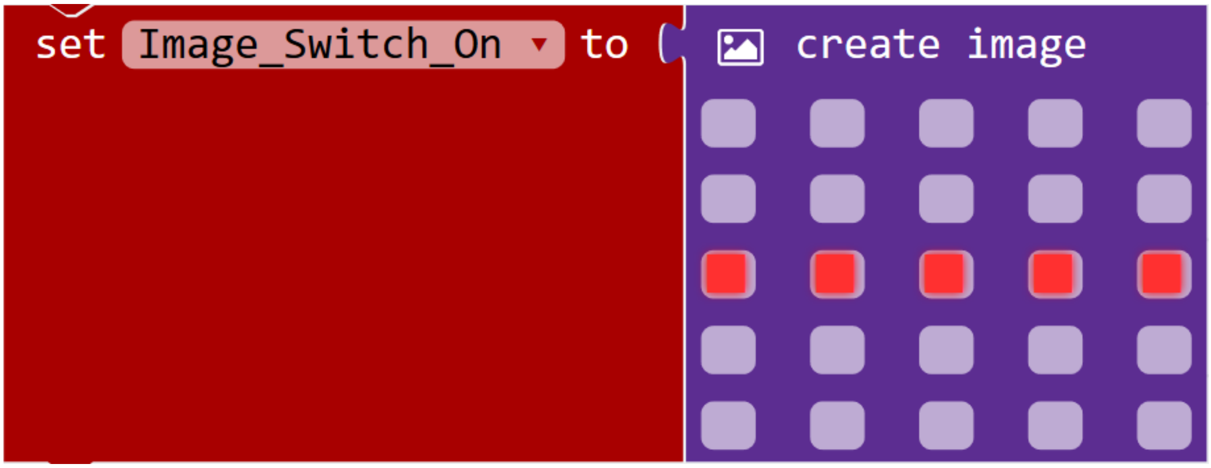
⬜	⬜	⬜	🟥	⬜
⬜	🟥	⬜	🟥	⬜
🟥	🟥	⬜	🟥	🟥
⬜	🟥	⬜	🟥	⬜
⬜	⬜	⬜	🟥	⬜

