

## MICRO:BITS IN LIBRARIES

You can now loan a micro:bit as you would a book in hundreds of UK libraries.

### Skulls or Snakes?

Build your own micro:bit reaction game.

### micro:bit Radio

Learn how to use the micro:bits onboard radio functions

### Meet the micro:bit Foundation

Who are the people behind the micro:bit foundation?

**EXCLUSIVE  
INTERVIEW!**



**micro:bit  
in Wonderland**

Kerry Kidd

Archie Roques

Joshua Lowe

Les Pounder

Tracy Gardner

Elbrie De Kock

Chris Penn

Gareth Stockdale

Michael Rimicans

Mr Zbit

Tina Fountain

# Hello World!

Hello and welcome to the first issue of micro:mag. This is the unofficial micro:bit magazine for the community run by the community.

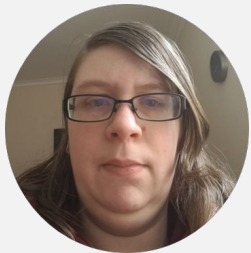
The micro:bit community has been awesome with the support they have given us already offering to write articles for us and working with us from day one to create this amazing issue.

I would also like to thank the Micro:bit Foundation for supporting us and working alongside us from the start. Keep reading to hear more from the team at the Foundation and how they plan on updating us in the future.

If you'd like to write something for a future issue of micro:mag please get in touch at [hello@micromag.cc](mailto:hello@micromag.cc)

**-Kerry Kidd**

## Meet the team



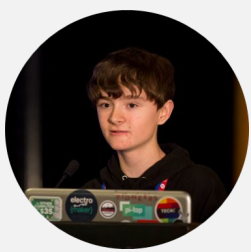
### Kerry Kidd

Kerry is a freelance programmer/educator who enjoys writing tutorials and tinkering with the micro:bit



### Archie Roques

Archie makes lots of different things, from circuit boards to tables. Some of them even work!



### Joshua Lowe

Josh is a young coder, creator of the Edublocks tool for micro:bit, and has done lots of workshops around the world.

## Contact us!

We love it when our readers get in touch with us, whether that's to help contribute to the magazine, show us your cool projects, or just say hello - we are a friendly bunch, so say hi!

### Email:

[hello@micromag.cc](mailto:hello@micromag.cc)

### Twitter:

[@micro\\_mag](https://twitter.com/micro_mag)

### Web:

[micromag.cc](http://micromag.cc)



# micro:mag needs you!

All of our content is written and provided by community members. We're really keen to hear from anyone who would like to contribute to the magazine, whether you're a seasoned writer or just want to have a go.

## Get in touch!

[micromag.cc/contribute](https://micromag.cc/contribute)

[hello@micromag.cc](mailto:hello@micromag.cc)

[@micro mag](#)



# :contents

## :news

- 6** **Liverpool Makefest**  
Annual maker event returns
- 7** **Young Coders Meetup**  
A new initiative sets up campaign
- 8** **micro:bits in Libraries**  
Now in hundreds of UK libraries

## :feature

- 11** **micro:bit in Wonderland**  
Exclusive interview with the authors
- 17** **micro:GUI**  
Build a GUI for your micro:bit

## :make

- 20** **micro:bit + Minecraft Pi**  
Control Minecraft with your micro:bit
- 25** **Sending radio messages**  
Send messages to another micro:bit
- 29** **Hide and seek a micro:bit**  
Make a fun hide and seek game

# Highlights

## :feature

Hear from the authors of *micro:bit in Wonderland* in this micro:mag exclusive interview!

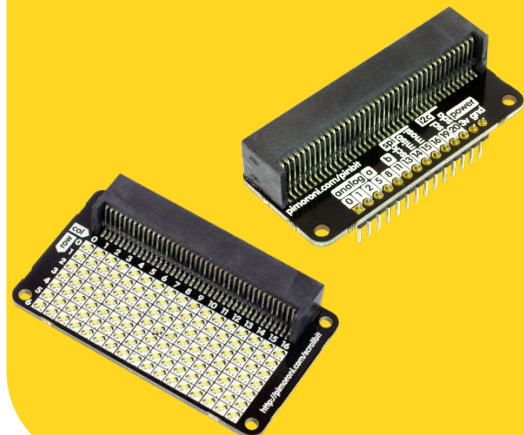


Page

**11**

## :review

We review Pimoroni's brand new micro:bit add-ons, pin:bit and scroll:bit



Page

**39**

**41**



# contents:

## Highlights

### news

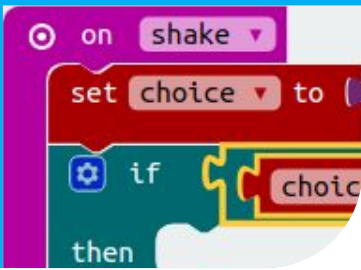
You can now loan micro:bits from hundreds of libraries across the UK. We hear from the project leader on how the scheme is going...

Page  
8

### micro:hit

In the first of our regular series, Les Pounder helps us take decisions with our micro:bit

Page  
33



### Other Articles

- micro:hit** 33  
9 Simple steps to create a fun game.
- Message of support** 35  
Gareth Stockdale on micro:mag.
- :foundation** 36  
Get to know who's behind micro:bit
- in:schools** 37  
micro:bits in Venezuela

### :review

- Pimoroni pin:bit** 39  
A simple micro:bit breakout board
- Pimoroni scroll:bit** 41  
A bigger LED display for the micro:bit

If you'd like to contribute a project, tutorial or article, send us an email at [hello@micromag.cc](mailto:hello@micromag.cc). We look forward to your contribution!

# Liverpool Makefest

## Set to return to Liverpool Central Library.

The annual Liverpool Makefest is returning for its 5th year where it will see hundreds of makers celebrating making and showing off different projects to the public in Liverpool.

### Josh says:

“I have been to this event every year. At Liverpool Makefest there is something for all the family to enjoy and it’s an event I always look forward to.”

Joshua Lowe

Now in its fifth year, Liverpool Makefest is once again returning to Liverpool Central Library where around a thousand people will gather to celebrate making and show off each others’ projects.

The event is packed with fun for all ages, whether you’re running a stall, a workshop or even just popping in for a look around, there is something for you to enjoy.

Entry to the event is free, and there are five floors in Liverpool’s iconic Central Library packed full of makers each with something unique to show.

This year’s exhibitor list ranges from glass-blowing to Code Club to lacemaking, along with plenty of micro:bit action from the Foundation and other community members.

Liverpool Makefest is on Saturday 31st June 2018 from 9 am to 5 pm at Liverpool Central Library, William Brown Street, Liverpool. Tickets can be acquired from Eventbrite: [go.micromag.cc/lmf88ca6](https://go.micromag.cc/lmf88ca6)

The Makefest team are also looking for volunteers - you can sign up to help at [go.micromag.cc/lmv685d](https://go.micromag.cc/lmv685d)



EduBlocks’ stand at MakeFest in 2017.  
Picture: Josh Lowe

# **Femi's back again!**

## **Young Coders Meetup campaign launched.**

**After a successful Young Coders Conference back in February, Femi Owolade-Coombes launches his new campaign to roll out a number of Young Coders Meetups across the UK.**

If your an active member of the micro:bit community, chances are you will have heard of 12 year old Femi Owolade-Coombes. Femi is an inspirational young person who goes out into the community across the world to spread the word about inclusive environments for coders and more recently, raising awareness of the Young Coders Community.

In February, Femi and the South London Raspberry Jam team ran their first Young Coders Conference, where 12 lucky young people were selected to be trained on how to run micro:bit workshops and deliver them to the public.

"We want to set up monthly MeetUps for young coders providing an inclusive & youth-led community for 10-16 yr olds to learn, collaborate & network together-featuring talks & workshops of their choice."

Building upon the success of the conference, Femi along with the South London Raspberry Jam team have launched their latest campaign to roll out Young Coders meetups across the UK. These meetups will be run by young people, for young people giving young coders the chance to share their knowledge with others, this is something that is important in order for young people to gain confidence in the coding community.

**Campaign Link:**

[go.micromag.cc/youngd2582](https://go.micromag.cc/youngd2582)





## Michael Rimicans

Michael has been tinkering with the micro:bit since it's release and using it for cool things. He's a STEM ambassador, Code Club volunteer



@heedt  
[heed.net](http://heed.net)

Pictures from Les Pounder.

