adafruit learning system

BBC micro:bit Lesson #0

Created by Carter Nelson

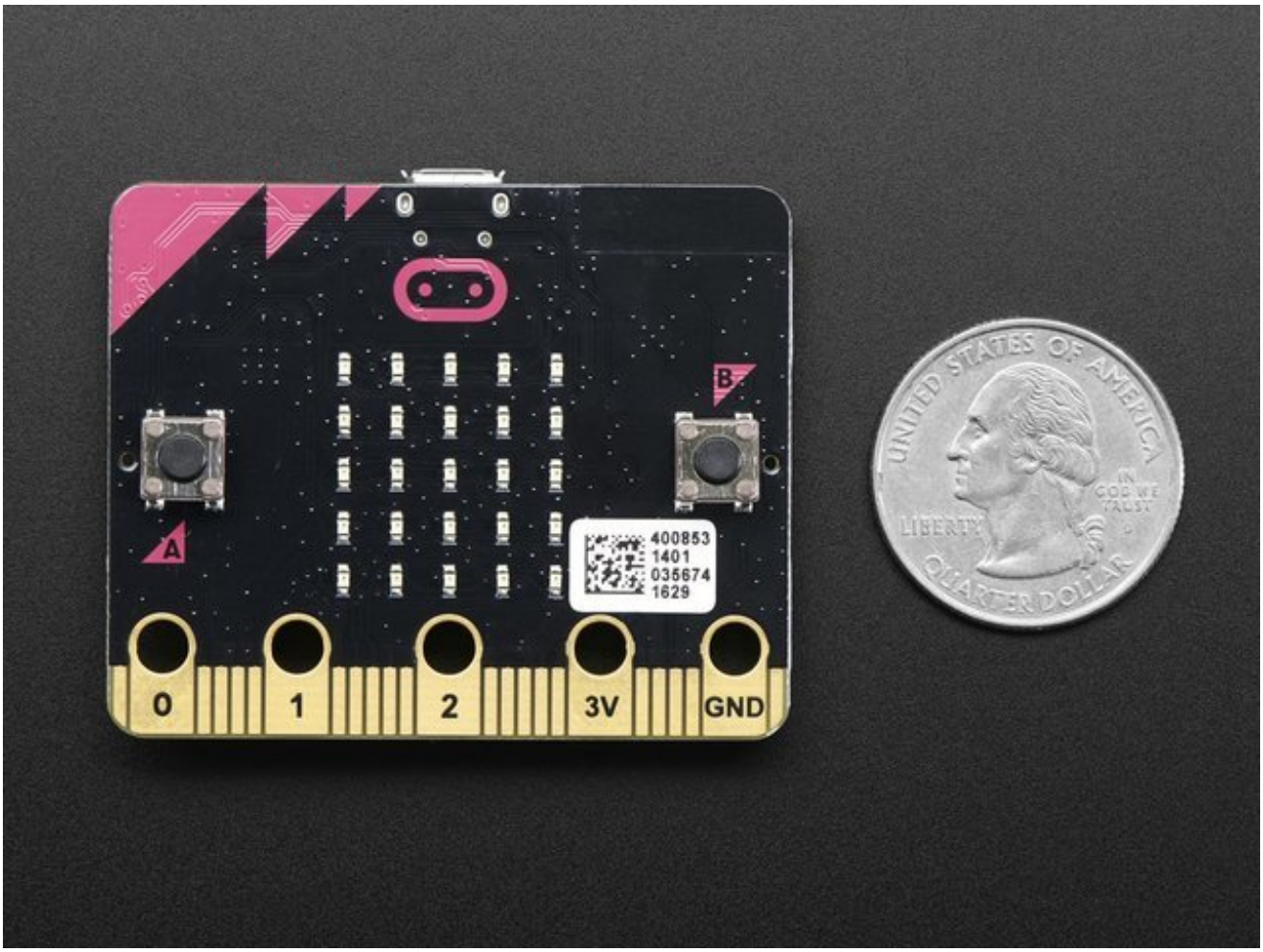
Last updated on 2017-08-30 10:49:03 PM UTC

Guide Contents

Intro

So what is this unassuming little board? It's name is the BBC micro:bit (http://adafru.it/yEK), and it is a little computer you can write programs for. It has various gadgets like buttons and lights that you can use to do all kinds of fun stuff. There's even a compass!

In this guide we'll briefly go over the main features of the BBC micro:bit, just so you have an idea what's going on. The main web site is at www.microbit.org (http://adafru.it/yEL) where you can go to find more information.



The BBC part of the name stands for British Broadcasting Corporation (http://adafru.it/yEM). There's a lot of history (http://adafru.it/yEN) to why the BBC would get involved with making a little computer. But the short version is they did. And then they gave it away to every year 7 student in the British schools.

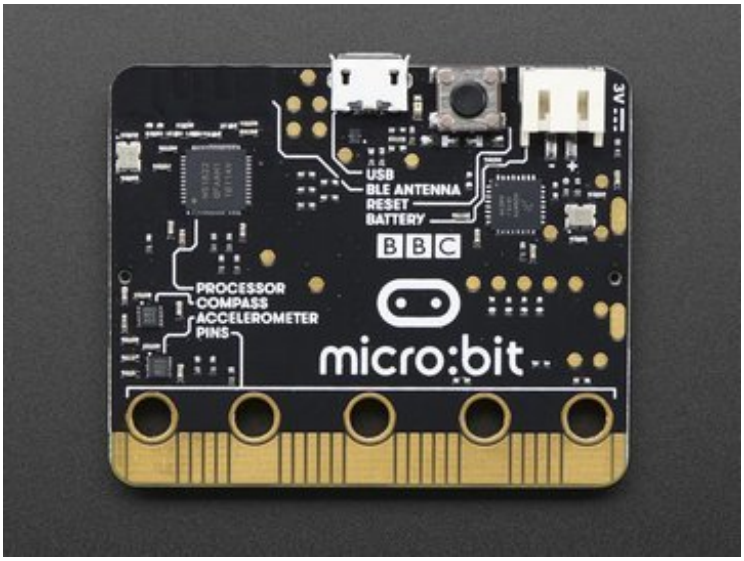
And now the little computer is making its way across the pond and into America. That's

right, it's sorta like a British invasion.

Lesson Parts

Here's a run down of what you'll need to play with the BBC micro:bit.

Required:

- 

  A **BBC micro:bit!** Of course, this is essential, and is required for all lessons.

  Available at Adafruit as either a bundle (http://adafru.it/yEO) or just the board (http://adafru.it/yEP).

- 

  Micro USB Cable any length. One comes with the bundle. Or you probably have one of these around, they're the most common USB cable.

  Make sure its a data/sync cable!

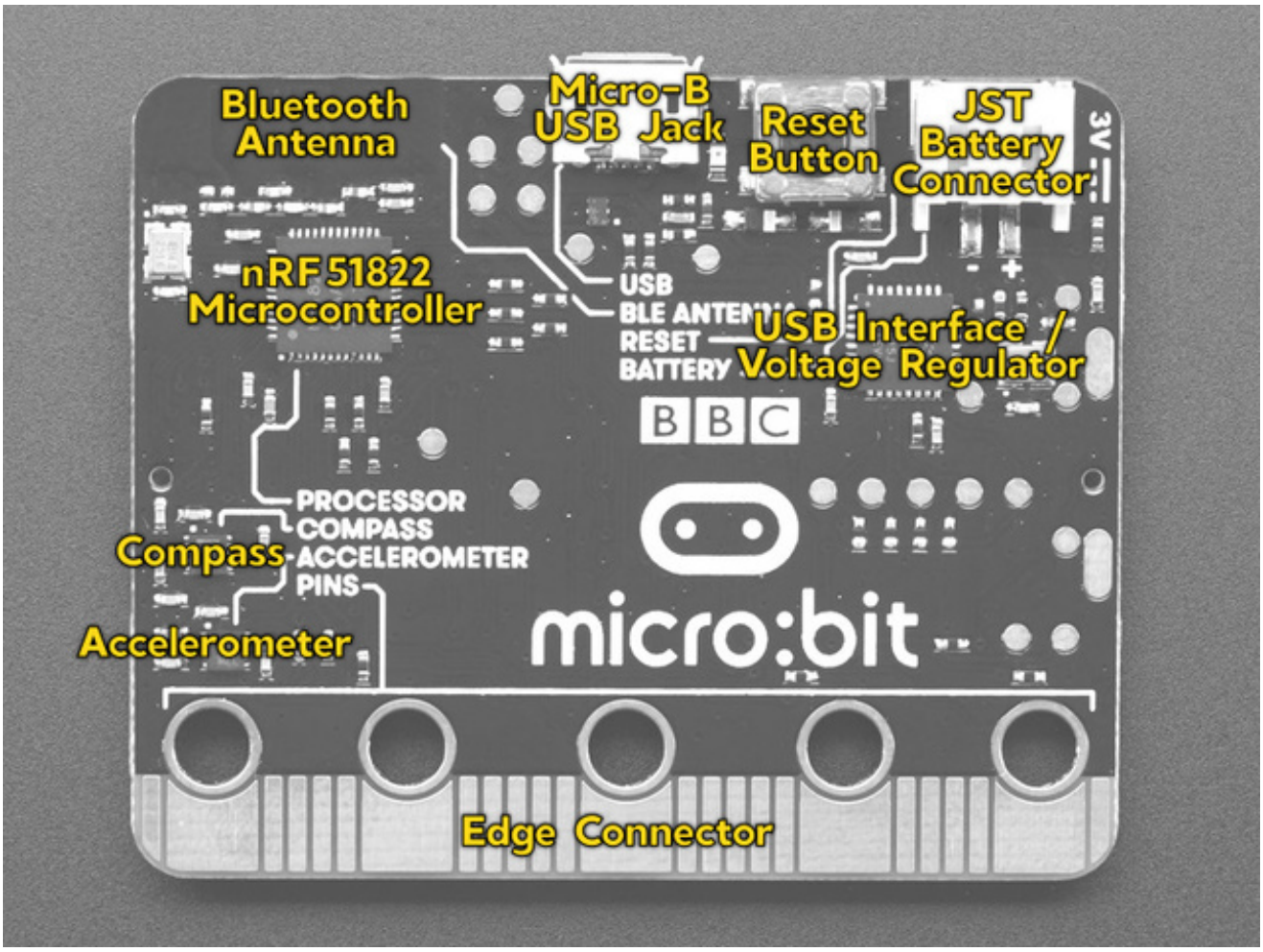
  USB cable available at Adafruit (http://adafru.it/592)

The BBC micro:bit Go Bundle comes with a 2xAAA battery holder and batteries which work

fine. However, there is no on/off switch.

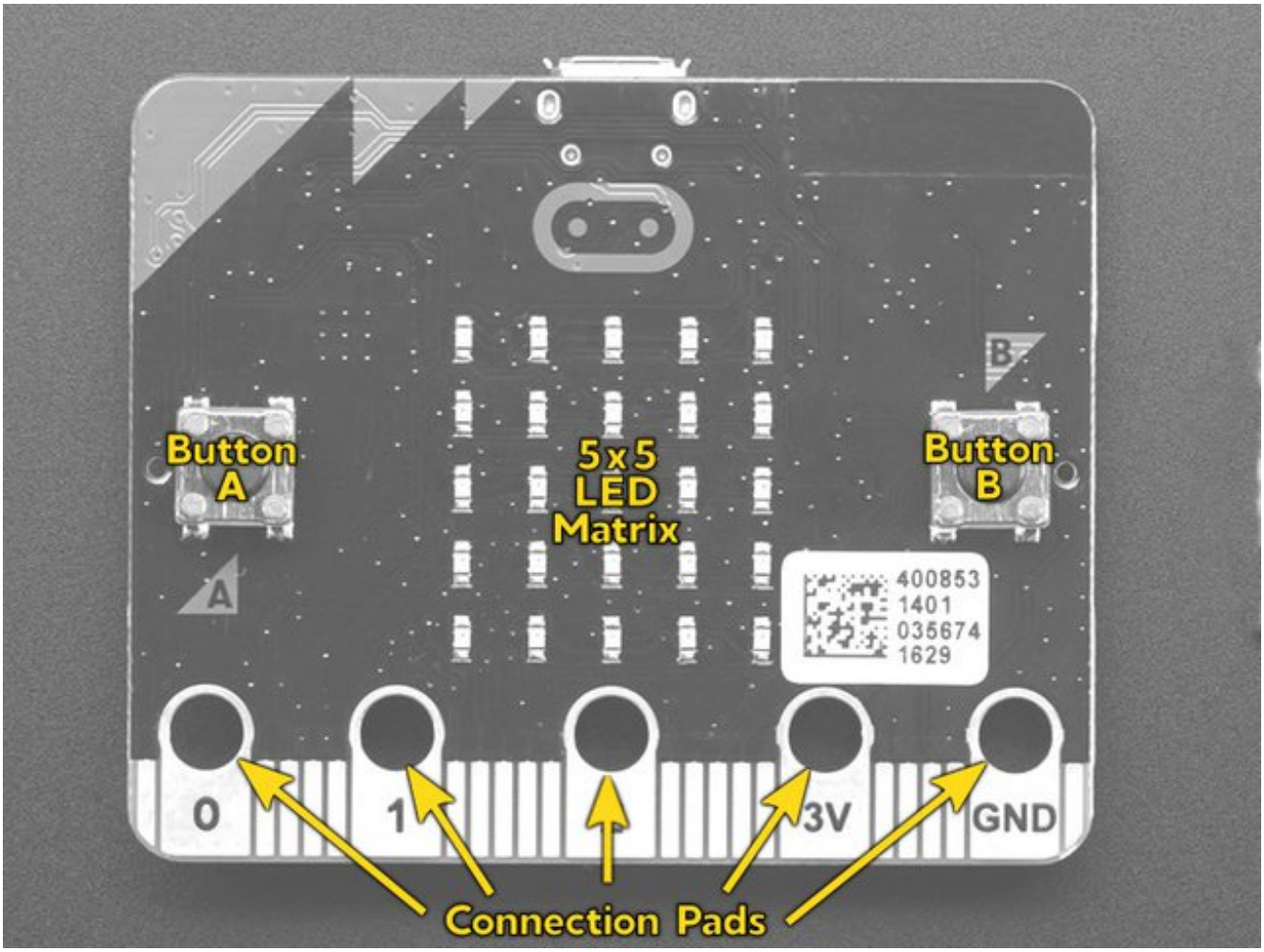
Take a Tour!

So let's take a look at this little doodad and see what it's got. There's stuff all over both sides of this thing. Not sure if anyone knows which is the front and which is the back, but here's what's on one side.



Compass? Accelerometer? Bluetooth? Neat.

But the other side may be even cooler. It's got two buttons which are useful. But the really cool thing is the 5 x 5 grid of little LED lights. That's 25 total LEDs. And each one can be turned on and off individually. You can even set the brightness!