The image for a battery when it is on



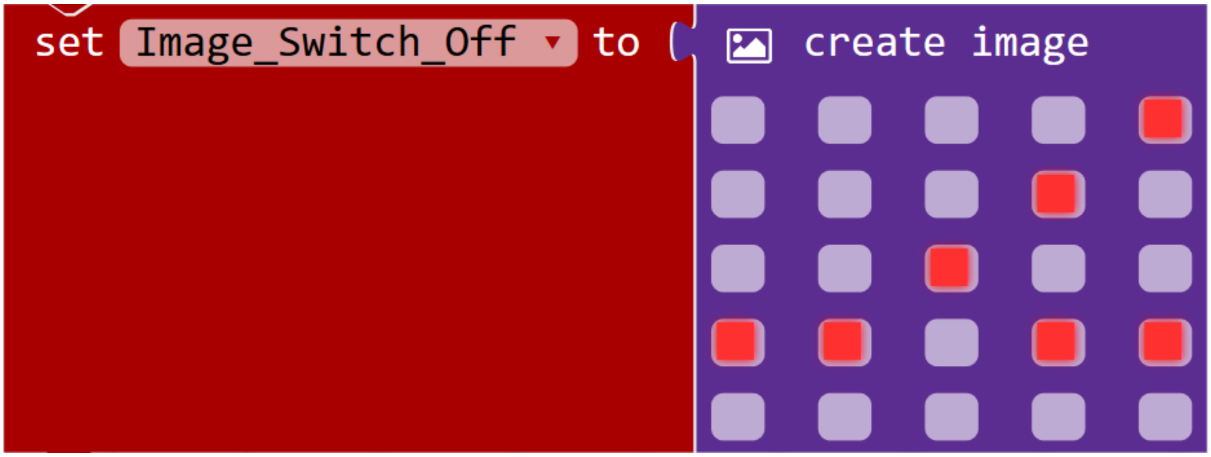
set Image_Battery_Off ▾ to create image

The image for a battery when it is off



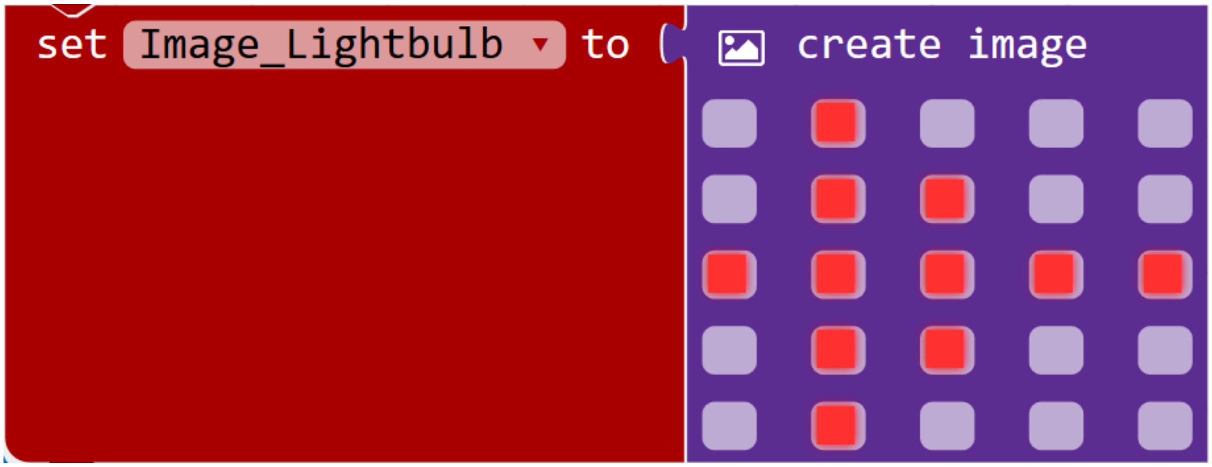
set Image_Switch_On ▾ to create image

The image for a switch when it is on



set Image_Switch_Off ▾ to create image

The image for a switch when it is off



set Image_Lightbulb ▾ to create image

The image for a lightbulb - an LED: Light Emitting Diode

Notes

The images are saved in **variables**. Variables are things which can save a value to be used later. We can change the value at any time, and retrieve it again at any time. They're really useful for storing things like numbers, text and images!

7 Drawing components

Now that we've created images for each of our components, it's time to work out how to draw them. This is the biggest bit of code in this workshop - so take care, there's plenty of time.

Notes

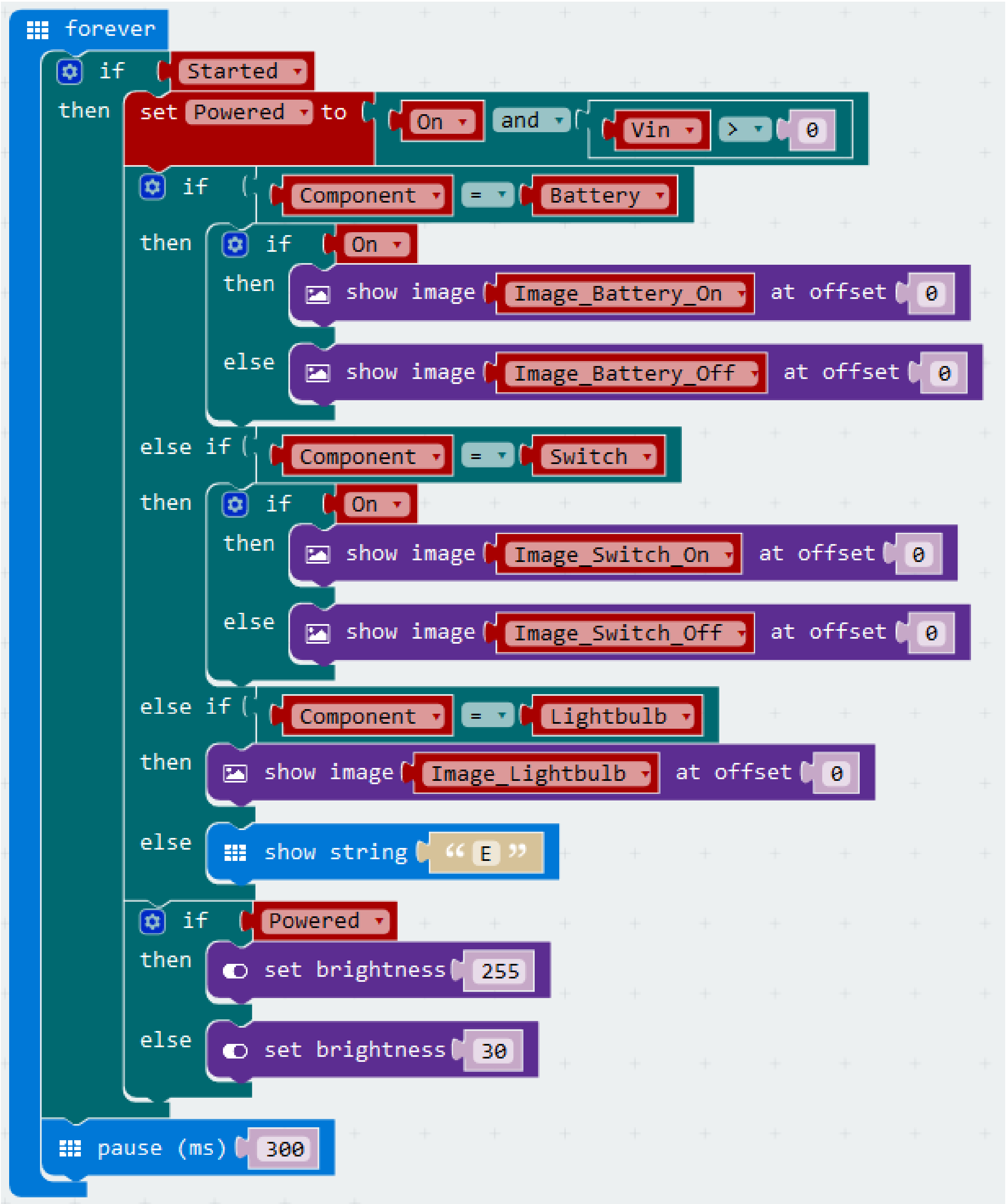
Some of the code for drawing already exists - you just need to fill in the spaces!