# micro:bits in Libraries

**Michael is giving us some insight into the micro:bit in libraries scheme.**

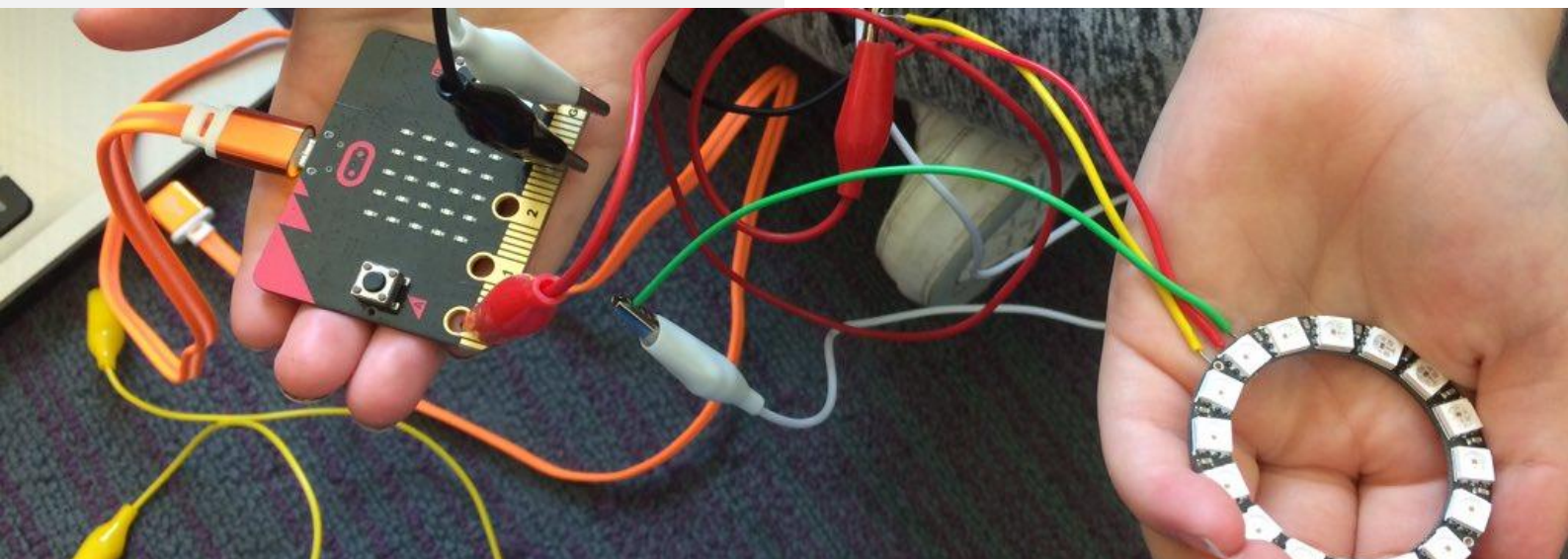
"The Raspberry Pi Foundation claimed the 'First to give away a computer free on a magazine' with the Pi Zero, micro:bits in libraries appear to have become the first to loan out: free of charge, a computer board in the same way a library book can be borrowed, allowing access to technology to everyone."

One of the aims of the micro:bit was to enable young people to have access to a simple computer board that enabled them to learn the basics of computing and physical computing. Initially, the board was to be given free of charge to all Year 7 children in 2016 with a follow up general public release shortly after.

By this time the Huddersfield Raspberry Jam had been in operation for two years and, as usually happens eventually, there was a staff change which resulted in me getting an invite to a meeting with my new library contact.

I walked into the meeting room to meet for the first time Amy Hearn and Kirstie Wilson, two of the development librarians for Kirklees, who were curious to find out what it was all about and could anything be used to support a Family Learning Festival the library had planned.

After explaining about the Jam in general and, as it was now generally available for purchase, I mentioned the micro:bit. Kirstie and Amy admitted that they had





heard of it and were curious. The micro:bit website and the block editor was quickly opened on the laptop and after a few minutes playing around a discussion began as to how they could be used with the existing library computers in the festival.

It was then I mentioned that as the micro:bit was relatively low cost then the library could "...could also buy some to loan out like books". It seemed that a light bulb went ping above the table with both Amy and Kirstie intrigued by the opportunities this could present.

A few weeks later I received an email from Amy informing me that forty micro:bit had been purchased and were being added to the library catalogue for general lending and could some sessions be carried out as part of the learning festival. These sessions were two hours long and held at four local libraries which each received a set of ten of micro:bits to lend out. The sessions were well received and there was a quick uptake of micro:bit being taken home.

With all this activity being tweeted it was eventually noticed by the micro:bit educational foundation and by Philip

Meitiner, who at the time was the International Program Manager and instrumental in getting the foundation further involved. The idea then snowballed, using Kirklees Libraries as a template, to involve other library authorities around England with Amy being the point of contact for the requests.

After a series of meetings a set of trial library authorities, Blackpool, Coventry, Newcastle and Southend each received a set of micro:bit and, where possible, put in touch with local support. In Blackpool, this was Les Pounder and in the South-East, Andy from Southend Tech provided support whilst also being part of the development process. It was up to the library how they were to be used.

Feedback from these trial deployments were positive and a further batch of micro:bit sets were made available from the foundation. This allowed for a further set of libraries to be supplied with micro:bit. By this time the scheme had started to be talked about on social media with the hashtag **#microbitsinlibraries** being quickly settled upon and became the name for the scheme. It was then made possible for library authorities to register





their interest in the scheme and eventually receive a pack of micro:bit for use in their libraries.

Now, a year later, the scheme has supplied micro:bit packs to nearly a thousand libraries in the England with plans for expansion into the rest of the UK and similar schemes starting to appear in eastern Europe. Whilst the Raspberry Pi Foundation claimed the 'First to give away a computer free on a magazine' with the Pi Zero, micro:bits In

Libraries appear to have become the first to loan out, free of charge, a computer board in the same way a library book can be borrowed allowing access to technology to everyone.

### **Want to Know More?**

You can find out more about the micro:bit in Libraries at [go.micromag.cc/lib477f7](http://go.micromag.cc/lib477f7). You can also find out how to get your library involved too.

## **Do you make micro:bit accessories?**

The micro:mag team are looking for products to review for future issues. If you would like us to consider reviewing your product, get in touch at

**[hello@micromag.cc](mailto:hello@micromag.cc)**





**feature:**



# **micro:bit in Wonderland**

**We interviewed Elbrie de Kock  
and Dr Tracy Gardner about  
their micro:bit activity book**



# :feature

**We got the chance to interview the authors of *micro:bit in Wonderland*, a *micro:bit* activity book for kids. Here's what they had to say....**

## **What inspired you to write *micro:bit in Wonderland*?**

**Tracy:** I was teaching Computing to upper key stage 2 and the school couldn't afford new resources. I managed to borrow some *micro:bits* and honestly I thought the black with red LEDs combination looked a bit harsh and wanted to make them engaging. I couldn't find any materials at the time that were inspiring so I decided to create some in my own time and I got a bit carried away!

**Elbrie:** Tracy told me about her lessons and showed me the projects she created and I was immediately hooked. The maker elements and licence to be creative got my attention and I knew we had to share the projects with more children.