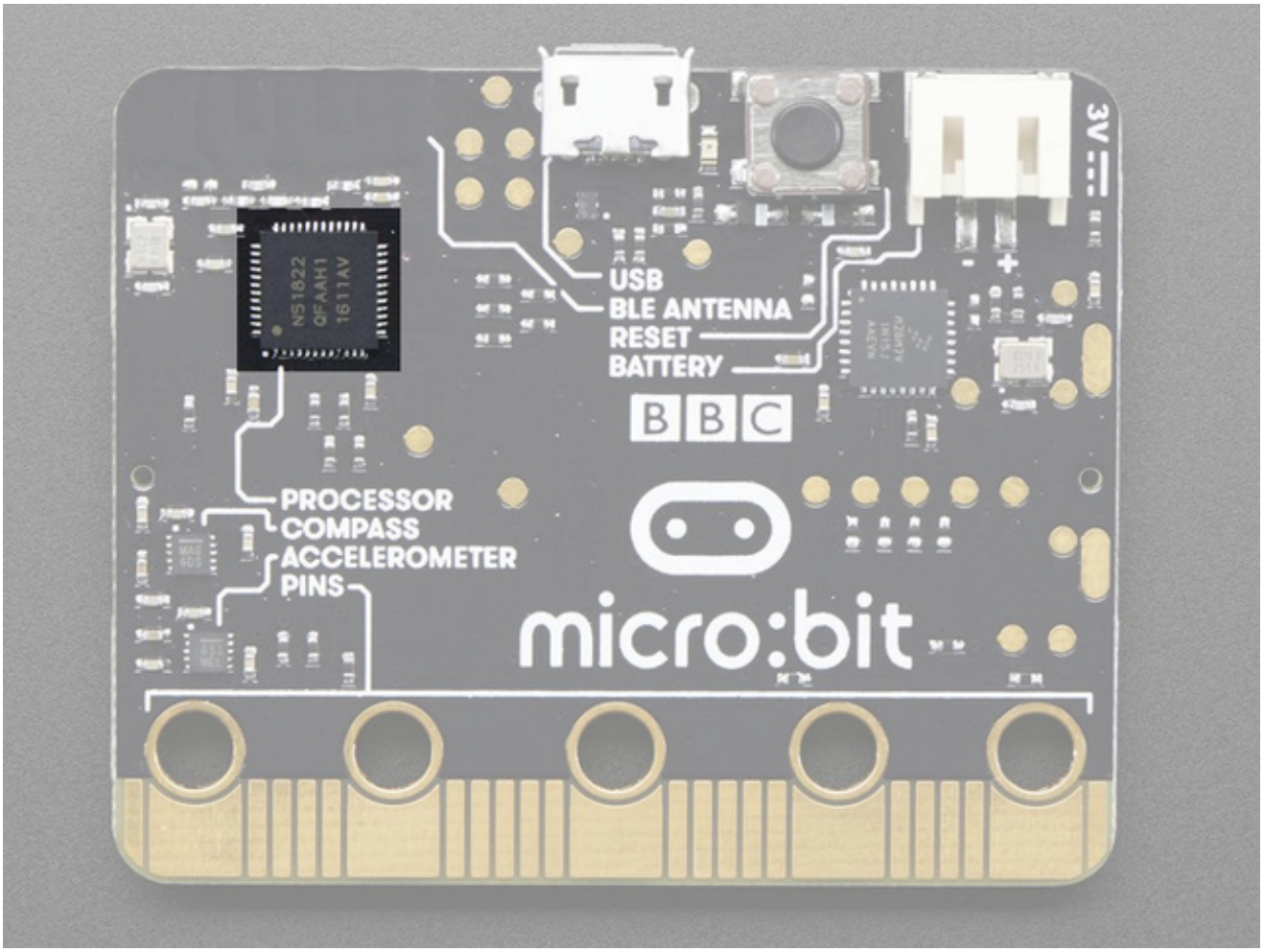


Along the bottom edge are a bunch of gold connection pads. The most important of these are the 5 bigs ones - the ones with the big holes. These will allow you to attach all kinds of additional items using alligator style clips. No soldering needed!

Now let's look at all this stuff in more detail. That is, if you want to. If not, go ahead and skip on to the Let's Code section to learn about programming. That's where the fun is at!

Microcontroller

This little chip, called a microcontroller or processor, is the brains of the BBC micro:bit. It's what actually runs your programs.



This particular microcontroller is made by a company called Nordic Semiconductor and has the cryptic name of nRF51822. Did some one say Nerf? Wah? Meh, it's just what they call it. It's probably 'n' for Nordic, and 'RF' for radio frequency or something. Don't worry about.

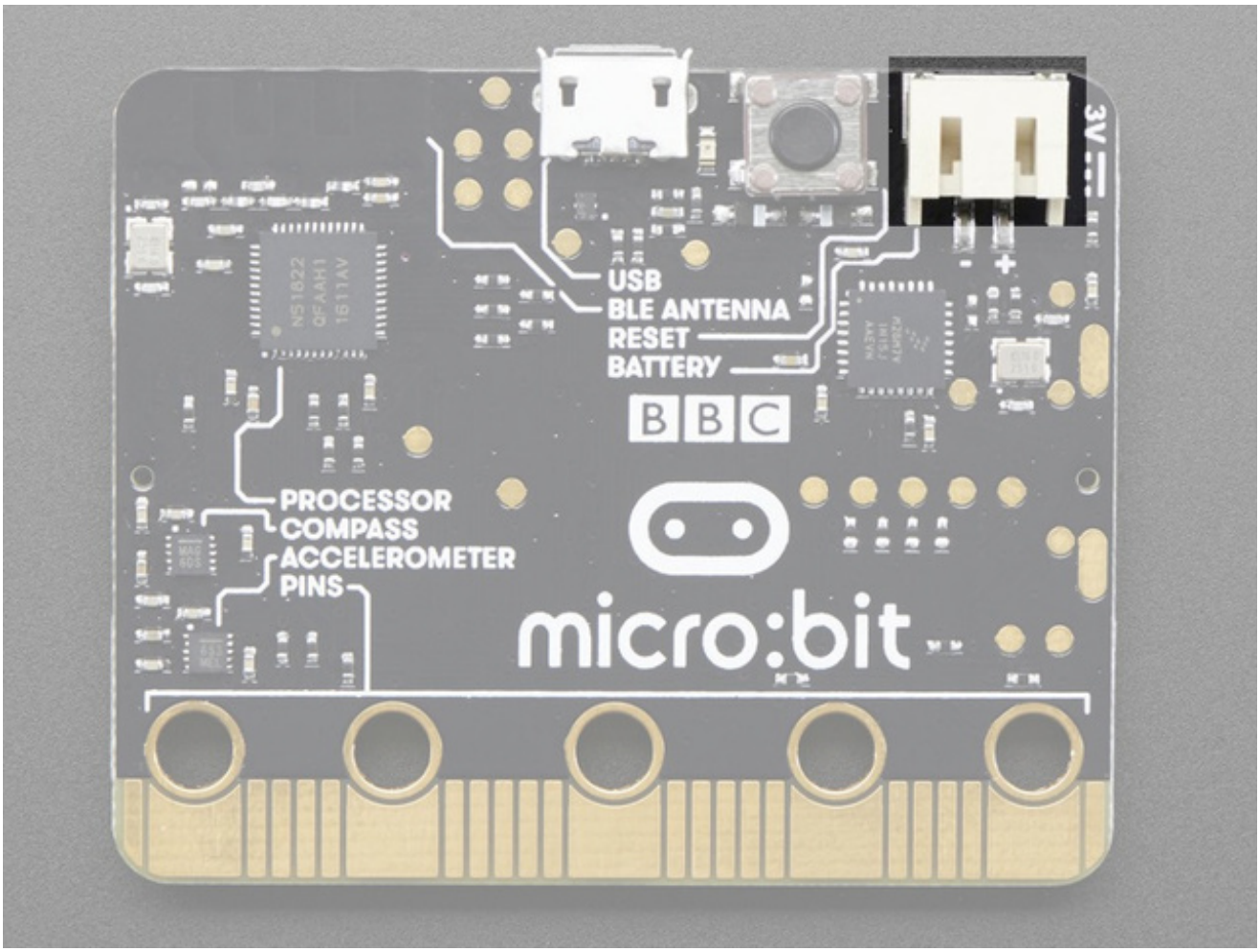
You also don't need to know these details, but it's kind of fun to geek out about them:

- It's a 32-bit ARM Cortex M0 running at 16MHz
- It has 256KB of flash memory
- It has 16KB of ram
- It has a 2.4 GHz Bluetooth low energy radio

Battery Jack & Supply

You can provide power to the BBC micro:bit two ways - through the USB cable or via battery. The main way to program the BBC micro:bit is through the USB cable. However, once you've uploaded your program, you can use battery power to make your project mobile.

Here's where you connect the battery pack:



This little plastic connector is called a[JST connector](http://adafru.it/yEQ). The battery pack that comes with the Go Bundle has the correct part to plug in there. The battery packs that Adafruit sell also come with the correct connector. Just plug and go.

Be careful unplugging though. The wires are not super strong and you shouldn't pull the plug out by yanking on them. Try and grab the edge of the connect plug as best you can.

Choosing Battery Power Supply

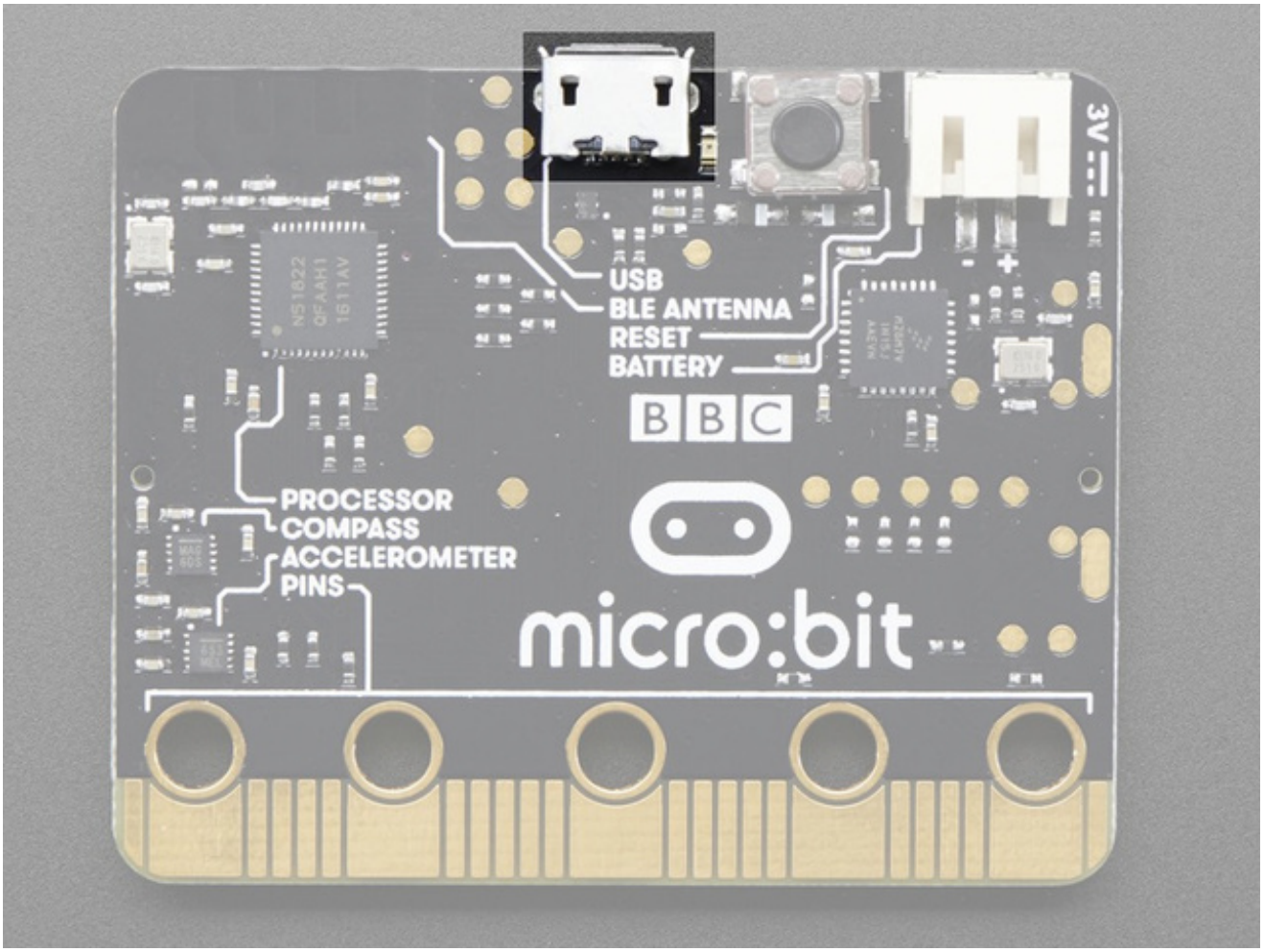
Powering via battery is a little different than powering via USB. The on board voltage regulator is bypassed, so you need to pick your battery supply carefully. See here for some guidance:

http://tech.microbit.org/hardware/powersupply/ (http://adafru.it/yKE)

(thanks David Hamilton from Scotland for pointing this out!)

USB Connection

This is where you plug in the USB cable for programming. It is also one of the two ways of providing power to the board.



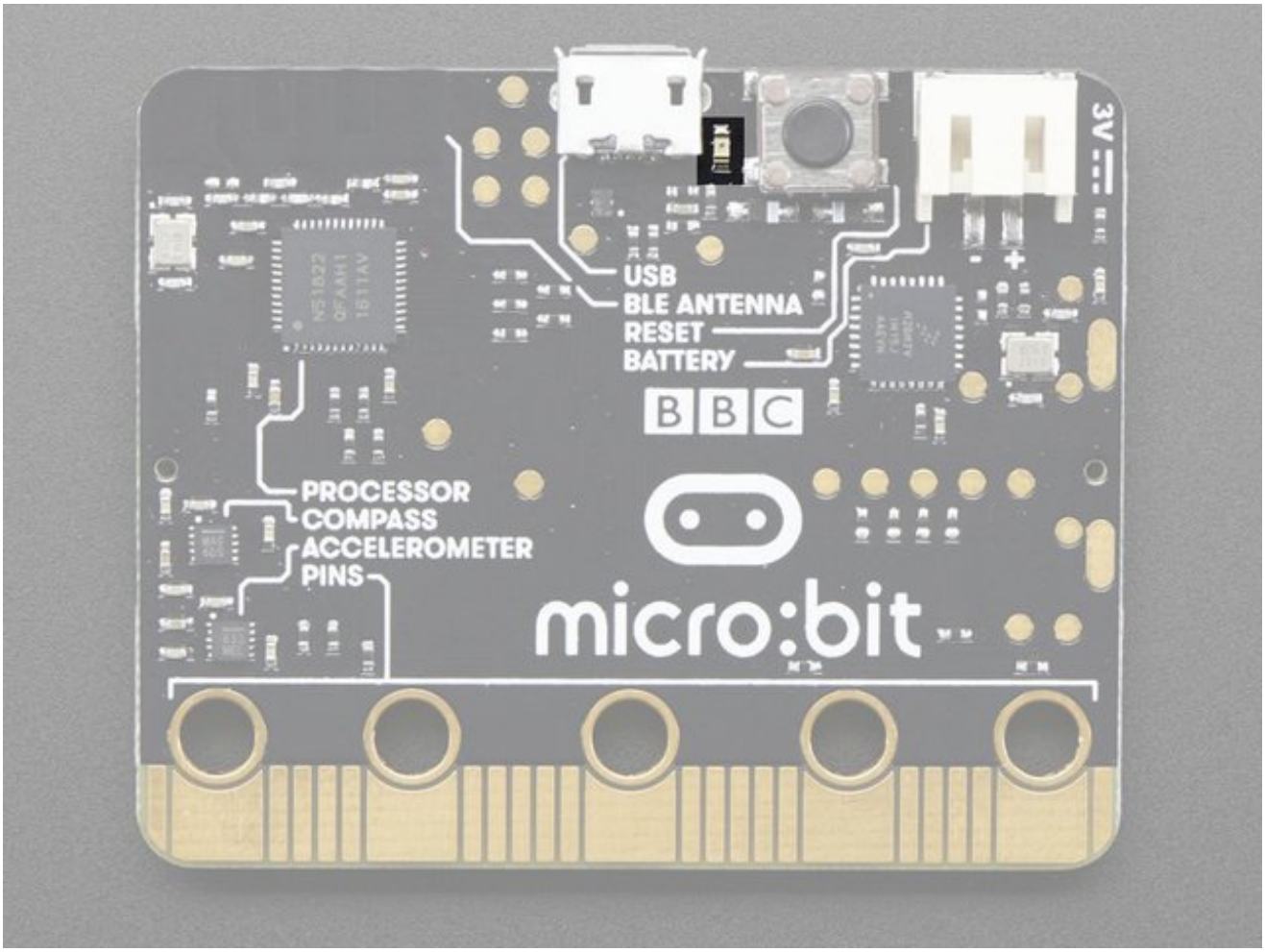
One end of the USB cable will attach here while the other end will got to a USB port on the computer you use to write your programs. We'll talk about how to actually do all that later.