Actions

1. In your code, find the existing blocks that look similar to the first two in the screenshot on the next page.
2. Add the "if/else" blocks (These can be found in the "Logic" tab).
3. To create an "elseif" section, click the cog icon on an "if/else" block then drag an "elseif" block into the list.
4. Add the remaining blocks from the screenshot to your code.
5. The "Show Image" block is under "Advanced" then "Images".
6. The red/brown blocks (that have names like "Component" and "Battery") are variable blocks and so are found in the "Variables" tab
7. The "Show String" block is in the "Basic" tab
8. The "Equals" block is in the "Logic" tab
9. The "Set Brightness" block is in the "More" section of the "LED" tab

Notes

The colour of the blocks matches the colour of their tab! But don't forget to check the "Advanced" section for more tabs/blocks.



```
forever
  if Started
  then
    set Powered to On and Vin > 0
    if Component = Battery
    then
      if On
      then
        show image Image_Battery_On at offset 0
      else
        show image Image_Battery_Off at offset 0
    else if Component = Switch
    then
      if On
      then
        show image Image_Switch_On at offset 0
      else
        show image Image_Switch_Off at offset 0
    else if Component = Lightbulb
    then
      show image Image_Lightbulb at offset 0
    else
      show string “ E ”
    if Powered
    then
      set brightness 255
    else
      set brightness 30
  pause (ms) 300
```

Code for drawing the components

Notes

Although our code can now draw our components, we won't see anything happen until later. This is because they aren't drawn until after our Micro:bits are programmed and connected together.

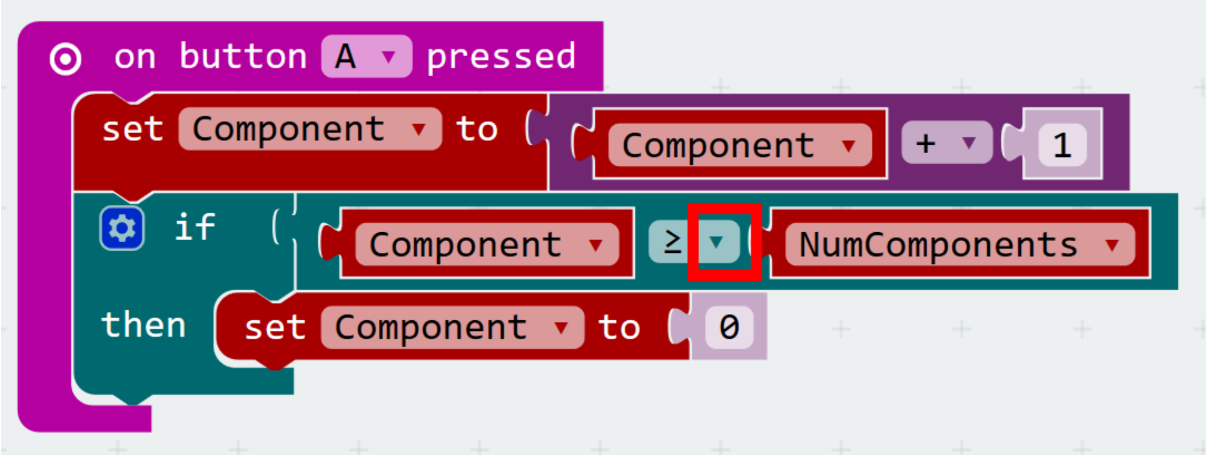
8 Switching between components

Now that we can draw the components, we need to work out how to switch between them. Fortunately, this is fairly simple.