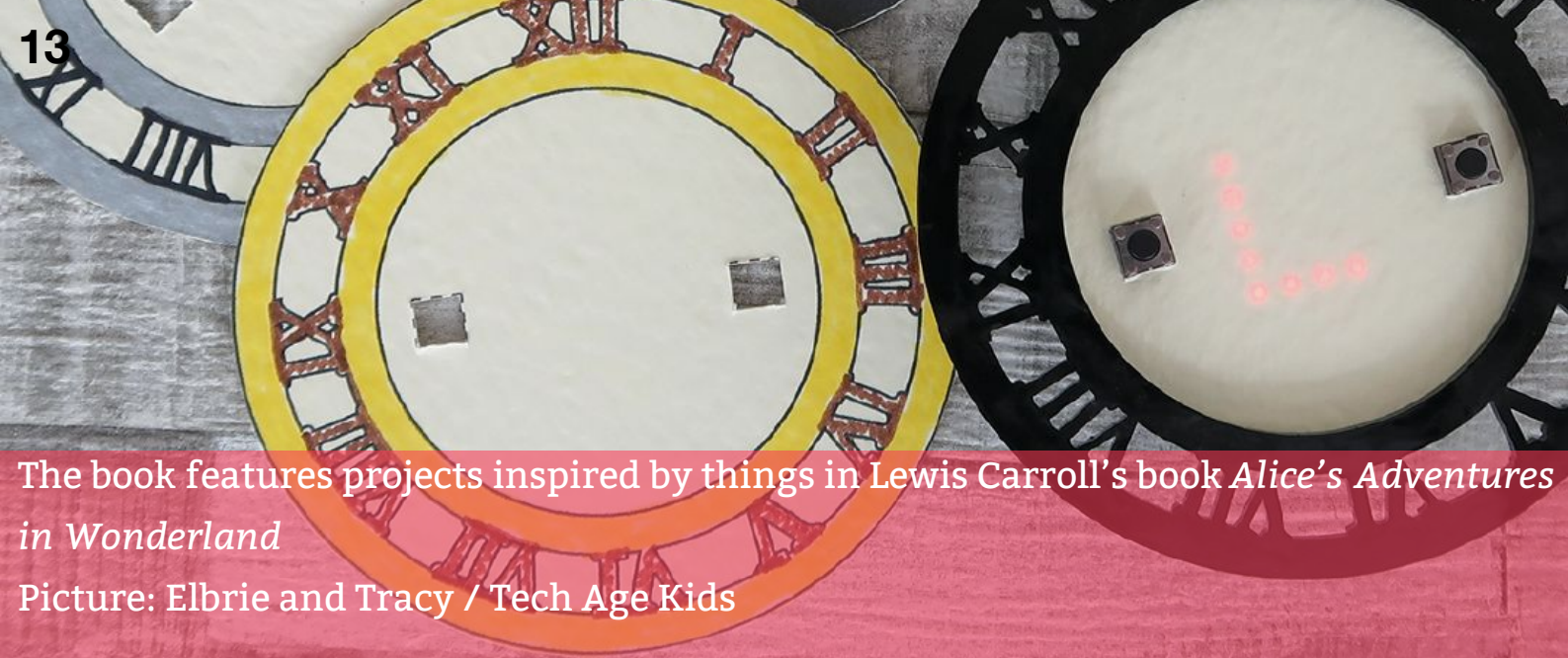
## About the makers:

### Dr Tracy Gardner

Tracy has a Computer Science PhD, and has worked as a software engineer and software architect, including working for 10 years at IBM. Tracy used to teach Computing to Key Stage 2 children (aged 7-11), and now focuses on introducing technology to the next generation as a director of Tech Age Kids. She also develops educational content for the Raspberry Pi Foundation, has 2 children, and volunteers at a Code Club and Coder Dojo.

### Elbrie de Kock

Elbrie de Kock has an Interior Design degree and has worked in a number of different industries as a designer and digital marketer. Elbrie has three children, and her eldest son's passion for computer programming inspired her to find opportunities for kids to learn to code. She uses her creative background and newly developed technology skills to create projects that combine craft, coding and electronics. Elbrie is a director of Tech Age Kids and organises creative technology events for children and families. She volunteers at a Code Club and founded the local CoderDojo.



The book features projects inspired by things in Lewis Carroll's book *Alice's Adventures in Wonderland*

Picture: Elbrie and Tracy / Tech Age Kids

**What kind of research did you do, and how long did you spend researching before beginning the book?**

**T:** Before beginning the projects, none really! I just came up with the idea and created the first project. But, I was already familiar with Lewis Carroll and Alice and had experience working with the micro:bit at Code Club / The Raspberry Pi Foundation where I also work. I was also familiar with the national curriculum for Computing.

**T&E:** Before deciding to write more projects and turn them into a book, we did lots of research. In particular, we read all about Lewis Carroll and the Victorian history of technology. The Victorians were real innovators and their discoveries inspired many aspects of the book.

**What was the biggest challenge you came across when writing the book?**

**T:** The biggest challenge for me was to make the book increase in complexity while also following the story in order and keeping the costs of the materials used low. Lots of constraints that all needed to be balanced.

**E:** We really wanted the book to be accessible to non-technical adults and children. I don't have a background in technology so it was a challenge for me to come to grips with the concepts and then find ways to present it through words and graphics for others.

**How long (roughly) did it take you to write the book?**

**T&E:** That's a tricky question because it was done in between other work. It was roughly 9 months from the first idea to the published book.



**Is there any particular author or book that influenced you in any way to writing micro:bit in Wonderland?**

**T&E:** We'd have to say, Charles Dodgson, the mathematician who used Lewis Carroll as a pen name for his children's books. *Alice's Adventures in Wonderland* is packed with creativity and also linked to logic, one of his professional interests. Dodgson also loved technology, he was an early adopter of photography.



**What was your favourite part of the book to write and why?**

**T:** For me, it was the 'Bit of History' sections. I think it's really important that we understand where today's technology came from. It didn't just appear from nowhere! The rate of invention in the Victorian age was amazing.

**E:** I made all the projects for the book and did the photography and graphics. The task really played to my strengths and I loved getting creative. Tracy had to hold me back at times. We could've done so much more, but wanted to leave room for others to get creative. The 'Make it yours' sections of the book is my favourite, where we give the readers and makers licence to be creative.

*Charles Dodgson  
[alias "Lewis Carroll"]*

Lewis Carroll, the author of *Alice's Adventures in Wonderland*, was actually called Charles Dodgson, and was a mathematician and logic expert at Oxford University. He told the *Alice in Wonderland* stories to a young friend of his called Alice, and later wrote them down for others to read.





Learn about programmable electronics  
Picture: Elbrie and Tracy / Tech Age Kids

## How did you get involved in Computing?

**T:** When I was 10 we got a Sinclair Spectrum computer and I learned to program in BASIC. I studied Maths & Computing at university followed by a Computer Science PhD. I spent my career in industry, most of it at IBM, before getting involved in children’s education once I had children myself.

**E:** My dad worked with computers but as a child, I never took much of an interest. In fact, I remember one day I was hand-crafting a school project, and my dad said I could use a computer to do some of the work. I told him, I will never use them! Years later, my son (now an adult) took a great interest in computer programming. In an effort to support his learning, I got involved in the technology industry and have never

looked back. It’s the most creative industry I have worked in and I love learning new skills all the time.

## How did you come up with the title?

**T&E:** The title micro:bit in Wonderland was quite easy, but the subtitle ‘Coding and Craft with the BBC micro:bit’ was harder, we wanted to emphasise that the book is about making stuff that includes technology but isn’t just about technology.

### 10/6 THE MAD HATTER'S HAT

BONUS CHAPTER 1

**FOLLOW THE STORY**


Read Chapter 7 *A Mad Tea-Party* and meet the Mad Hatter with his top hat.

At the Mad Tea Party the Hatter is wearing a top hat with a label on it. The label reads *In this style, 2016*. It's a price label.

The Hatter makes hats and this one is from his shop. Lewis Carroll wrote *Alice's Adventures in Wonderland* in 1865 when hats were handmade by Hatters.