If you are using the USB cable from the Go Bundle, then you're set. If you are using one of your own USB cables, make sure it is a good one. While the world is full of USB cables, not all of them are made equal.

A HUUUUUUGE number of people have problems because they pick a 'charge only' USB cable rather than a "Data/Sync" cable. Make 100% sure you have a good quality syncing cable. Srsly, I can't even express how many times students have nearly given up due to a flakey USB cable!

Status LED

So small you may have missed it. But it's there. This little yellow LED will turn on and sometimes do a little blink dance to let you know stuff is happening. Just let it do its thing.

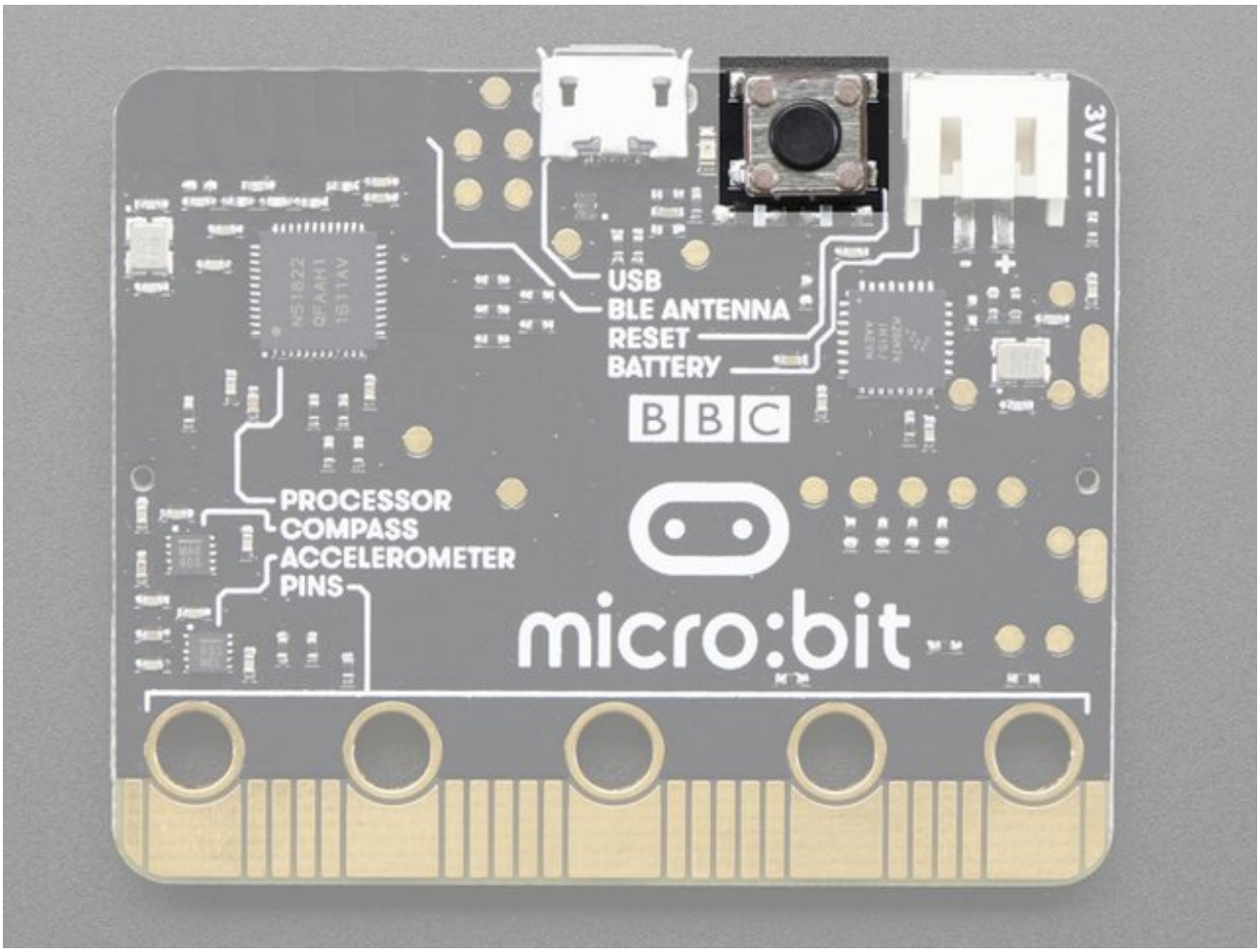


The status LED will NOT come on when powering via the battery jack. This is normal.

Reset Button

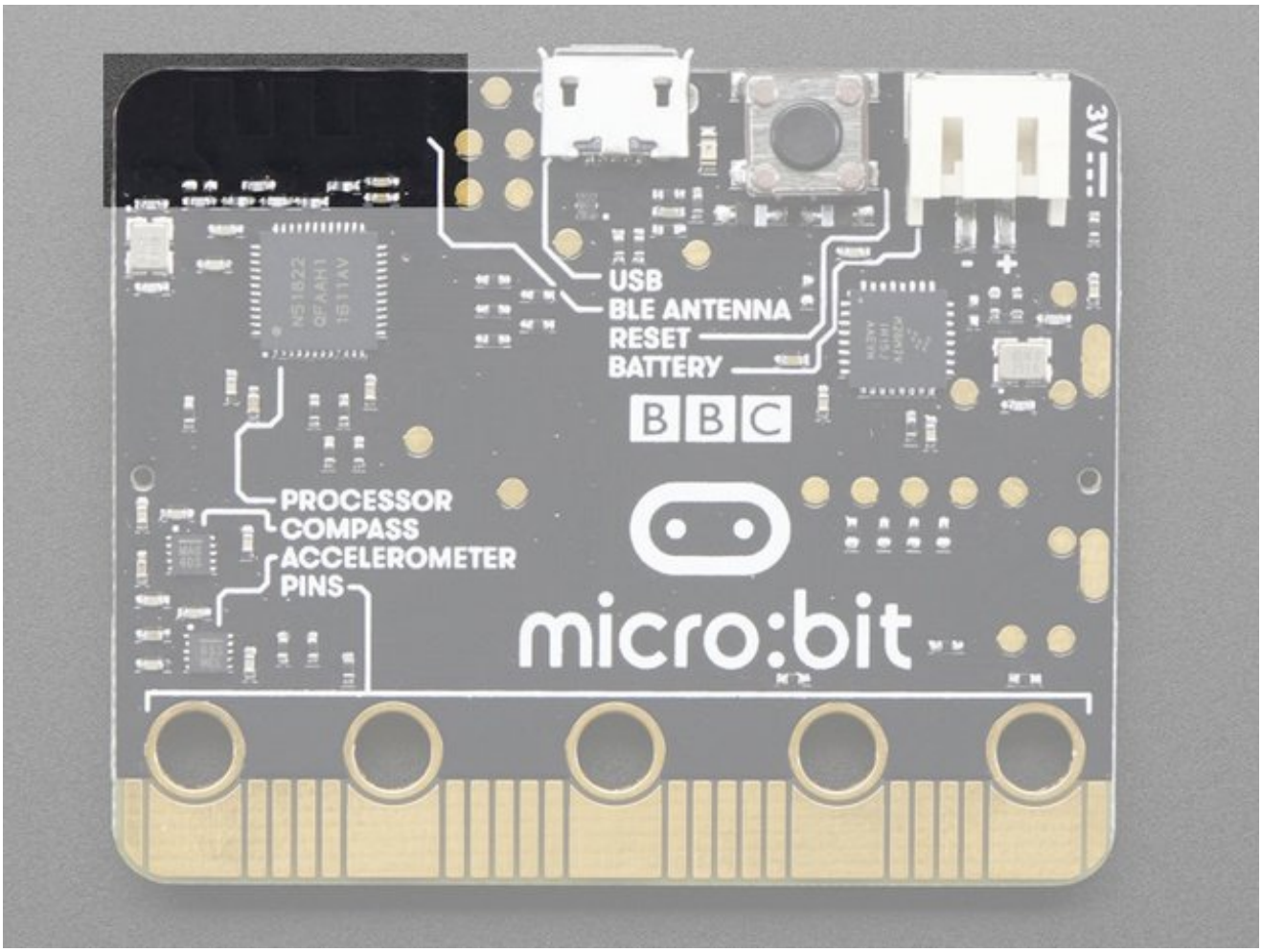
Sometimes you want to have your program start all over. That's what this reset button will do. You don't need to write anything in your program to use it, it will just happen. Press it and it will "reset" everything to the beginning.

Be careful though, maybe you don't want that to happen. That's why it's put on a different side than the other two buttons.



Bluetooth Antenna

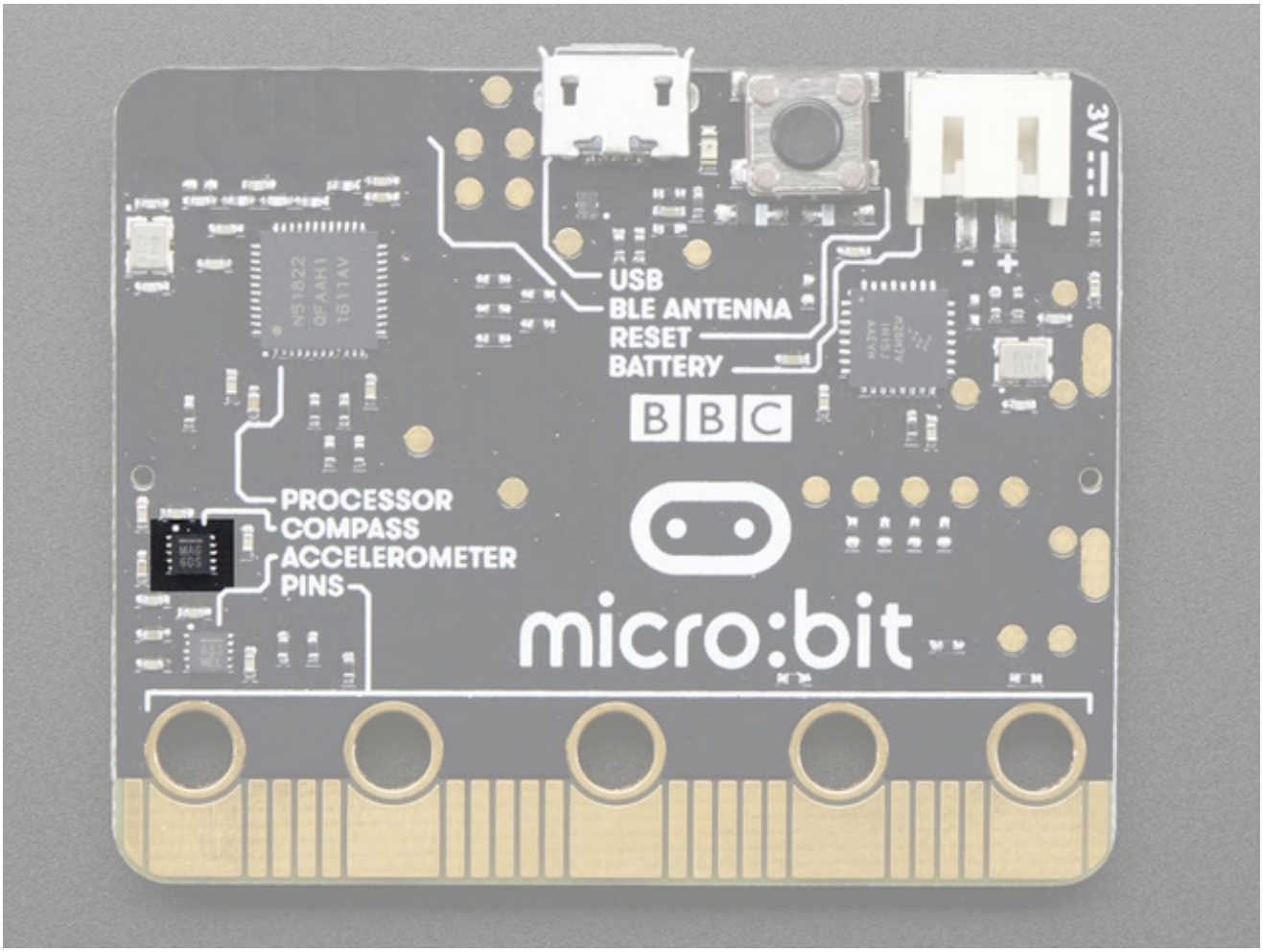
Just so you know where it is, the antenna used to communicate over Bluetooth is located here. You can kind of see it if you hold the board just right in the light. It looks like a square squiggle snake thing.



The actual hardware that does the Bluetooth communications is part of the main microcontroller (processor). This is just the antenna part, so maybe don't wrap that area in tinfoil or go drilling holes in it or something.

Compass

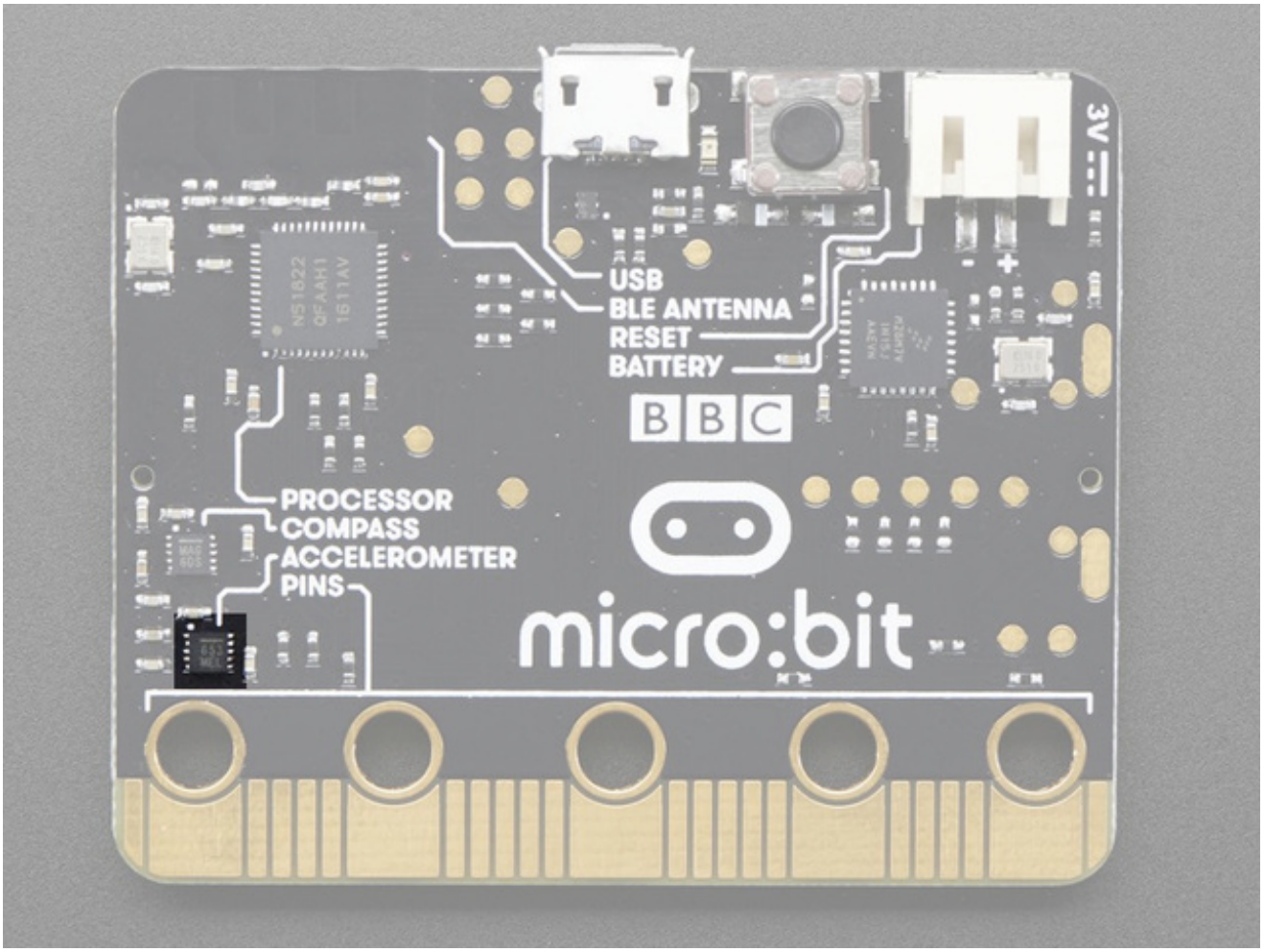
Which way is North? Well, a compass can tell you. And the BBC micro:bit has a little one right here:



It's actually more like a magnet sensor. The Earth has a magnetic field that you can use to find North. However, anything else with a magnetic field will also be seen by the compass and can throw off the readings. Lot's of electrical devices generate magnetic fields, so you want to be sure and be away from them if you want to find out where North is.