Whenever button A is pressed (the left hand button) the component is going to switch to next in the list. If the end of the list is reached then we'll go back to the beginning again.

Actions

1. Find the existing "On Button A Pressed" block in your code
2. Add the blocks shown in the screenshot.
3. The "Set Item" block is in the "Variables" tab - this stores a new value in a variable, overwriting what was there before.
4. The "$+$" block is in the "Math" tab.
5. The "$\geqslant$" block can be found by adding a "$<$" block from the "Math" tab, then using the small, green arrow on the block to open the list and change the symbol.



```
on button A pressed
    set Component to Component + 1
    if Component ≥ NumComponents
    then
        set Component to 0
```

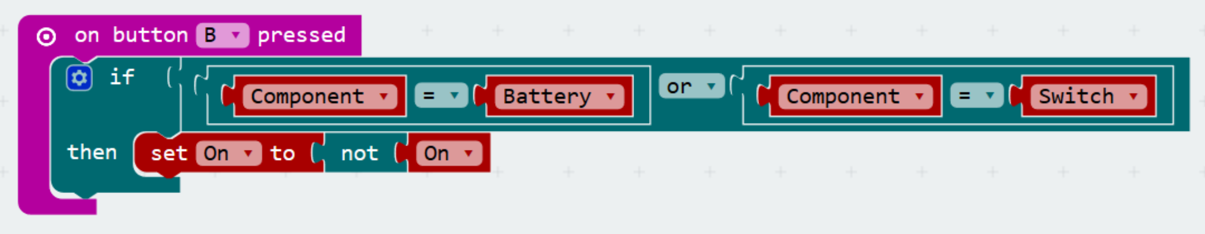
Code for switching between components

9 Switching a component on and off

So now we can change components, there's just one last thing to do - turning batteries and switches on and off! For this, we will use button B. Whenever it is pressed, if our component is a battery **or** a switch *(but not an LED)* we will toggle whether the component is on or not.

Actions

1. Find your existing "On Button B Pressed" block
2. Add the code in the screenshot below.
3. The "or" and "not" blocks are found in the "Logic" tab.



```
on button B pressed
    if ( Component = Battery ) or ( Component = Switch )
    then set On to not On
```

Code for switching a component on and off

Goals

You should now have all the code needed to make your Micro:bit work! Ask your teacher or one of the helpers to check it for you.

Your Micro:bit won't start working yet though - we need to save your code to your Micro:bit and then connect all the Micro:bits in your group together wirelessly. We'll show you how in the next 2 sections.

10 Saving to your Micro:bit

Great! Your code is ready - let's save it to your Micro:bit.

Actions

1. If you haven't already, find your Micro:bit and plug it into your computer using the USB cable.
2. In the editor (your web browser) click the big Download button
3. Save the file to your Micro:bit, by clicking on the Micro:bit in the left-hand column, then click Save.

11 Connecting the components of the circuit

You need to wait for all of your group to finish. Once all your group's Micro:bits are ready, follow the instructions below - remember to work together and slowly, otherwise it won't work properly.

Actions

1. Sit in a circle (at the table or on the floor). Face inwards with your Microbits.
2. Plug the batteries into your Micro:bits
3. Choose one person to go first (you'll all get to do this next step in a moment)
4. Whoever is going first, press **both** the A and B buttons together then let go of them. Everyone else's Micro:bits should now show the letter "C". If one of them doesn't, please ask your teacher or a helper for assistance.
5. Now, one by one around the circle, press both A and B together (just once each!) until you get back to the first person (but the first person doesn't press again yet!). The person who went first should now see a letter S on their Micro:bit. Everyone else should now see the letter P on their Micro:bit. If you see only a Q or anything else or the screen is flickering between letters, please stop and ask your teacher or a helper for assistance.
6. Now the last step - everyone press both A and B together one final time!
7. You should all now begin to see your components appear. Whoever went first will see a battery. Everyone else will see an LED - but you can change component using button A. Remember, you can turn off batteries and switches using button B.
8. The virtual circuit goes round the circle in the same order you connected the Micro:bits by pressing A and B. So if someone turns off a switch before you in the circle, your component will not be powered!

Questions

1. Experiment with different components at different places in the circle - can you find any combinations that don't match what a real circuit would do?

12 Wrap-up

We hope you enjoyed this workshop! This workshop has hopefully taught you more about how to program the Micro:bit and some of its wireless capability.

We also run this workshop in schools so if you'd like to see it in your school, please recommend it to your class teacher and get your parents to connect us with them!