**YOU WILL MAKE**

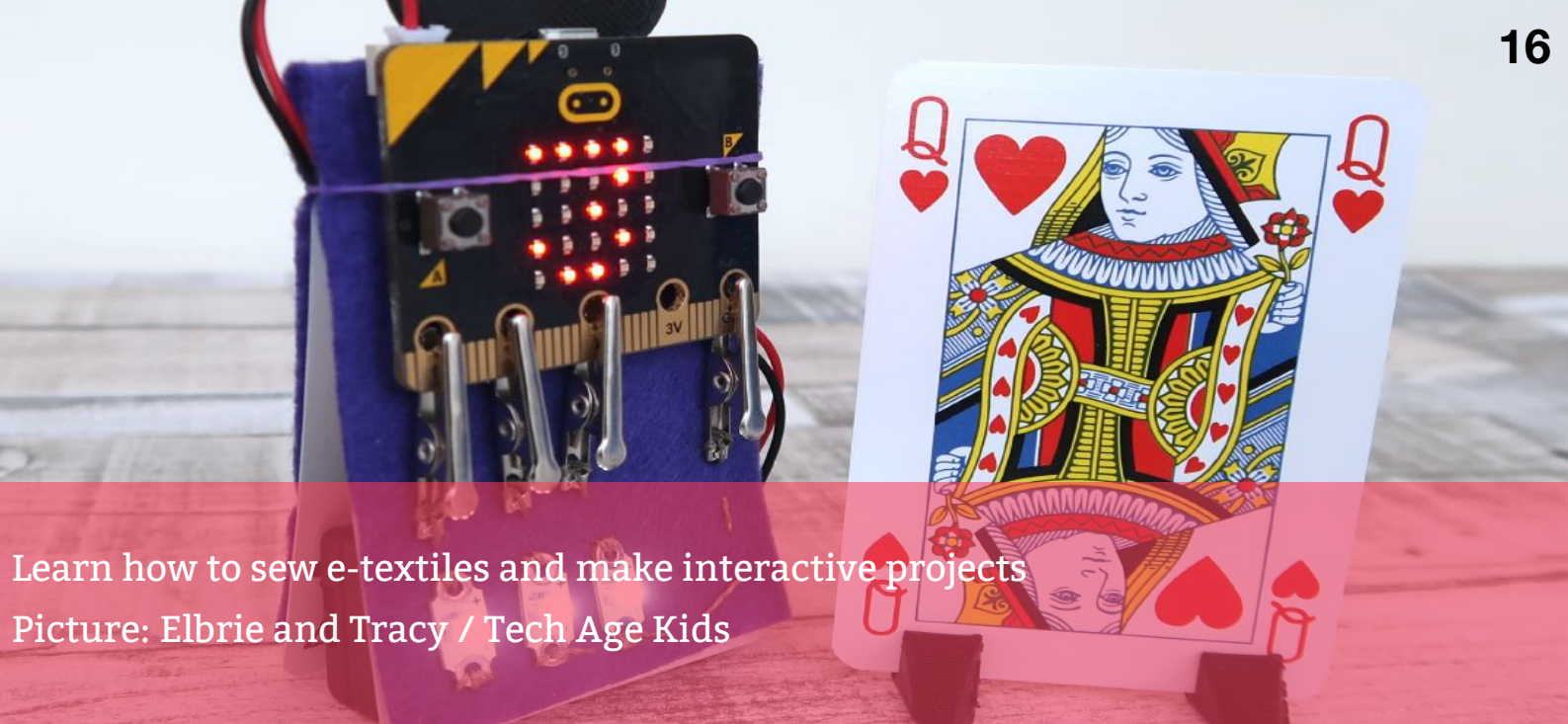
In this project you'll make an electronic price tag and a papercraft top hat to put it on. You could also sew a top hat using felt and embroidery thread. A template is available.



**YOU WILL NEED**

- micro:bit (USB cable and battery pack)
- black card (any other colour)
- elastic band (or loom bands)
- decorative material (ribbon, coloured paper, feathers, etc.)
- scissors/craft knife
- double sided tape (glue/sticky tape)
- felt, thread, needle (if sewing a top hat)

micro:bit in Wonderland      techagekids.com      10/6 The Mad Hatter's Hat ©



Learn how to sew e-textiles and make interactive projects

Picture: Elbrie and Tracy / Tech Age Kids

## Tell us about your other work that Tech Age Kids does...

**T&E:** We run the Tech Age Kids blog where we help parents to prepare their kids for the modern world. We focus on positive and creative uses of technology and support the STEAM and Digital Making Movements. We won the UK Blog Awards Digital & Technology Category last year.

We also partner with other organisations on projects where our interests align.

## What projects are you working on just now?

**T&E:** We're having a lot of fun working with the publisher Miles Kelly on books and resources for younger children aged 5-7. The Get Set Go Computing series is

coming out in July and we're working with them on further publications.

## Do you have any plans to write another book?

**T&E:** Yes, we're writing *micro:bit through the Looking Glass* at the moment. It follows the same approach but steps up the difficulty a little and uses some different features of the micro:bit and external components.

You can find out more about what Tracy and Elbrie are up to on social media here:

Twitter: [go.micromag.cc/techaa5ab7](https://go.micromag.cc/techaa5ab7)

Facebook: [go.micromag.cc/techa6c82e](https://go.micromag.cc/techa6c82e)

Instagram: [go.micromag.cc/techad7d99](https://go.micromag.cc/techad7d99)

Pinterest: [go.micromag.cc/techa955fe](https://go.micromag.cc/techa955fe)

**micro:bit in Wonderland is available from Tech Age Kids:**

**[alice.techagekids.com](https://alice.techagekids.com)**



## Mr Zbit

Is a micro:bit enthusiast who writes code and invents accessories for the micro:bit designed to help everyone 'Have fun while you learn!'



[@ZbitConnect](https://twitter.com/ZbitConnect)  
[zbit-connect.co.uk](http://zbit-connect.co.uk)

# feature:

# micro:GUI - Graphical User Interface for the micro:bit

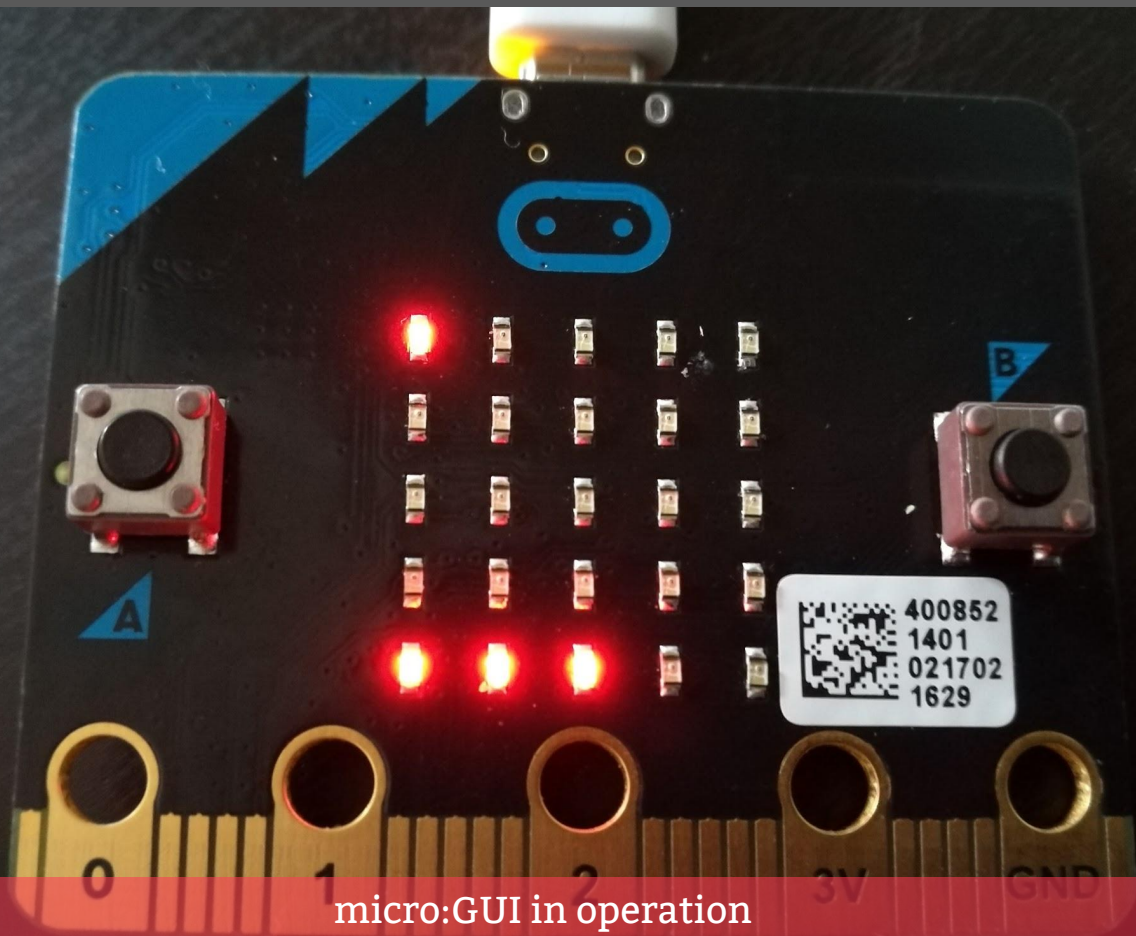
You'll be familiar with using GUIs on phones, tablets and computers - this project gives the micro:bit one of its very own.



When we interact with computers we often do so via a Graphical User Interface or GUI. On PCs most people use Microsoft Windows' GUI. On mobile phones, Apple iOS' GUI is popular. The best GUIs are what we describe as *intuitive* - easy to use and understand, even for people who have never used one before.

So, is it possible to create an intuitive GUI that can be used on a micro:bit? Well, let's see!

# :feature



micro:GUI in operation

Picture: Kerry Kidd

There are two main elements that are necessary for a GUI: a display and a way for users to input information.

**Display** - PCs and mobile phones usually have high-resolution screens, ideally suited to GUIs. A micro:bit only has a 5x5 LED Display. But, with a bit of creativity, it's possible to display lots of different images like hearts, giraffes and umbrellas on this display!