Accelerometer

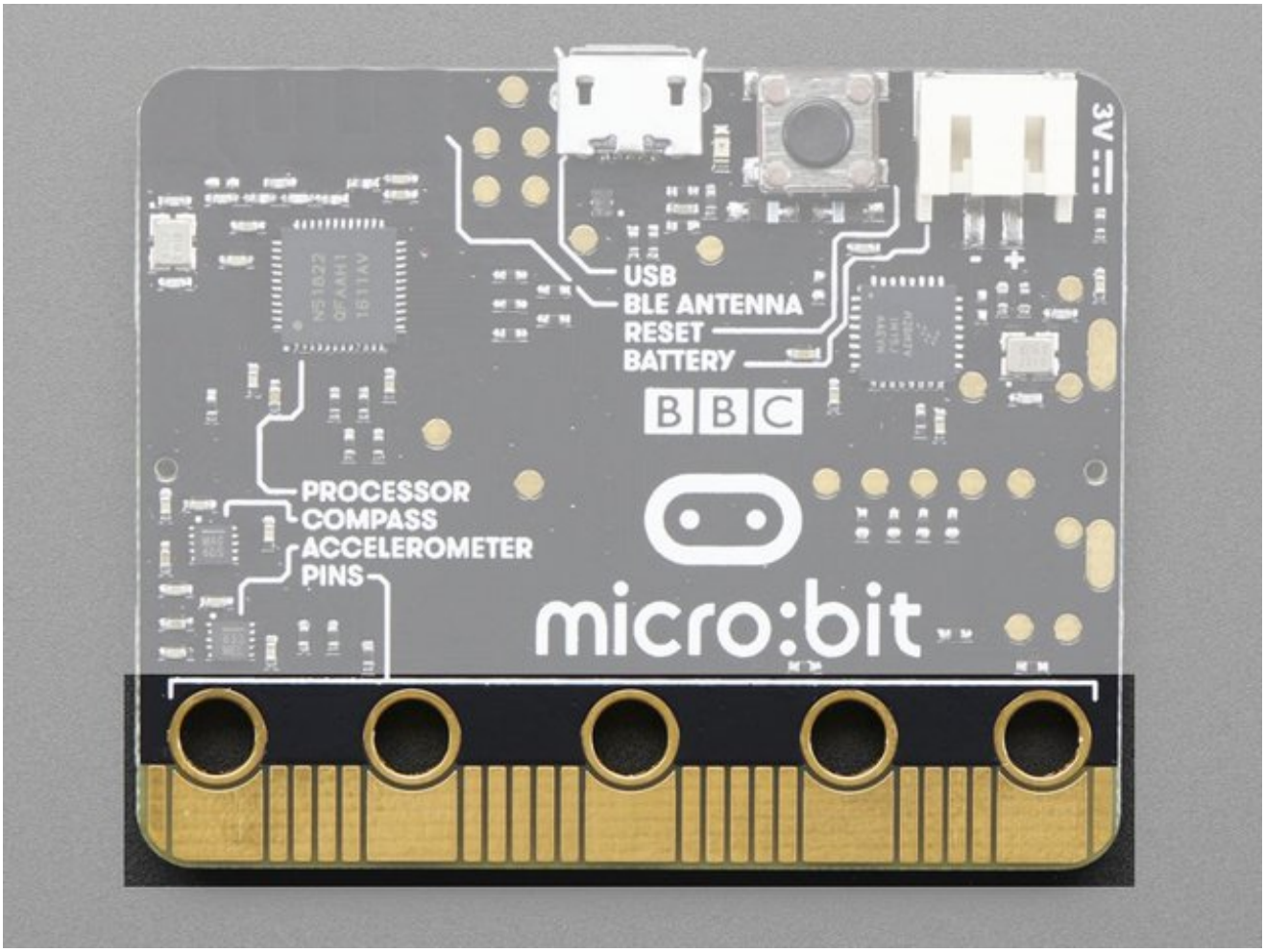
The traffic light just turned green. Hit the gas and accelerate! That's what this little sensor called an accelerometer will measure:



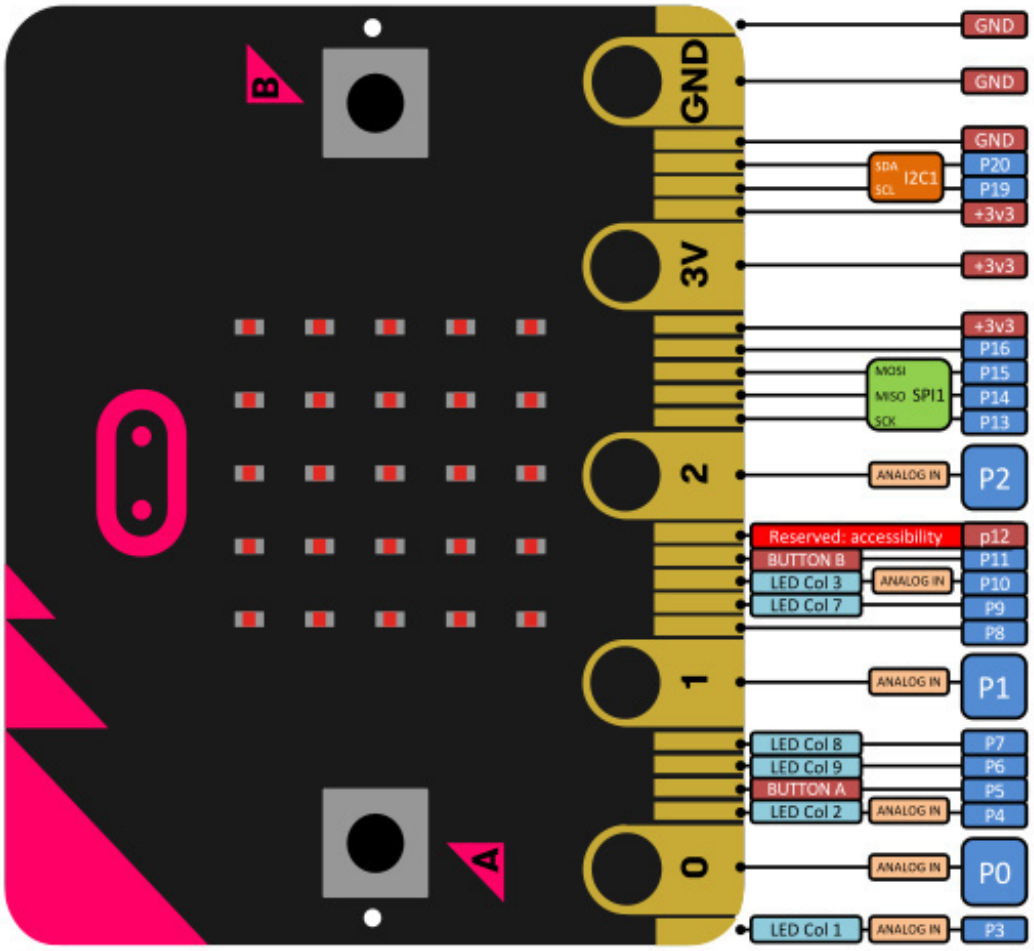
You can also generate acceleration by shaking the BBC micro:bit. This will let you do things like have parts of your program start only when the board is shaken. Like shaking dice or something.

Edge Connector

Look at all those gold teeth! Yep, there's a lot of connections along the bottom edge of the BBC micro:bit.



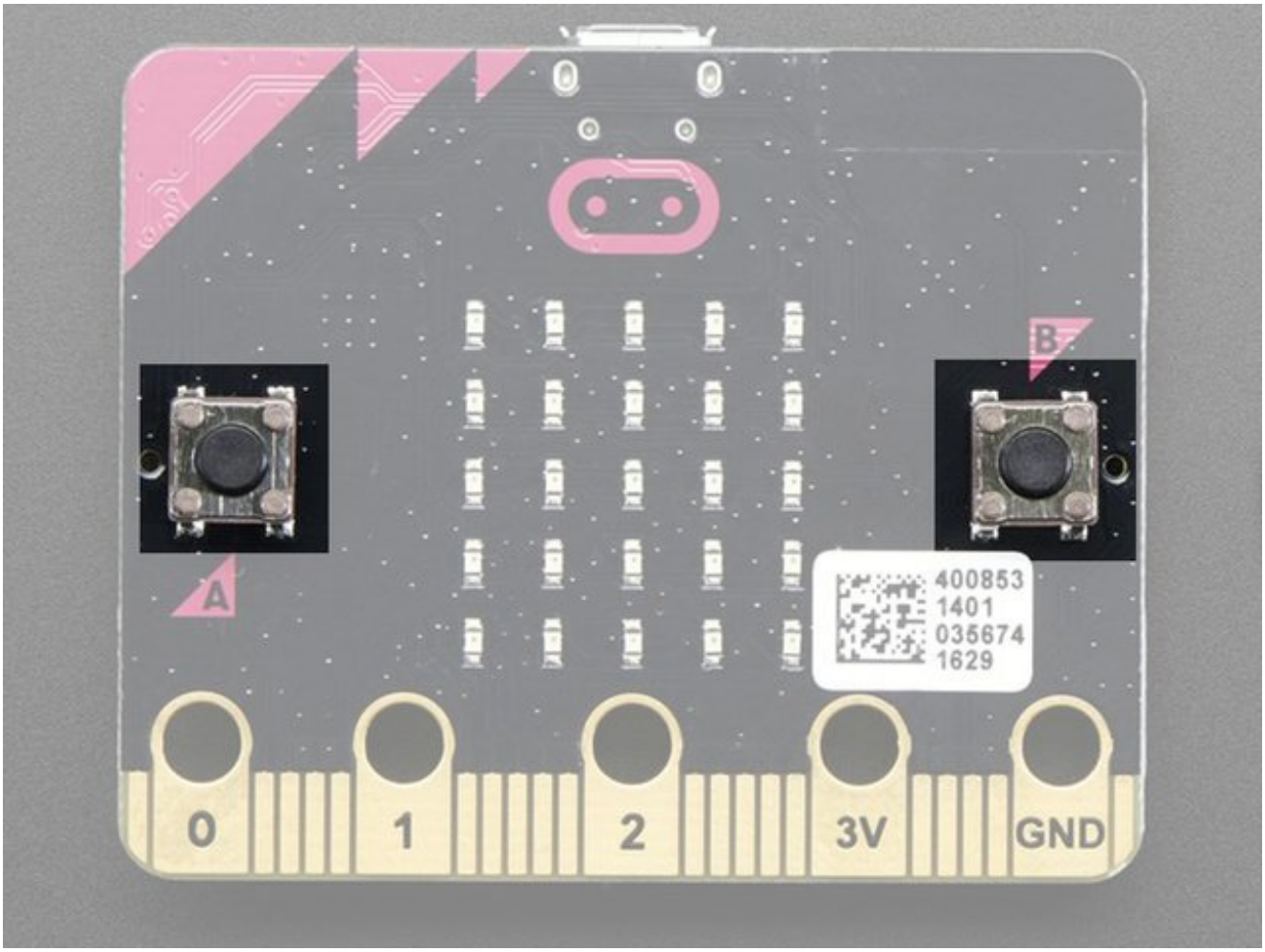
They do all kinds of crazy things as you can see in the image below. You can even get a special connector (http://adafru.it/yER) to make using them easier. The big brain of the BBC micro:bit can do a lot of things, and they just wanted to make sure there was a place to get access to them.



Pretty much every program will just use the 5 pads with the big holes. So for now, don't worry about all those other little ones.

Buttons

Buttons are super useful, and the BBC micro:bit has two of them right here. One's called A and one's called B. There's a little label next to them that shows this.

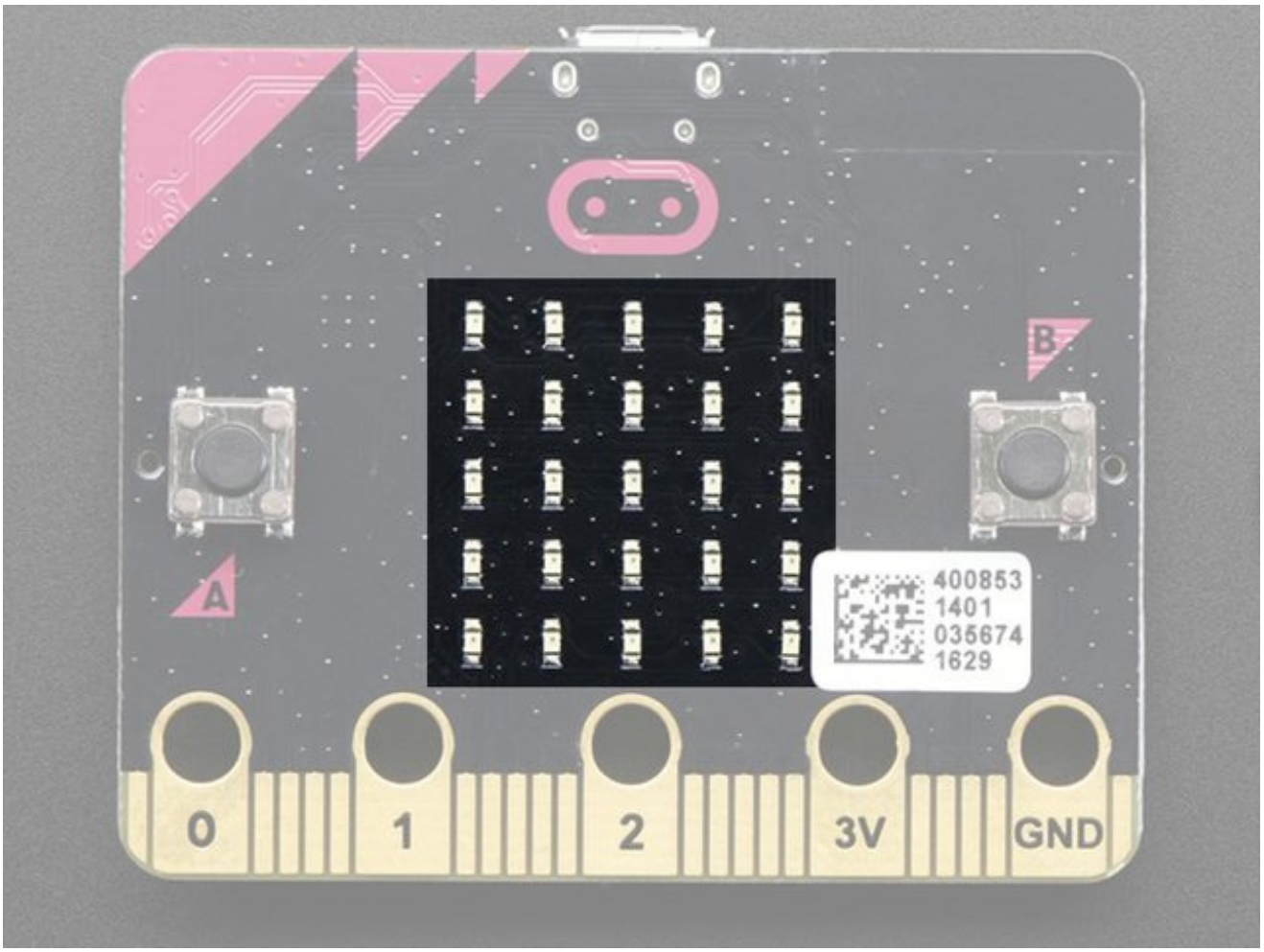


These are your primary form of input. This is how you'll tell the BBC micro:bit to do something or start some action. Like "Hey, I just pressed this button, now blink me some LEDs!" Maybe one button can be "start" and one can be "stop". Or "faster" and "slower". Or "brighter" and "dimmer". It's up to you.