**User Input** - With a PC or phone, you normally use a mouse, keyboard or touchscreen for user input. A micro:bit has two Push-Buttons (plus a Reset Button) for User Input.

So, here's how to create a "micro:GUI" for the micro:bit!

micro:GUI's home screen has two main elements:

**The micro:cursor** - The top left LED of the home screen is the micro:cursor LED which can be thought of as acting like the cursor or pointer on a PC screen.

**micro:apps** - The bottom row of LEDs of the home screen are the micro:app LEDs, which can be thought of as being like the app icons on a mobile phone.

These are shown in the diagram on the previous page.

## Navigating the micro:GUI

The micro:bit's A & B buttons can be used to control the position of the micro:cursor LED, acting like the mouse on a PC. Pushbutton A moves the

micro:cursor down, whilst pushbutton B moves the cursor to the right. When the cursor reaches the edge of the display, it *wraps* round from right-to-left and from bottom-to-top.

## Launching a micro:app

Move the micro:cursor down onto one of the micro:app LEDs using the 'A' & 'B' Push Buttons. When the micro:cursor is on a micro:app LED a micro:icon is displayed. In the above example the micro:icon is an umbrella: this micro:app plays the tune *Singing in the Rain!*

Press buttons A and B at the same time to launch a micro:app, this acts like a double mouse click would on a computer.

A micro:app can be any micro:bit program. (This means that by using the micro:GUI it is possible to have multiple programs in a single hex file with the micro:GUI used to select which program to run). When running a micro:app, or at any other time, you can return to the micro:GUI home screen by pressing the reset button on the back of the micro:bit.

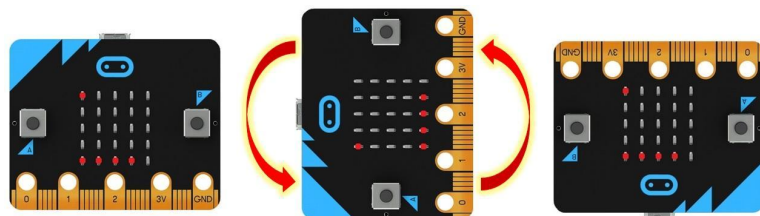
micro:GUI is all about education. It is designed to help you understand the elements of a GUI, to help you appreciate what makes a GUI intuitive, then to give you the opportunity to write the code for your own micro:bit GUI.



## Why not have a go at writing the code for your own micro:GUI?

The micro:GUI Specification, Worksheets and Examples can be downloaded from

[www.zbit-connect.co.uk/microGUI](http://www.zbit-connect.co.uk/microGUI)



## Taking the project further:

Making use of the micro:bits accelerometer (motion sensor), can you make your micro:GUI display auto-rotate like a phone's display, so the micro:app LEDs still appear at the bottom even when you turn the micro:bit upside down? Have a go!



## Chris Penn

Chris is a teacher from Warwickshire who teaches Computer Science and IT. He also runs the Coventry and Warwickshire Raspberry Jam.



[@ChrisPenn84](https://twitter.com/ChrisPenn84)

# micro:bit and Minecraft Pi

Input, process, output - in this tutorial, Chris Penn explains how to use sensors on the micro:bit to control the Raspberry Pi edition of Minecraft

## You Will Need:

- A micro:bit
- A Raspberry Pi with internet connection
- A micro USB cable

Using the micro:bit to interact with Minecraft Pi / Java edition has been possible for the last 2 years. All routes point back to David Whale, co-author of Adventures with Minecraft (AWMC). The first route was usable but quite technically fiddly. While writing his updated version AWMC David refined the above method and made it super to code with the micro:bit and Minecraft. This method is called BitIO Python library available here link

[go.micromag.cc/bitiolib](https://go.micromag.cc/bitiolib) In essence, it allows you to code the Micro:bit using MicroPython as a Python3 library. You can read about 13 of my experiments with this library here:

[go.micromag.cc/jammye6c46](https://go.micromag.cc/jammye6c46)

BitIO is a library developed by David Whale. It allows you to code your micro:bit as part of a normal Python3 program by importing it. This is very

handy for incorporating the micro:bit into larger Python projects, for example, weather stations e.g. pressing the "A" button to trigger a reading or using the accelerometer to control your player in Minecraft.

The final piece of context is that this BitIO has recently been added to the EduBlocks visual coding platform created by Josh Lowe. For clarity, there are two versions the excellent version which works with coding MicroPython directly on the micro:bit which works on any internet connected PC available here [go.micromag.cc/edublocks](https://go.micromag.cc/edublocks) The original EduBlocks works on the Raspberry Pi and creates a drag and drop IDE that allows you to snap into place programs in Python. This is the version we will be using to code with, let's start.

## Installing

To install EduBlocks you will need to:

1. Open up a Terminal window, this can be done by clicking on this icon in the top left-hand corner of the Pi's screen.

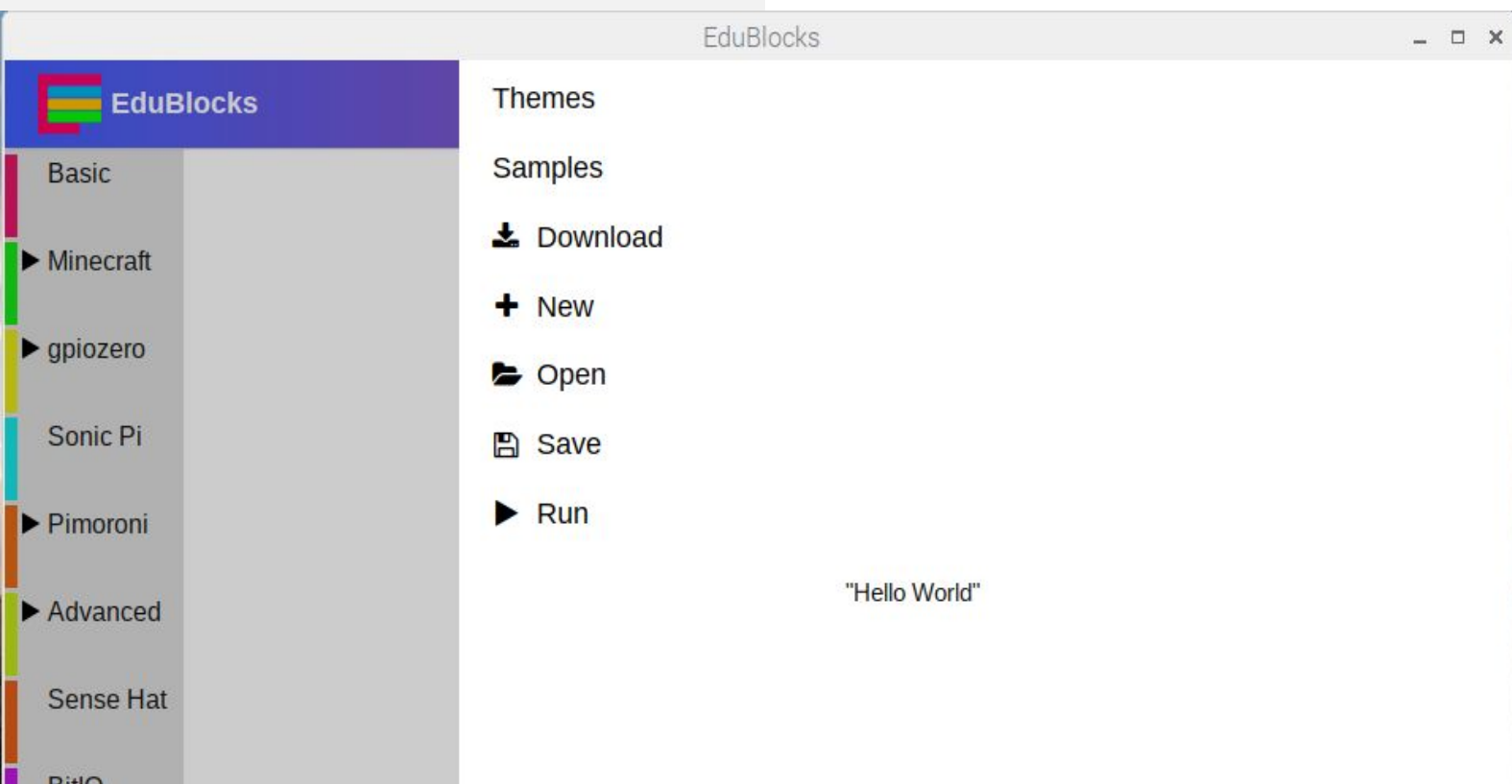
2. Type the following command to install EduBlocks

```
curl -sSL get.edublocks.org | bash
```

As long as you have an internet connection this should install it may take a few minutes.

When it is installed then you can access the software from a shortcut on the desktop or the programming menu.

3. Plug in your micro:bit to one of the available USB ports.
4. You will need to download this file and drag a copy of it onto the micro:bit. This file allows the micro:bit communicate with your program. [Link  
go.micromag.cc/bitio6464b](https://go.micromag.cc/bitio6464b)



## Coding

1. Open Minecraft and create a new world, then minimise this for later. Tip, use the tab key to allow your mouse to minimise the Minecraft window.
2. Open EduBlocks from the programming menu, this should open in the chromium browser.

5. We will start out with the classic 'hello world'. When the 'A' button is pressed the text will scroll across the micro:bit screen.
6. Using the BitIO and Basic tabs on EduBlocks you should be able to easily snap together the program

# :make

7. When your program is complete you will need to click on the 'menu' button on the right-hand side. Then select run to see your program in action.

8. When you run the program EduBlocks will show a black screen which will show you the shell output. This tells you what is going on with your program. First of all, it will display this message:

*No micro:bit has previously been detected  
Scanning for serial ports  
remove the device, then press ENTER*

9. So unplug the micro:bit and press enter.

```
import microbit
```

```
while True :
```

```
    if microbit.button_a.was_pressed() :
```

```
        microbit.display.scroll( "Hello World" )
```

10. Next, the following text will display:

*Scanning...  
found xx device(s)  
plug in the device, then press ENTER*

11. Now press enter and the following text will display:

*Scanning...  
found 68 device(s)  
found 1 new device  
selected:/dev/ttyACM0*

*Do you want this device to be remembered? (Y/N)*

12. Press 'y' and enter to confirm and the following text will display:

*Your micro:bit has been detected.  
Now running your program*

Now your program should be working and your Micro:bit should display the message that you have entered into your code.

Well done you are a successful coder :)

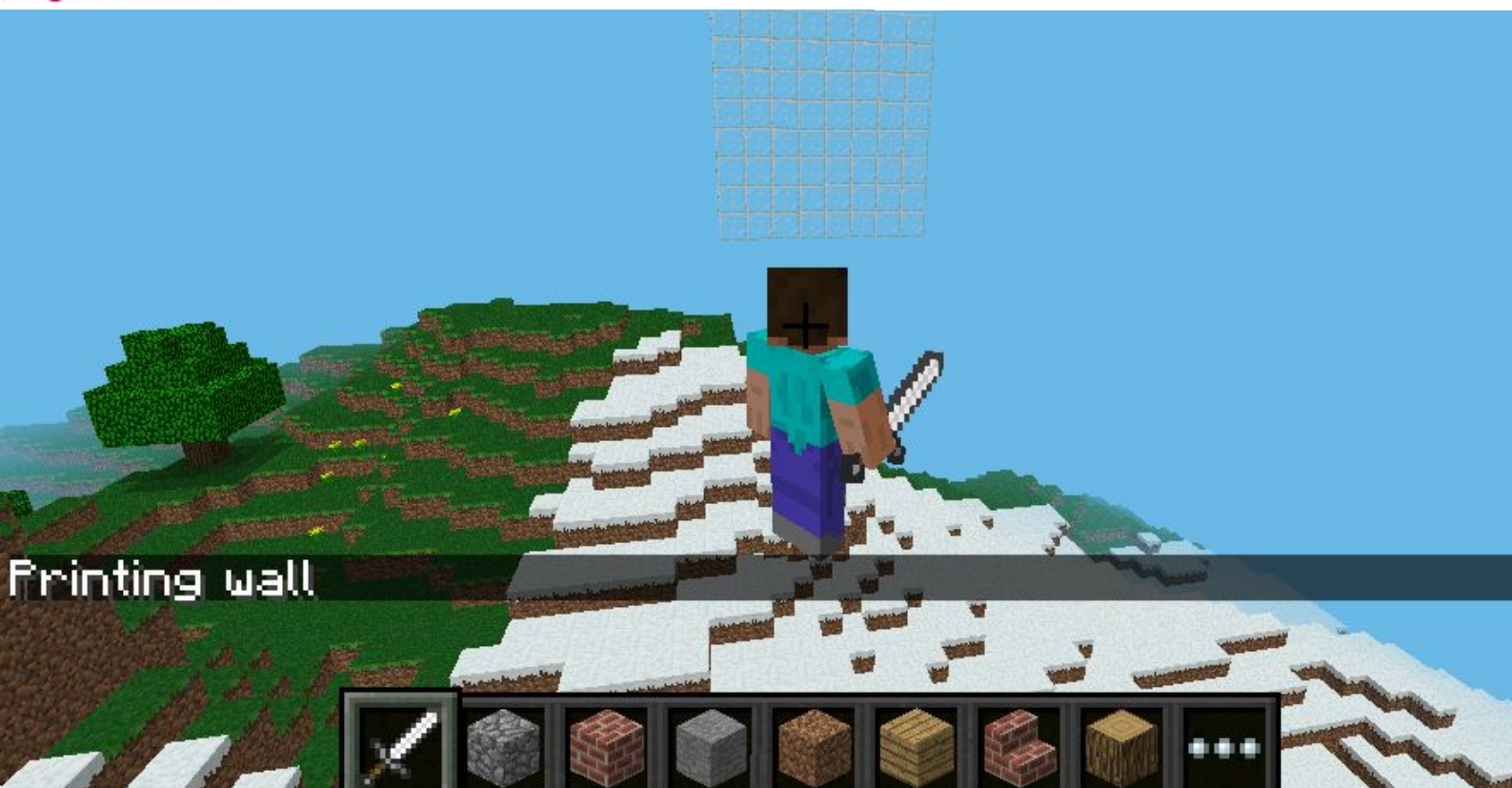
## Now for the Minecraft stuff!

1. We will now use the Minecraft tabs and the basic tabs to create the blocks that will create the code listed below in the next picture. This is our EduBlocks code, you can see that the Python text is on the blocks that you drag and drop. This is to help you learn how to eventually type out Python code. This is a really easy way to get started with Python and you can build simple and complex programs with it.

2. You will need to run this in the same way as before. It should create a wall of glass when the 'a' button is pressed on the Micro:bit.



3. Adapt your code to look like the code example on the next page
4. Run the code as you did before.



# :make

```
from mcpi.minecraft import Minecraft
import microbit
from mcpi import block
import random
mc = Minecraft.create()

Blocks_List = [2,3,4,5,15,16,17,18,19,20,45,46,47,246,247]

while True:
```

```
    if microbit.button_a.was_pressed():
        microbit.display.scroll( "Print wall" )
        RB1 = random.choice(Blocks_List)
        RB2 = random.choice(Blocks_List)
        RB3 = random.choice(Blocks_List)
        RB4 = random.choice(Blocks_List)
```

```
        playerPos = mc.player.getPos()
```

```
        PrintWall(PixelArt(
```

```
            RB4 , RB2 , RB1 , RB1 , RB3 , RB4 , RB2 , RB4 ,
            RB3 , RB1 , RB1 , RB3 , RB2 , RB4 , RB3 , RB2 ,
            RB1 , RB4 , RB3 , RB1 , RB3 , RB2 , RB1 , RB4 ,
            RB2 , RB3 , RB4 , RB1 , RB3 , RB3 , RB1 , RB2 ,
            RB4 , RB2 , RB1 , RB3 , RB2 , RB2 , RB2 , RB1 ,
            RB3 , RB4 , RB2 , RB1 , RB4 , RB3 , RB1 , RB3 ,
            RB2 , RB4 , RB3 , RB1 , RB2 , RB4 , RB3 , RB4 ,
            RB1 , RB4 , RB4 , RB3 , RB2 , RB3 , RB1 , RB4 ,
```

```
        ))
```

**Taking the project further:**

Now see what you can do to extend this basic hack!





## Mark Vanstone

Mark is an educational software author, consultant, animator, musician and stem ambassador.



@MindExplorers

[technovisualeducation.co.uk](http://technovisualeducation.co.uk)

make:

# Sending and Receiving Radio Messages

In this make, you'll learn how to get two micro:bits talking to each other using the radio function, using either MakeCode, EduBlocks or Python to write code.