LED Matrix

OK, this is probably the coolest feature of the BBC micro:bit. It's a 5x5 matrix of little LEDs. 5 rows. 5 columns. 5x5=25. Right in the middle of the board.

Check them out:

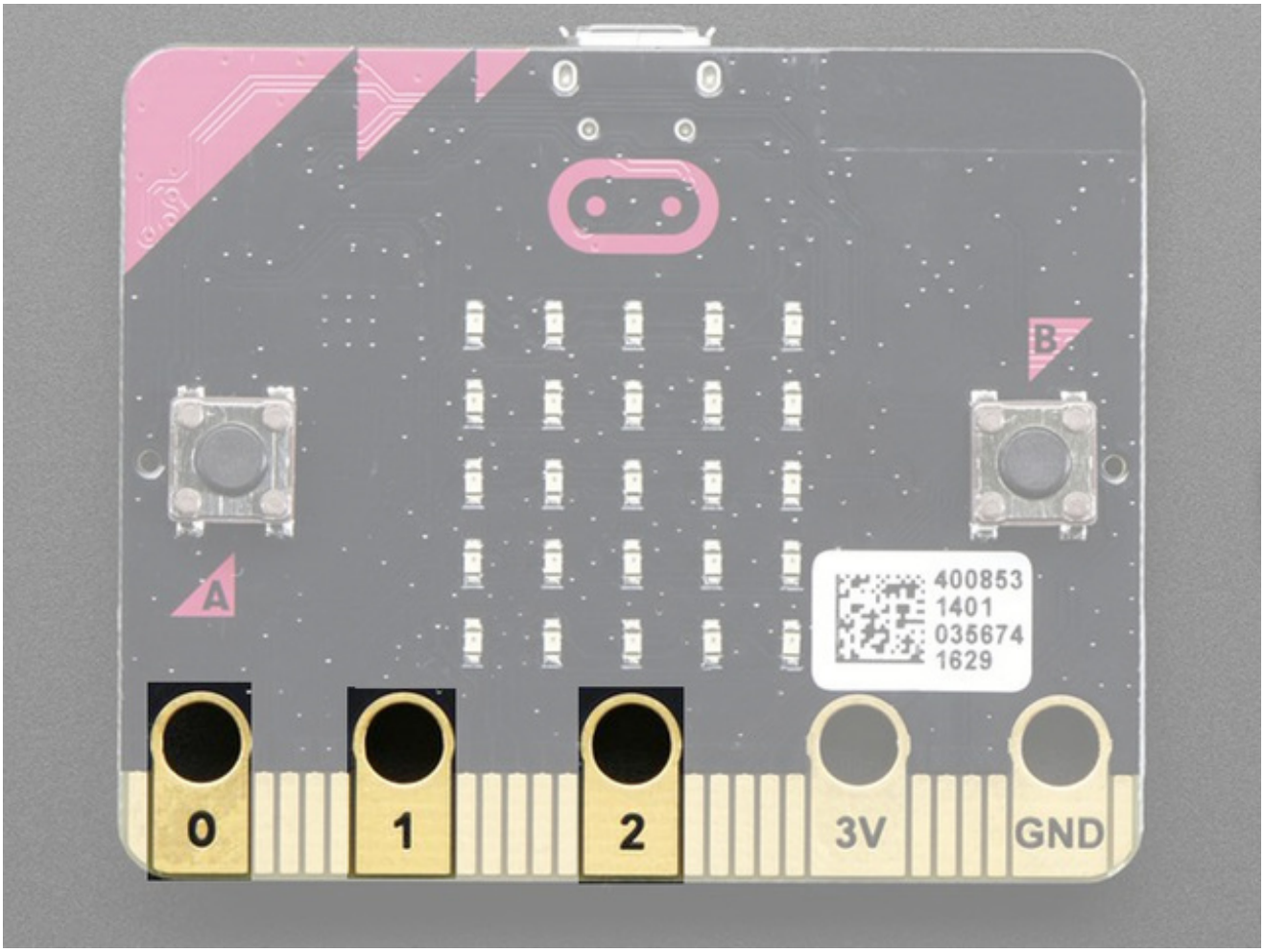


With this 5x5 matrix of LEDs, you can do all sorts of neat things. Each one can be turned on and off individually. So you can draw patterns like hearts and smiley faces. There's even a way to scroll text messages across the display, like "HELLO WORLD!" or "MIND THE GAP!".

Pads

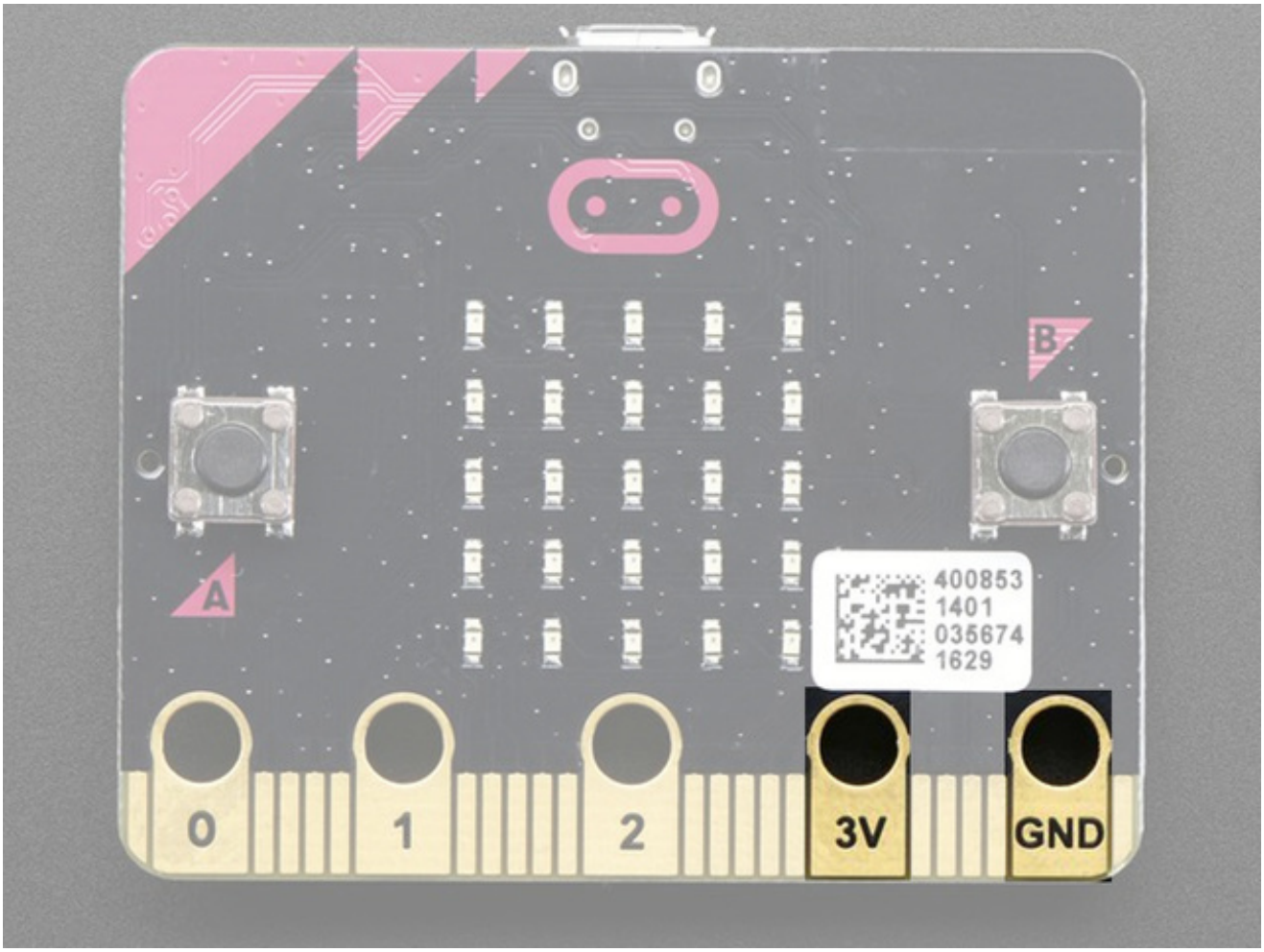
Those 5 pads with the big holes are labeled on this side of the board. You will use these to attach various external devices, typically using alligator clips.

The first 3 are labeled 0, 1, and 2. Each one of these is like the other, they can either be an input or an output. And you get 3 of them to use.



Why not label them 1, 2, 3? Good question. Computery things just like to start counting at 0.

These next two are a little different. They are labeled 3V and GND and are related to power. The 3V stands for "three volts", which refers to the voltage level of the power source. The GND stands for "ground", because everybody needs something to stand on, right? Three volts can't just float in space!


B
A
400853
1401
035674
1629
0
1
2
3V
GND

Let's Code

Alright, so how do you program this thing? You've got two main options:

- **JavaScript Blocks Editor** (http://adafru.it/yES) - drag and drop visual blocks
- **Python Editor** (http://adafru.it/yET) - write code in Python

The really cool thing about both of these is that you do not need to download or install any software. Everything is done using a web browser - which pretty much every computer comes with these days. We'll briefly show how this is done for both options. More details can be found here: http://microbit.org/code/ (http://adafru.it/yEU)

It's Like a USB Thumb Drive!

The other cool feature of the BBC micro:bit is that it behaves and looks just like a USB thumb drive (http://adafru.it/yEV) to your computer. This is the other key feature that means no additional software or drivers need to be installed. Pretty much every computer supports USB storage devices.

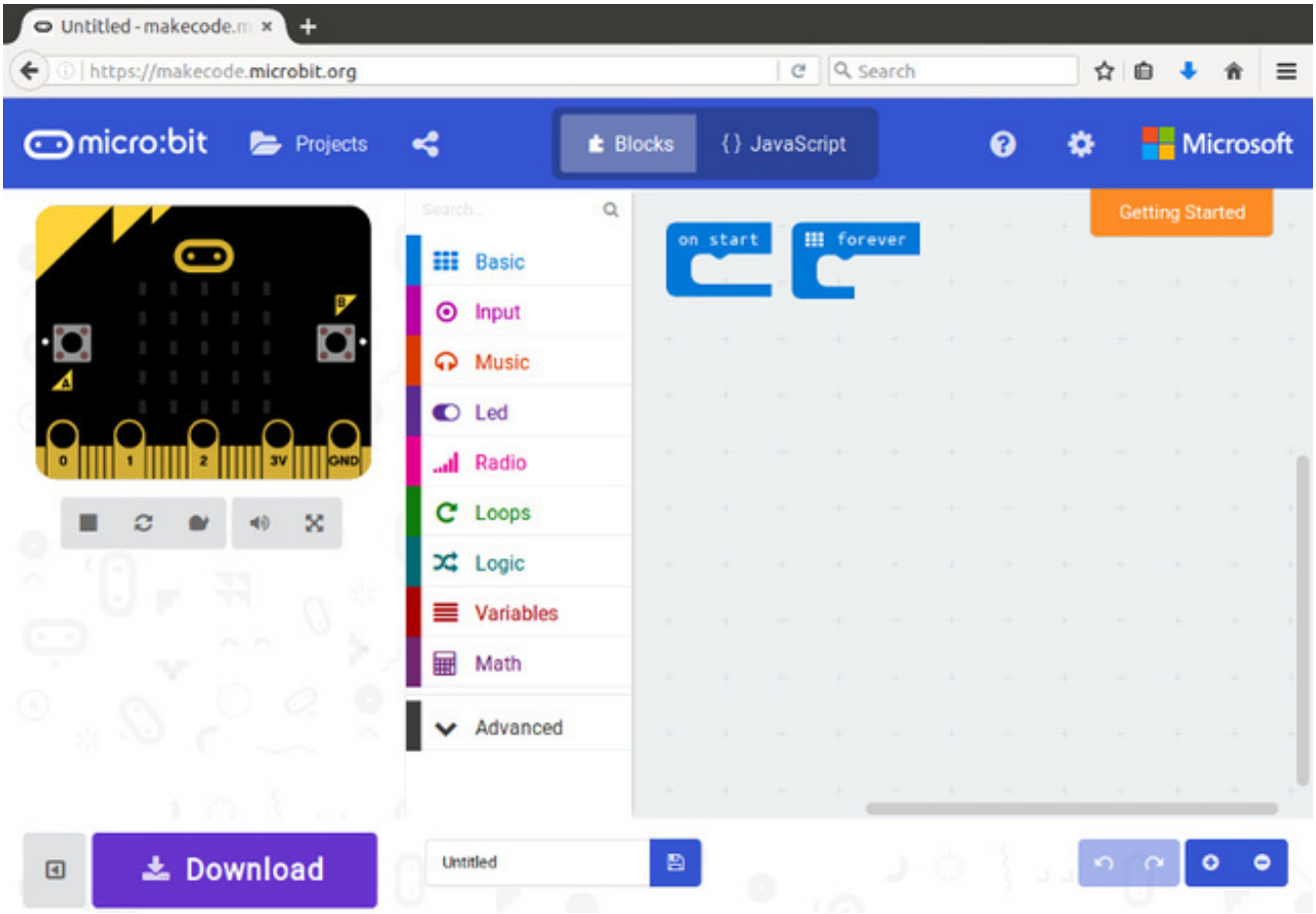
Uploading a program to the BBC micro:bit is like putting a JPG image on a USB thumb drive to share with someone.

JavaScript Blocks Editor

Navigate your web browser to the following location:

https://makecode.microbit.org/ (http://adafru.it/yEW)

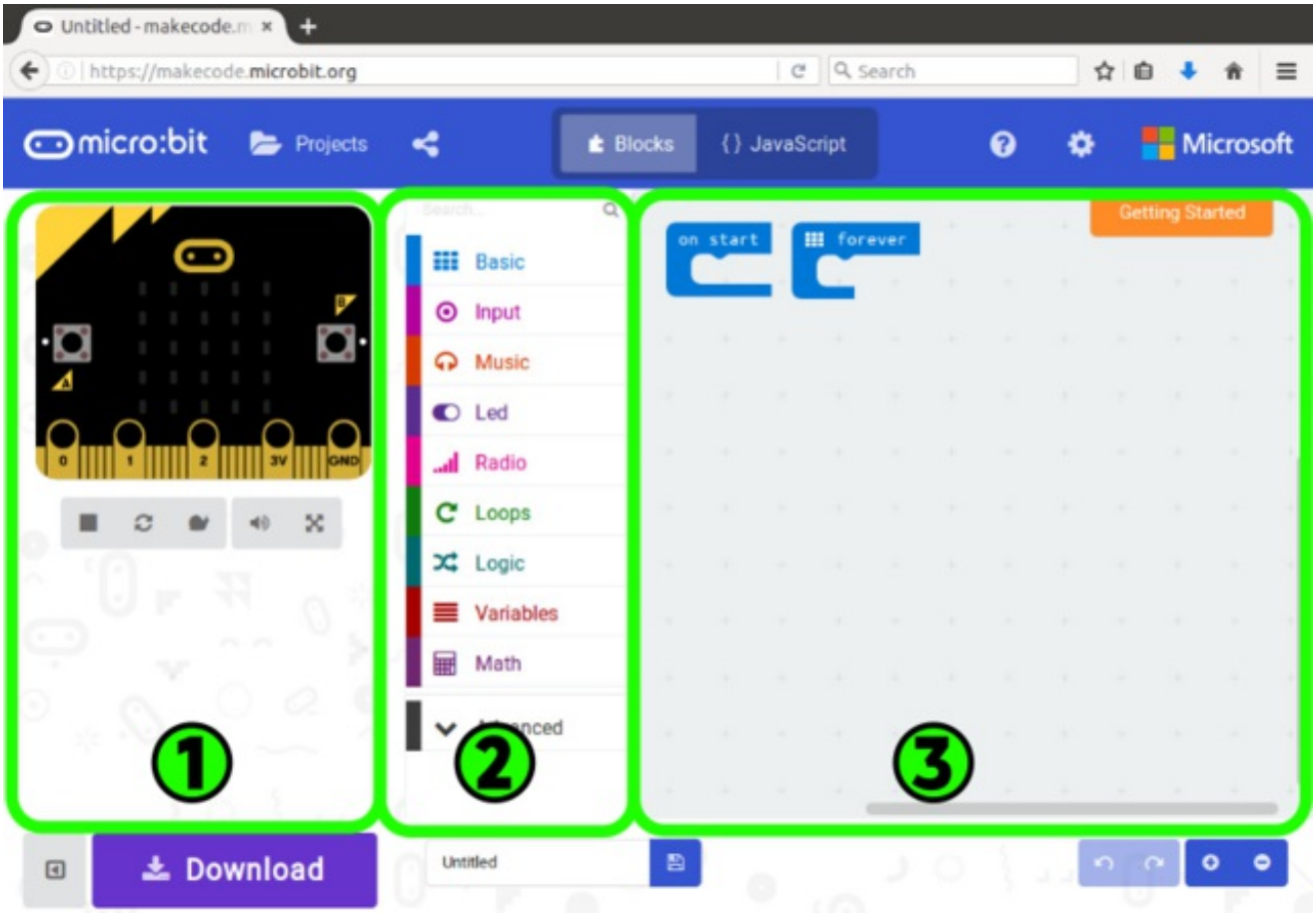
and you should get a window that looks like this:



We'll go over more details of this editor and all its features in a separate guide. Here we'll just run through a quick example to show the basics of creating and uploading a program.

Pressing the orange "Getting Started" button will launch an interactive tutorial.

There are three main areas as shown below:



These three areas are:

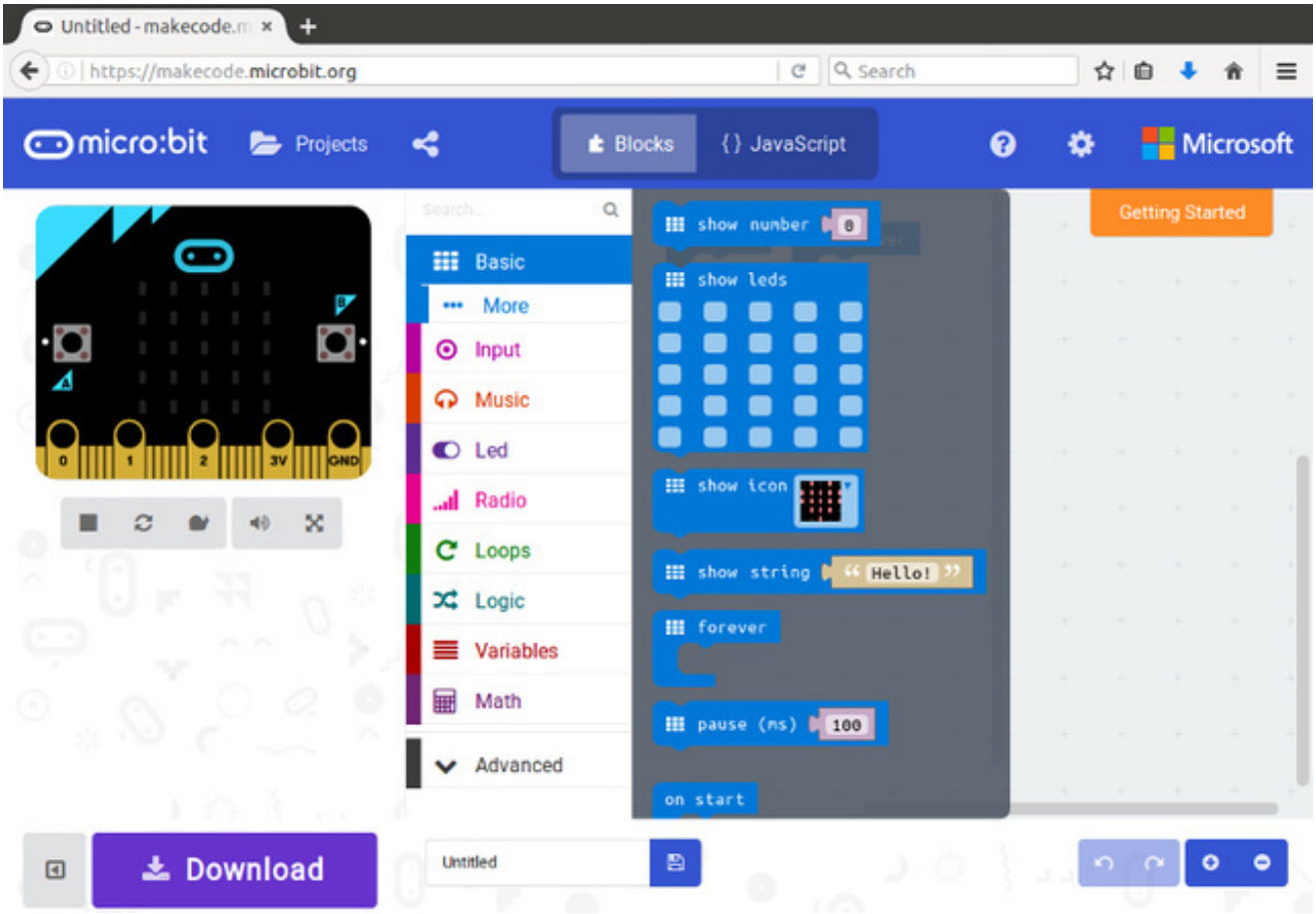
1. **The Simulator**
2. **The Pallet**
3. **The Coding Area**

To write programs, you drag blocks from the **pallet (2)** on to the **coding area (3)** and snap them together. The **simulator (1)** will update on the fly to show the resulting program running on a virtual BBC micro:bit.

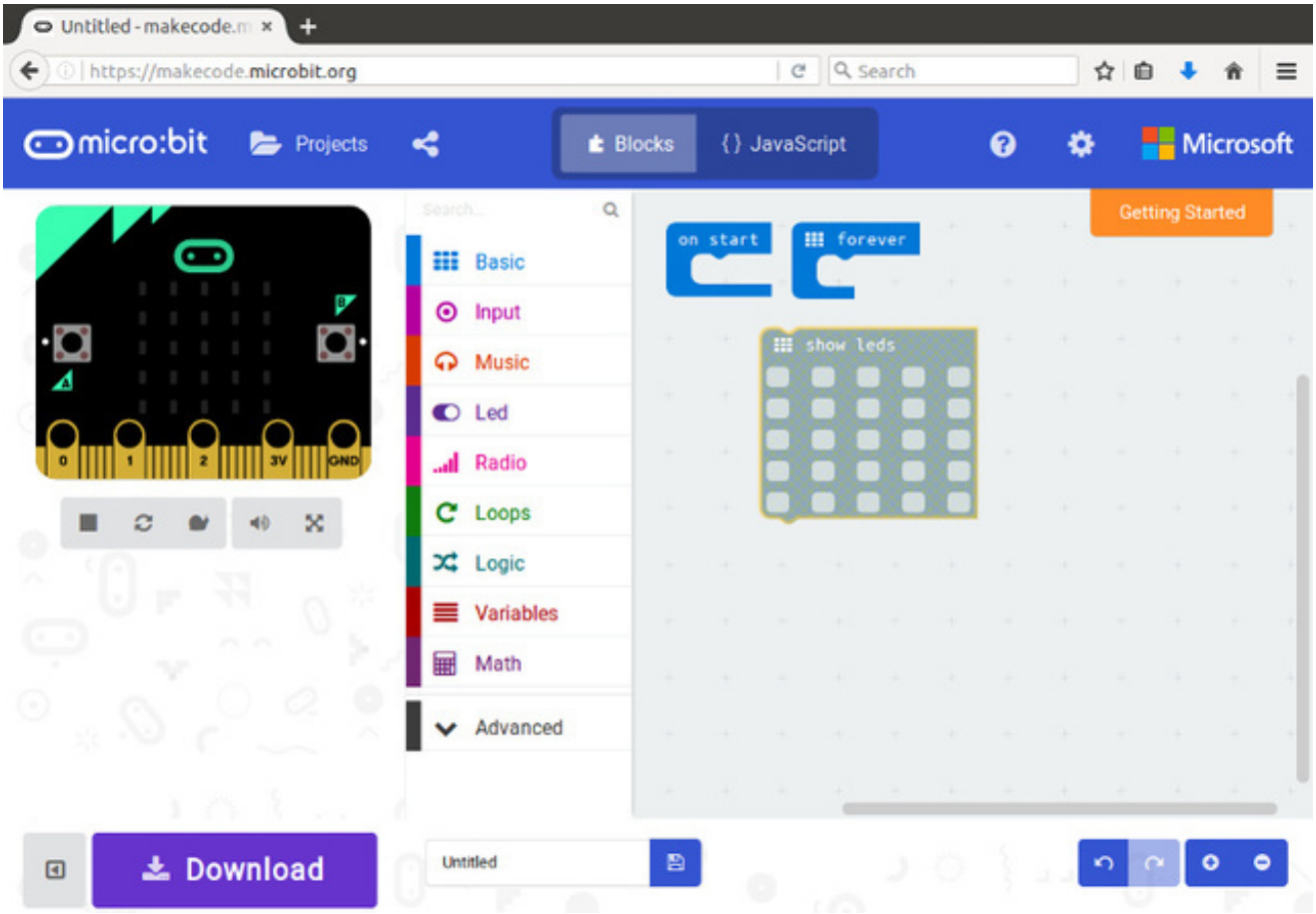
Hello Heart Example

Let's show a very short example.

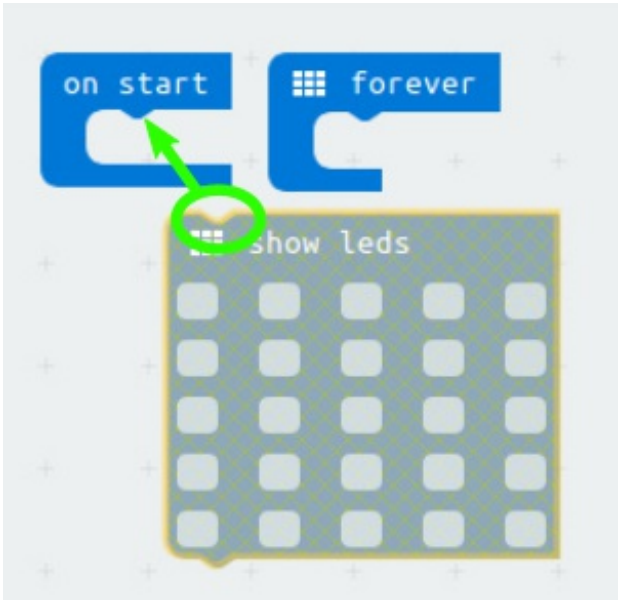
Click on the item in the pallet named **Basic**. This will bring up a variety of block items you can choose from as shown below.



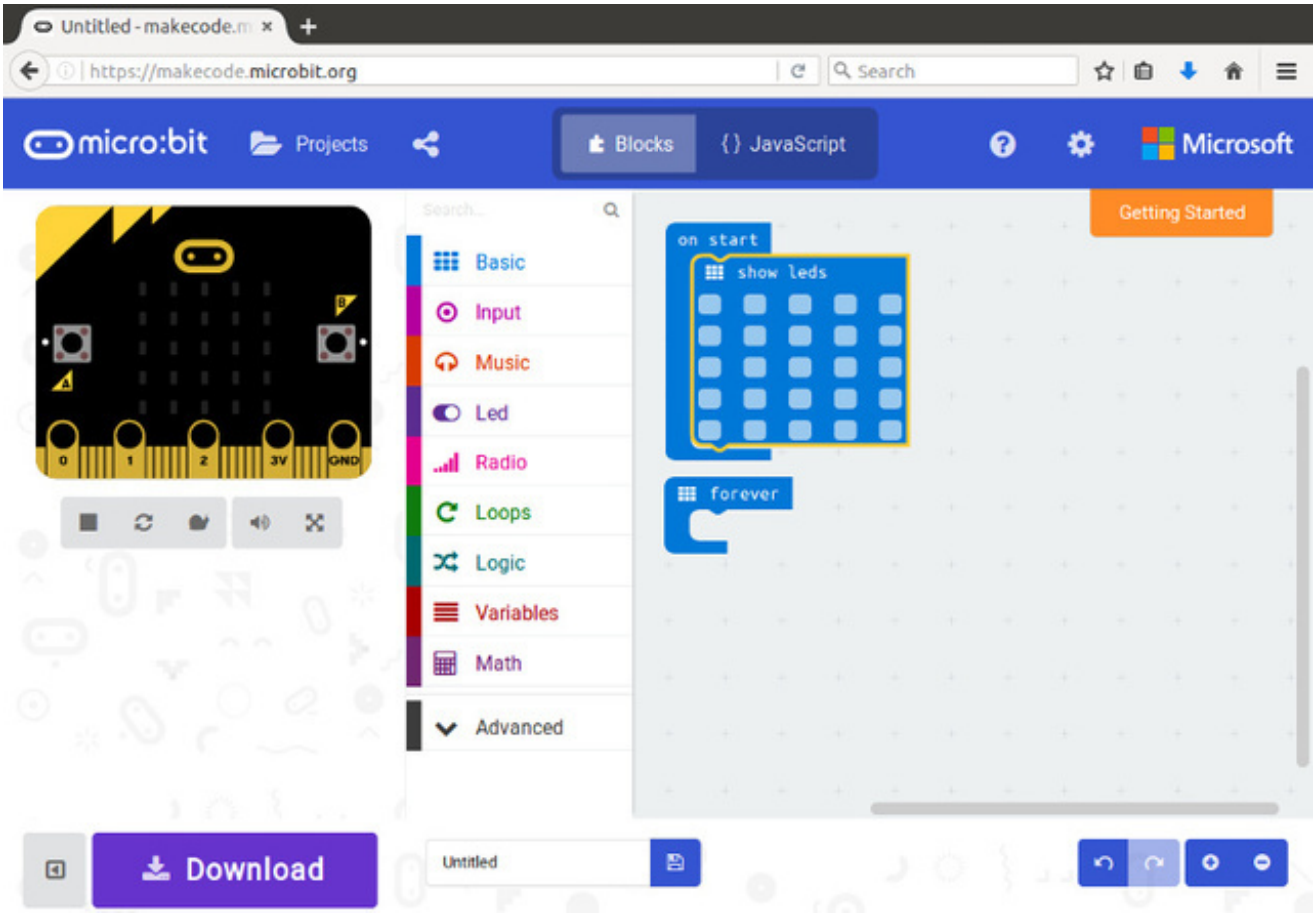
Now click and drag the item named **show leds** on to the coding area.