## You Will Need:

- A micro:bit
- A computer with internet access
- A micro USB cable

## Plugging in a micro:bit to a Computer

Start by connecting the micro:bit to the computer. If you have used a micro:bit before you probably already know how to do this but if not, you can find the instructions online at

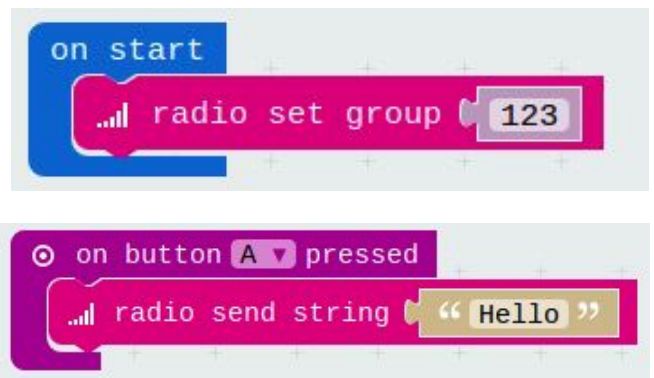
[go.micromag.cc/quick917d5](http://go.micromag.cc/quick917d5)

A little further down that page, you will see instructions about how to get started using the MakeCode editor.

[go.micromag.cc/makecode](http://go.micromag.cc/makecode)

## Writing A Radio Program With Blocks

We can write the message sending part of our program using the following blocks:



The group number can be between 1 and 255 but it must be the same for the sender and the receiver.

Now we can add a block to receive a message and do something to show it arrived.



# :make

Now we can flash this program to the micro:bit. Click the Download button. Then on the computer, drag and drop the downloaded file to the 'MICROBIT' USB drive. The orange LED on the back of the micro:bit will flash for a while – wait until it is finished before trying the code.

## Sending Radio Signals

When we send a radio message we will want to set up another micro:bit to receive the message. We can use exactly the same program for our receiver micro:bit.

Make sure the program has finished flashing to the first micro:bit, then unplug it from the computer and do the same for a second micro:bit.

To test our program you will need to plug a battery pack (or another USB cable) into the first micro:bit so that it can switch on.

To test our program we press the 'A' button on the first micro:bit. We should see a tick on the second micro:bit. Then press the 'A' button on the second micro:bit and you should see a tick on the first micro:bit. Both should now be showing a tick.

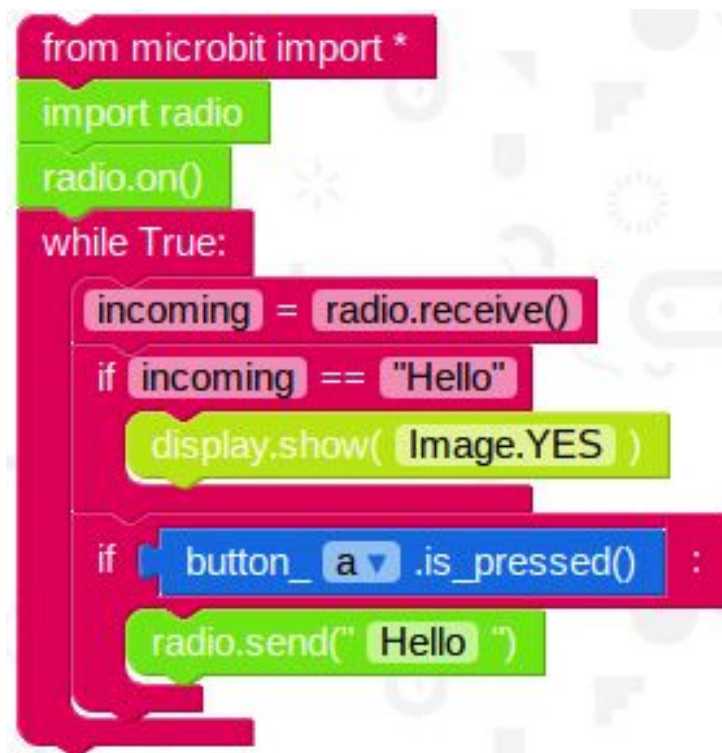
## Writing The Program With EduBlocks

EduBlocks is a bit like MakeCode except it allows you to program more like the Python language and switch between blocks and text-based Python code.

To use EduBlocks, open a browser and go to the following website:

[go.micromag.cc/edublocks](http://go.micromag.cc/edublocks)

To write the program we need to put together the following blocks:



When the program is ready, you can flash it to both micro:bits using the 'Download Hex' button. Have a look at the Python version of the program – you can switch between the two on EduBlocks.

## Writing The Program With Python

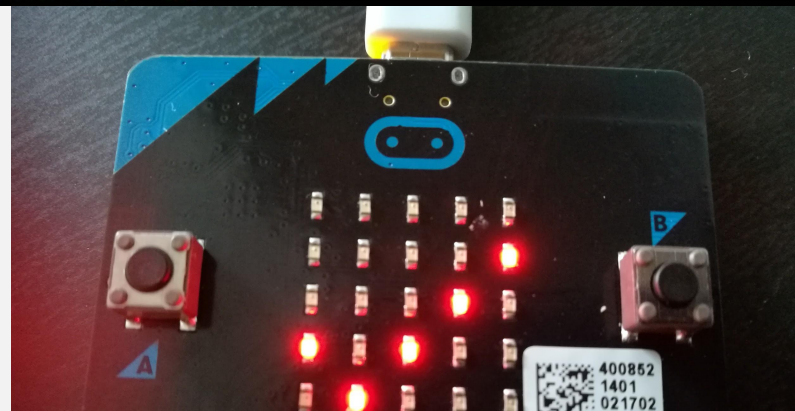
When we write a program with Python, we don't use any blocks but type it into an editor. In this example, we will be using the online micro:bit Python editor. First, open a browser and go to [go.micromag.cc/pyeditor](http://go.micromag.cc/pyeditor)

You will see an example program already there. Replace the example with the code on the following page.

```
1 import radio
2 from microbit import display, Image, button_a
3
4 radio.on()
5
6 while True:
7     incoming = radio.receive()
8     if incoming == 'Hello':
9         display.show(Image.YES)
10    if button_a.was_pressed():
11        radio.send('Hello')
```

When your program is ready, make sure that your micro:bit is plugged into the computer and flash the program by selecting the “Download” button.

Then test your code by flashing the program to the other micro:bit. Press the A button, and the other micro:bit should display a tick too.



The working micro:bit code - message received!

Picture: Kerry Kidd

## Taking the project further:

- Try changing the program so that the message that is sent gets displayed as scrolling text across the LED matrix of the other micro:bit.
- See how far away the two micro:bits can be from each other before they stop picking up the message that is being sent.
- Try sending messages in a chain between many micro:bits so that one micro:bit sends a message to a second and that one sends a message to a third and then a fourth and each one will display a different symbol on its LED matrix as it receives its message.



# Get fresh copies of micro:mag delivered to your inbox

The background of the advertisement features two Micro:bit boards. One board is positioned in the upper right corner, and another is in the lower left corner. Both boards are black with yellow accents and have several red LEDs lit up. The boards are slightly angled, giving a sense of depth.

Make sure you never miss an issue  
of the unofficial community  
magazine for micro:bit lovers -  
we'll email each issue as soon as  
it's released (and we won't ever  
spam you or give your details to  
anyone else)

To sign up, go to  
[micromag.cc/email](https://micromag.cc/email)

# Niels Swinkels

Niels is an interaction design researcher at RISE in Sweden. He has introduced many teachers to programming with micro:bit.



[mermicrobit.se](http://mermicrobit.se)

make:

## Hide & Seek A micro:bit

Niels will show us how to make a fun hide and seek game using the radio function on the micro:bit. If you only have one micro:bit grab a friend and program this :make together.

### You Will Need:

- 2 x micro:bits
- 2 x battery packs for the micro:bits
- A computer with internet access
- A micro USB cable

With the radio functionality of your micro:bit, it can send and receive information from other micro:bits.

This can be used in many fun ways. How about playing some hide & seek? In this tutorial, we will program one micro:bit (the hidden one) to reply to any radio message it receives. Another micro:bit (the one that searches) will call out via the radio and show on the display how strong the signal is of the answer it receives. Then by keeping an eye on this signal strength while walking around, we will know when we are close by or far away from the hidden micro:bit.

### The hidden micro:bit

Let's start by programming the hidden

micro:bit. The only thing it needs to do is send out something via the radio. To save battery power, we will only make it send out something after it has first received something via the radio. This way it will not do anything if no one is searching.