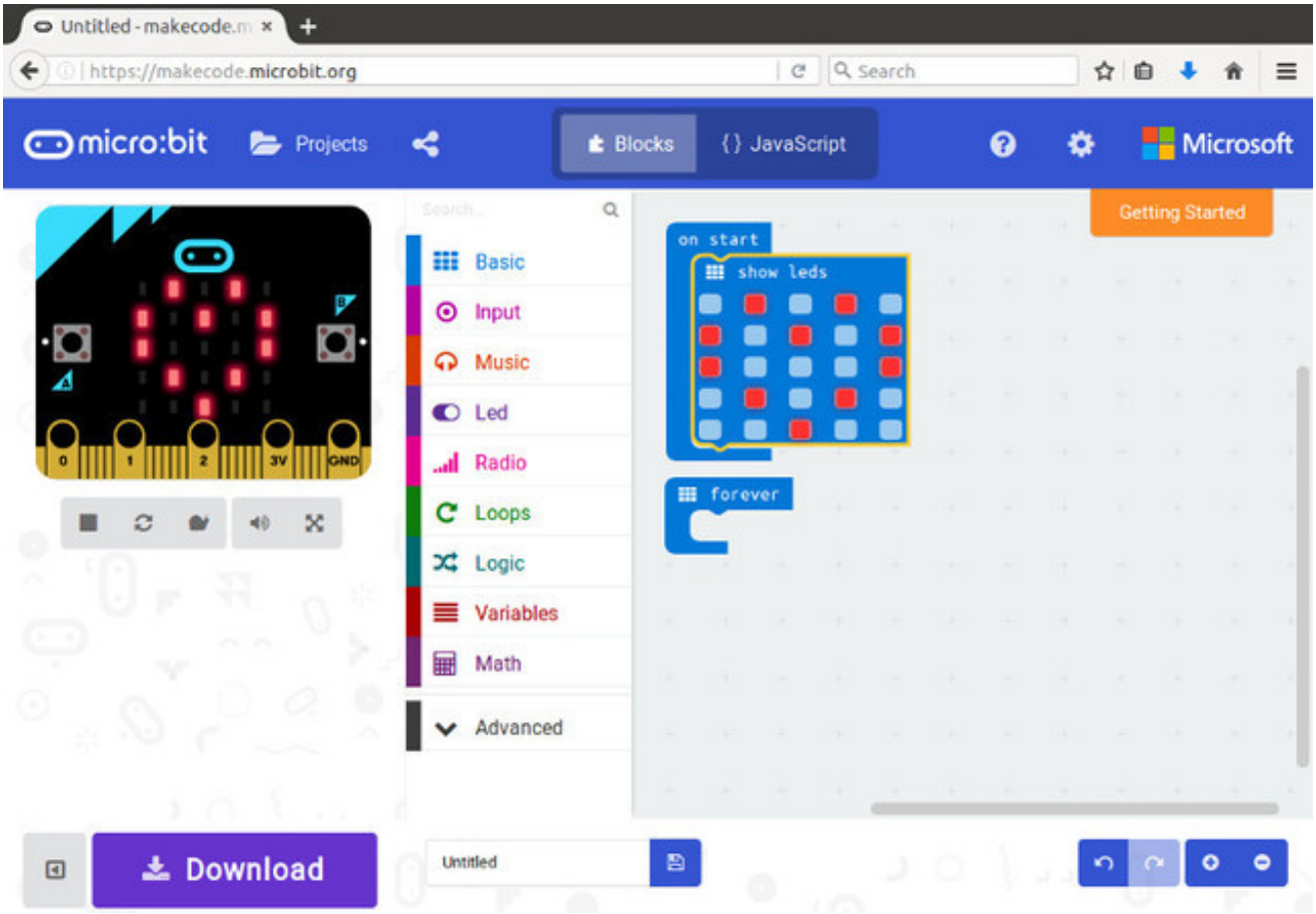
Now drag the **show leds** block so that it "clicks" in to the**on start** block. To do this, you line up the little tab and notch as shown below. Kind of like putting Legos together.



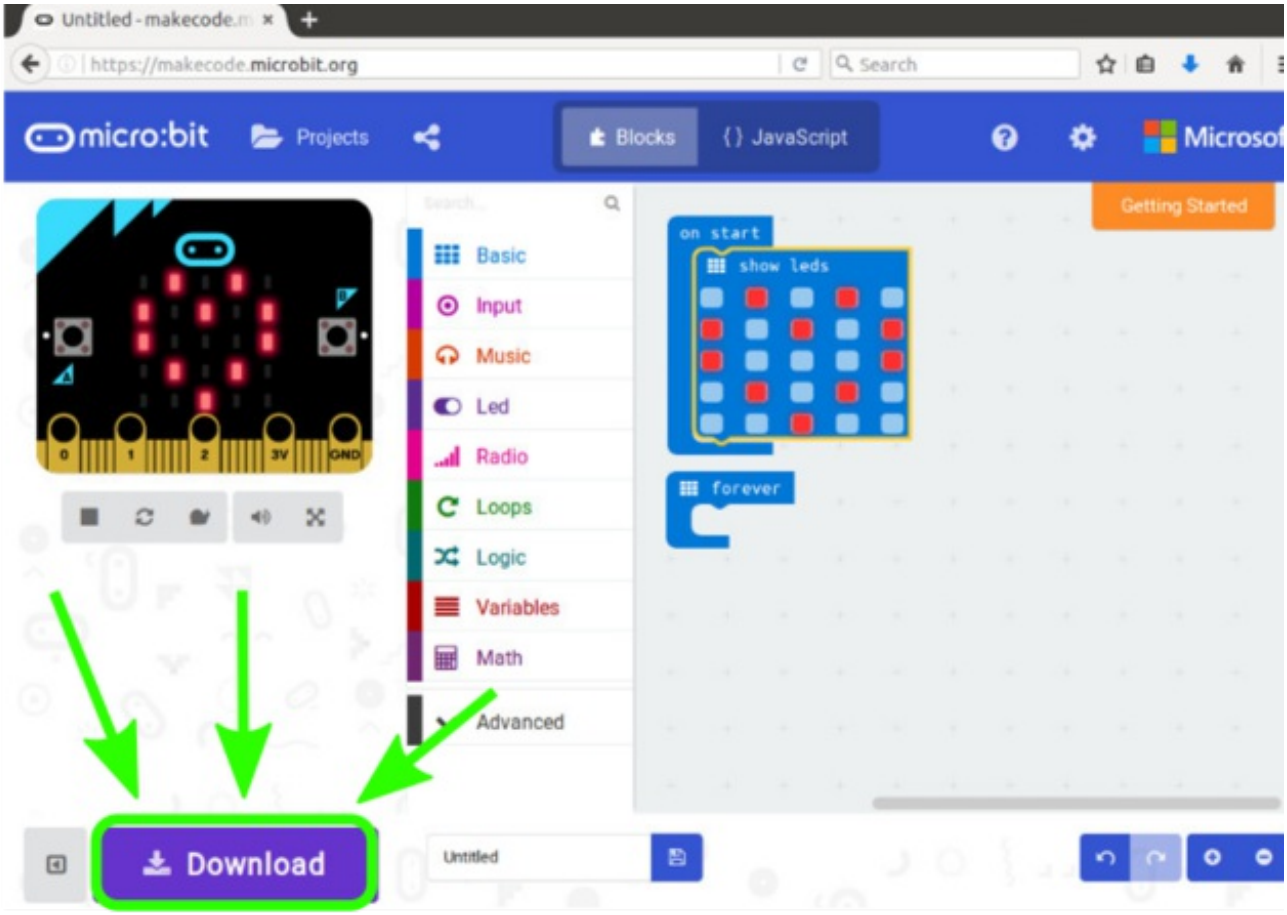
And now it should look like this (I moved the**forever** block down just to get it out of the way):



Now you can click the individual little squares in the**show leds** block to create a pattern. What you are doing is setting which LEDs in the 5x5 matrix are either on or off. They just toggle on and off, so you can just keep clicking until you come up with something you like. I did a heart:



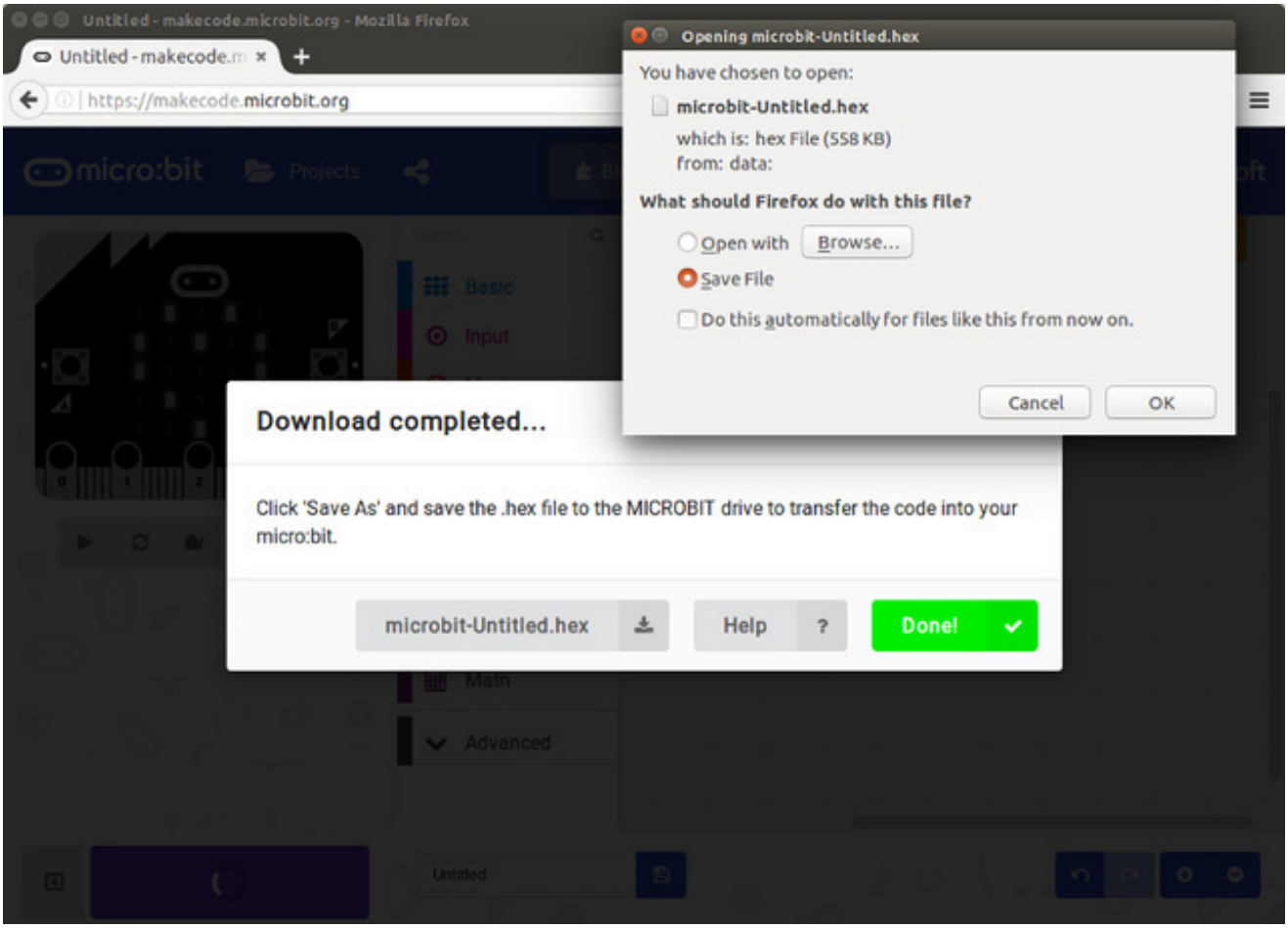
OK, done coding! Now let's download this program to the BBC micro:bit. Press the big button down at the bottom that says **Download**:



What happens next will depend on what operating system you are using and what browser you are using. But what is happening is that a file was created by the Javascript Blocks Editor and it is trying to send it to you. The file is called **mircobit-Untitled.hex** and you just need to save it somewhere on your computer.

It is possible to give the file a name other than 'Untitled', but we don't cover that here.
It is possible to save the file directly to the BBC micro:bit, but we don't cover that here.
These files are often called 'hex files' because they have a .hex filename extension.

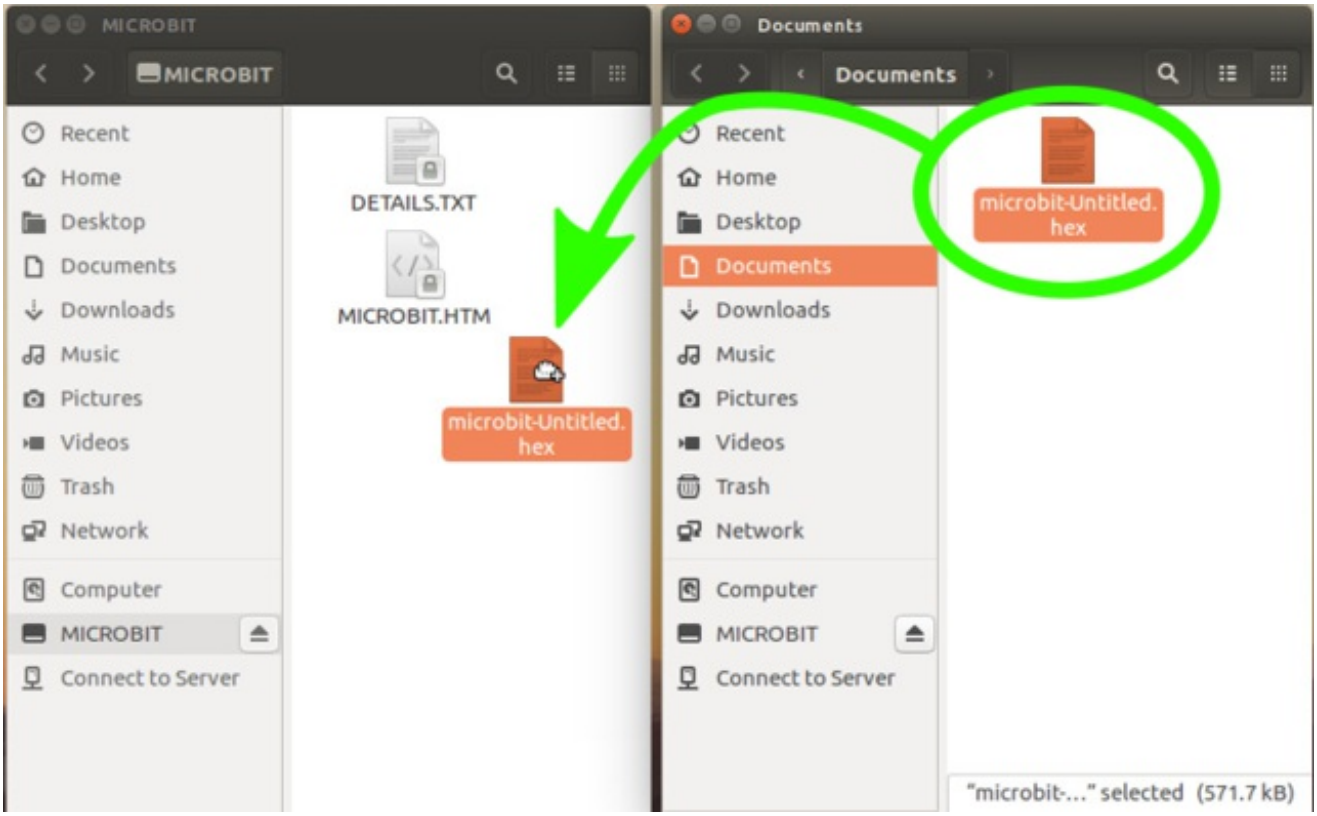
You will likely be prompted to save the file somewhere as shown below.



Go ahead and save it and make note of the location where it was saved. I put mine in my Documents folder.

Now imagine that hex file you saved is a JPG image file and that the BBC micro:bit is a USB thumb drive. The process for uploading the program (the hex file) to the BBC micro:bit is the same process you would use for copying the JPG image file onto a USB thumb drive.

Connect the BBC micro:bit to a USB port on your computer. A folder name MICROBIT should appear. Next, open the folder where you saved the hex file from above. Now simply drag the hex file into the MICROBIT folder.



The little yellow status LED on the BBC micro:bit will blink as the file is uploaded. Once complete, the status LED should stop blinking and you should see your pattern on the 5x5 LED matrix.

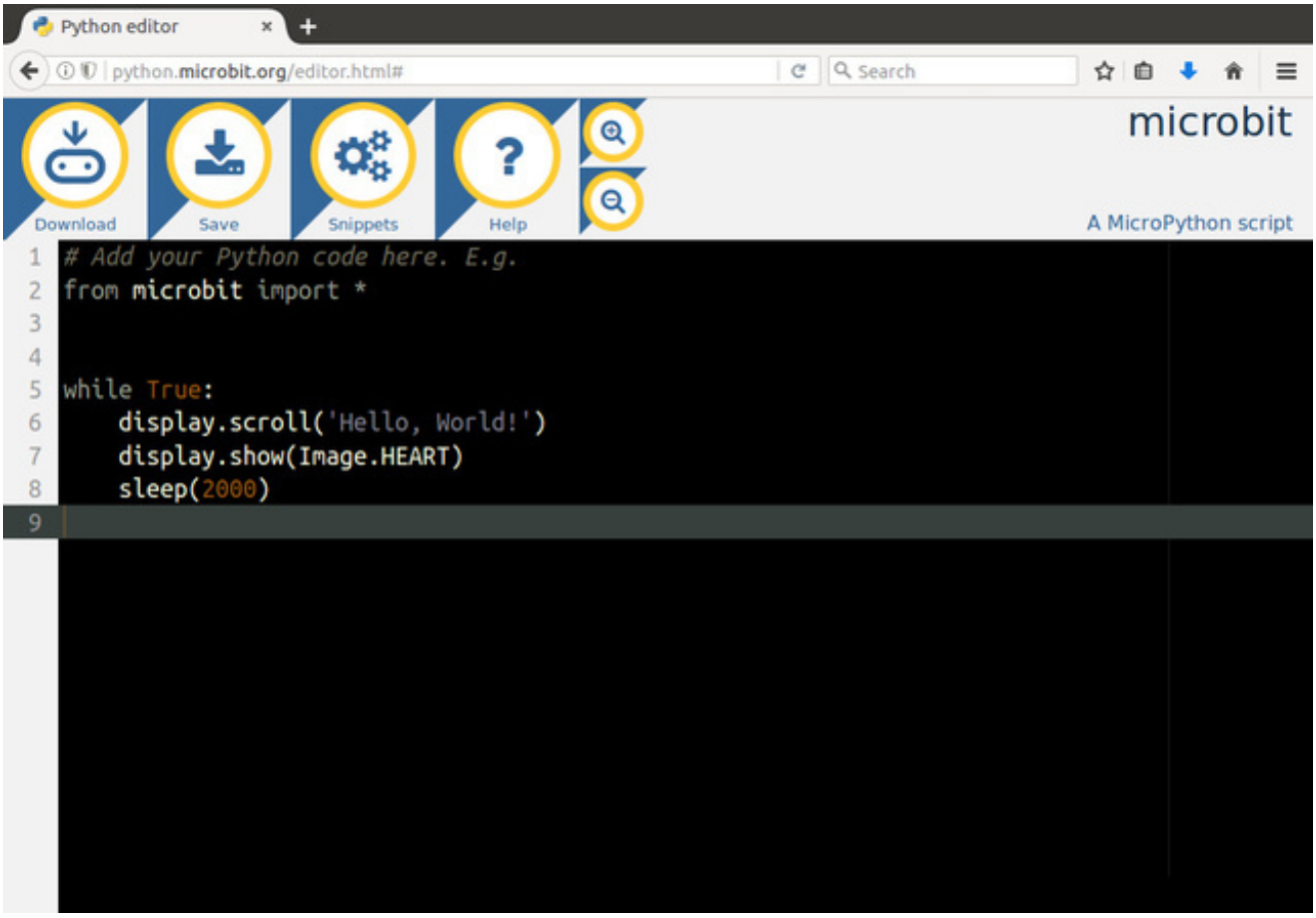
Neat! And that's the basic process for writing and uploading a program using the Javascript Blocks Editor.

Python Editor

Navigate your web browser to the following location:

http://python.microbit.org (http://adafru.it/yET)

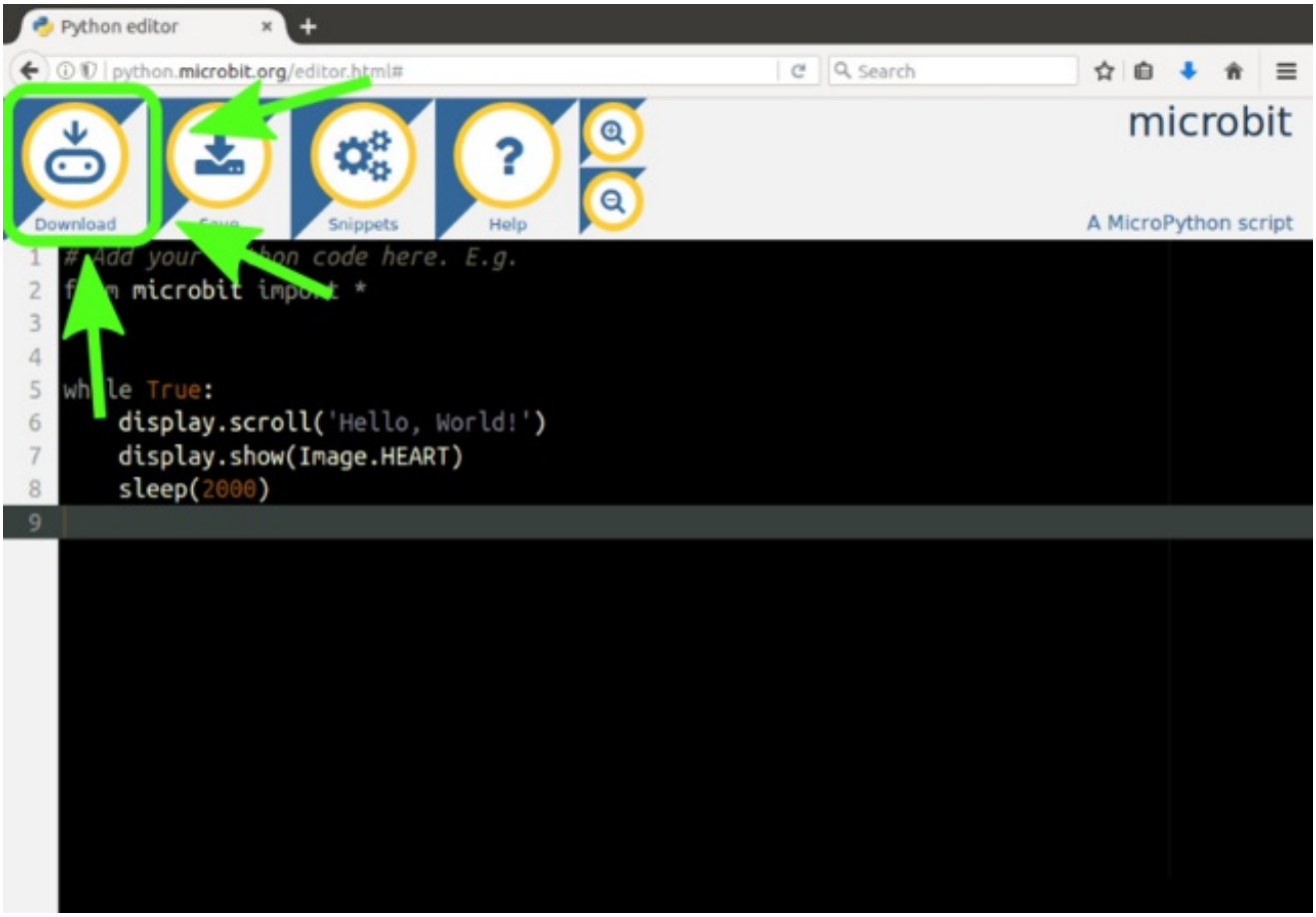
and you should get a window that looks like this:



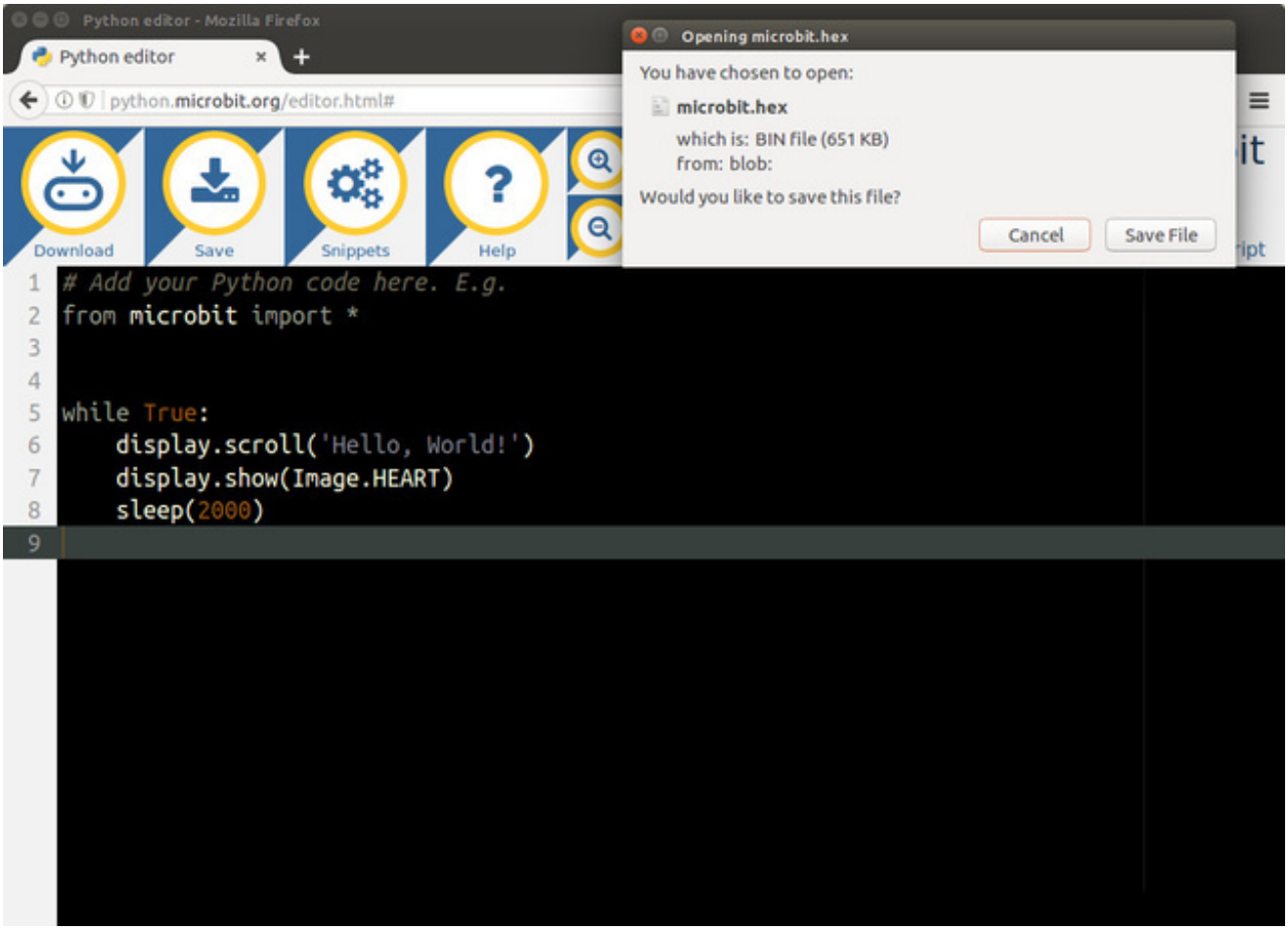
This is a much simpler layout than the JavaScript Blocks Editor. You basically write your Python code in the text editor, and then click Download when you're ready to download the code to the BBC micro:bit.

The web page even loads with a basic demo program already written. We won't go into details of programming in Python in this guide, but what this program does is scroll the message "Hello, World!" across the LEDs and then display a heart. It will repeat this every 2 seconds. Let's just go ahead and use this program as is.

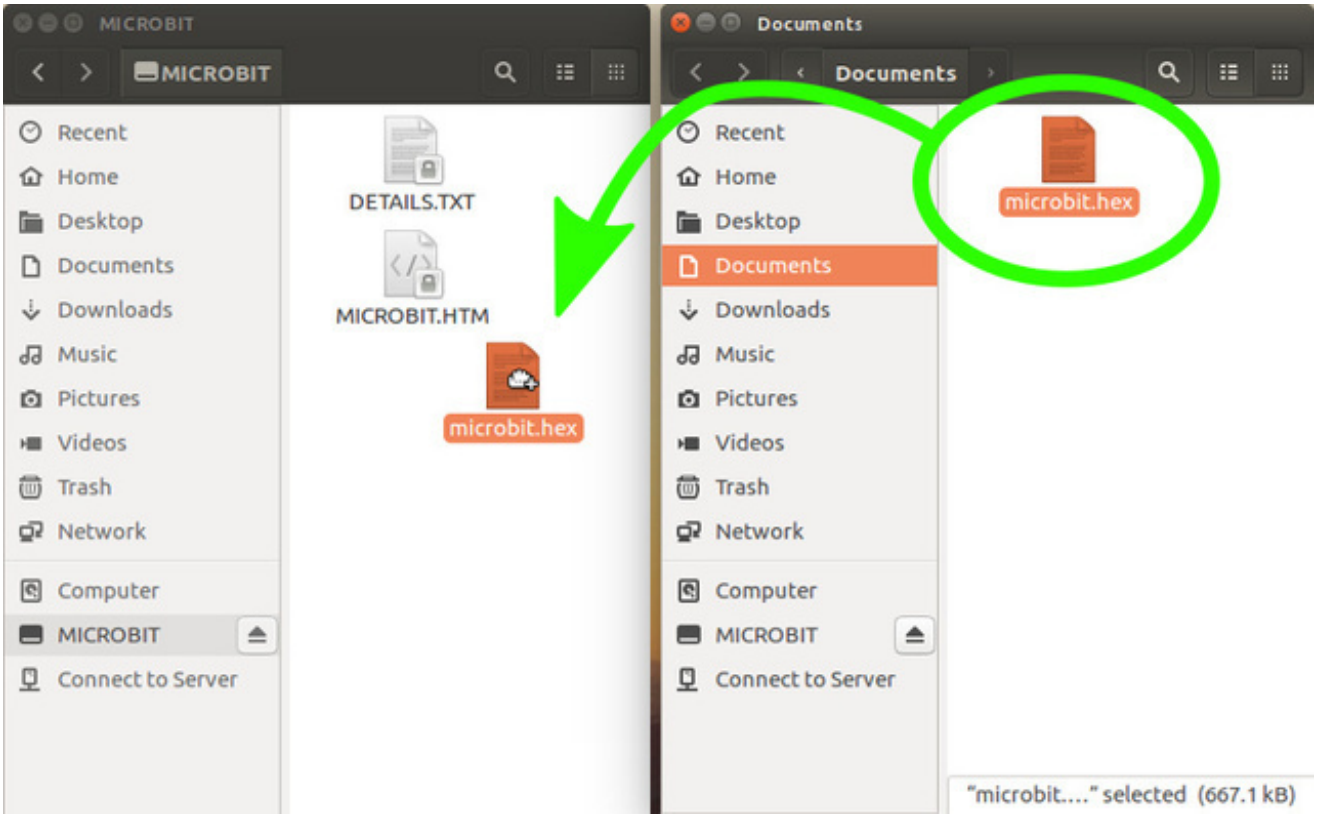
There's no code for us to write, so just click the**Download** button.



You'll be prompted for a location to save the file. This is just like what happened when using the JavaScript Blocks Editor. Just pick a location you want to save it.



This step is also the same as the JavaScript Blocks Editor. Just drag and drop the hex file saved from the previous step on to the BBC micro:bit. Don't forget to plug it in to your computer using the USB cable.



And that's it! You should get the "Hello, World!" message scrolling across the LEDs followed by a cute heart symbol.

What Next?

This guide was meant to simply provide an introductory overview of the BBC micro:bit hardware and the basic process for programming it. There's a lot more to discover and learn playing around with this little board. We hope to cover those in separate guides. But if you were able to get the JavaScript Blocks and Python examples to work, you are ready to move on!

Have fun. The sky is not the limit!