From the Radio category, drag out the **on radio received receivedString** block. Make sure you take the one with **receivedString** because string means“ a piece of text” and we will be sending and receiving text.

Inside of this, place a **radio send string** block. Which text it will send does not really matter, so, for example, you can write “bit!”

The last thing we need to add is to choose which radio group to use. You can compare this with normal radio stations, which you can only listen to when you tune in on the right frequency. Drag in the **radio set group** block to **on start**, and set it to any number from 0 to 255.



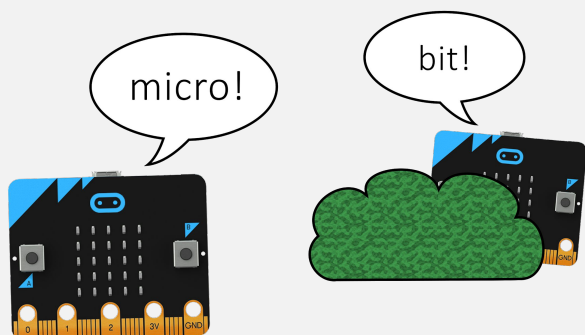
Make sure you remember the number because we will need to set the other micro:bit to the same group!

on start

radio set group 135

on radio received receivedString

radio send string " bit "



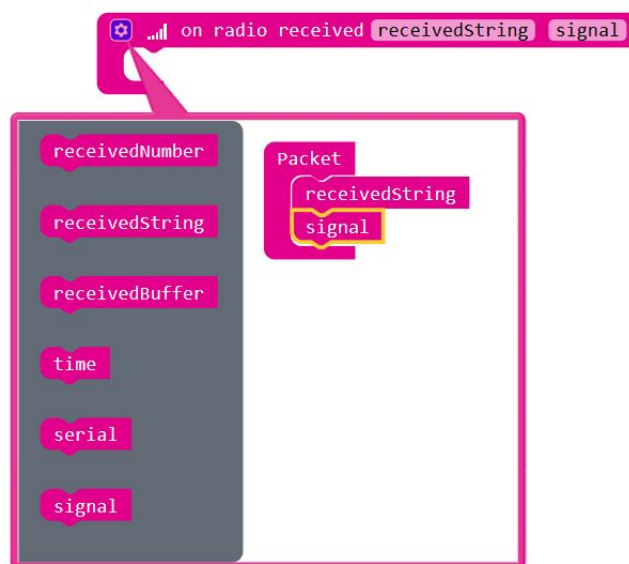
## The seeking micro:bit

Next up is the micro:bit that will search. This one will continuously send out a word, listen for an answer, and show how strong the signal is. Let us start by setting it to the same radio group as our hidden micro:bit, now that we still remember the group number. Drag in a **radio set group** to **on start** and fill in the number.

Now take a **radio send string** block and place it in the **forever** loop. This way it will call out to the hidden micro:bit as long as it is turned on. Write in any word you like, for example, "micro". To make sure the other micro:bit has some time to answer, we add a **pause** block to the **forever** loop with a value of 500 milliseconds (that is half a second).

Drag out an **on radio received** **receivedString** block. Here we will listen for any answers.

Because we want to know how strong the signal is of the answer, we need to customise this block. Click the small blue gear icon on the block, and drag the **signal** block from the left to fit after the **receivedString** block. Click the blue gear icon again to close the pop-up.



An easy way to show the signal strength is by using the built-in bar graph.

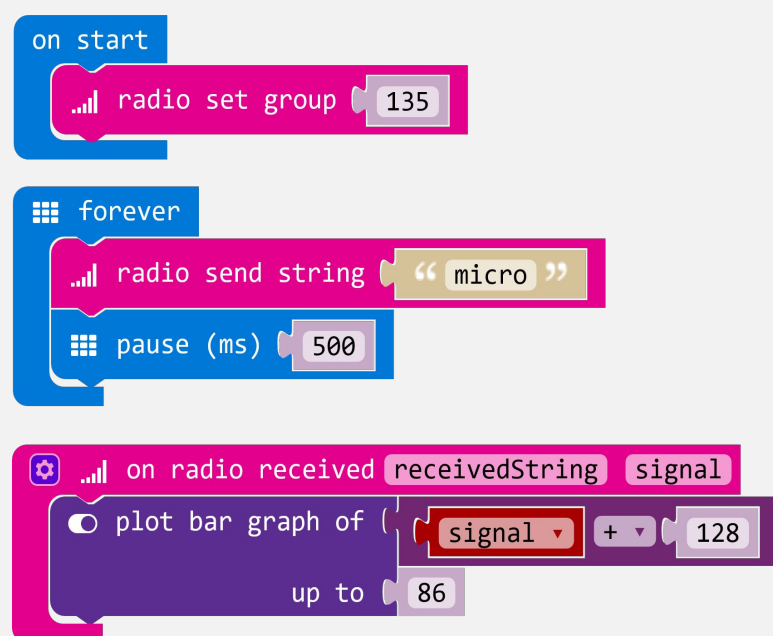
From the **LED** category, take the **plot bar graph** block and place it inside the **on radio received**. This block will fill the screen, based on the value you connect to it on the first position. The higher the value, the more LEDs go on, up to a maximum value that you set at the second position.

Now we want to show the signal strength that we get from the **on radio received** block. We will find this as a **signal** block in the **Variables** category, but we first

need to know what kind of values this signal strength will have.

Right-click the **on radio received block**, and choose **help**. There we read that the signal ranges from -128 for very weak to -42 for very strong. To be able to draw a bar graph from this, we should change this to be a range of positive numbers, with the lowest number being zero. We can do that by adding 128 to the signal we get.

Then the range will be 0 for very weak to 86 for very strong, so drag out a **+** block from the **Math** category and put the **signal** block in it on one side, and type **128** on the other side. Connect this to the first position of the **plot bar graph** block. On the second position, we write the maximum value, which is 86(128-42).



## Time to play!

Let's test our game. Plug in a battery pack to the hiding micro:bit, and hide it somewhere in the room, then plug in a battery pack to the seeking micro:bit and walk with it through the house. It should show you more lights the closer you get to the hidden micro:bit.

With some experimenting, you can find that objects, such as walls and your body, will block the signal, and make it a lot weaker. If you do not get any signal at all, go back to your code and search for bugs(mistakes in the code). The first thing to double check is if you put in the same radio group for both micro:bits.

## Taking the project further:

You can continue to build on to your game. How about making the seeking micro:bit play different sounds depending on the signal strength? Or being able to search for more than one micro:bit? Can you use something else instead of the bar graph to show how close you are?



# Advertise in micro:mag!

micro:mag is the community magazine for micro:bit lovers - if you want to reach an audience of students, teachers and hobbyists at reasonable cost, micro:mag is the place to do it! We've got full and half pages available, at reasonable rates, and you'll help cover the costs of producing the magazine.

Get in touch for more  
info - email us on

[hello@micromag.cc](mailto:hello@micromag.cc)

## Les Pounder

Les is a maker and trainer who has worked with the Raspberry Pi Foundation and the BBC to deliver computing training.



[bigl.es](http://bigl.es)  
@biglesp

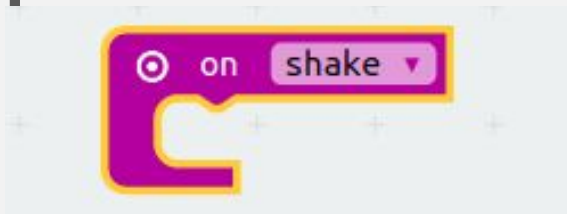
# micro:hit

# A game of chance

**Heads or tails?** A game where you have a 1 in 2 chance of winning, has been used to decide many things such as who kicks off at a football / soccer match and who gets the last sweet in the bag.

In this project you will create your own digital game of chance using a micro:bit and a few blocks of code to decide... skulls or snakes?! For this project you'll be using the Javascript Blocks Editor [go.micromag.cc/makecode](http://go.micromag.cc/makecode) and you should already be comfortable with copying code to your micro:bit.

## Step 1: Working with input



From the Input menu drag the “on shake” block into the coding area. Any code inside the block will be run when the micro:bit is shaken.

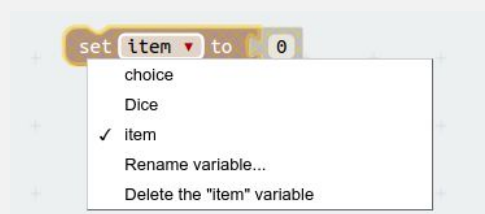
## Step 2: Create a variable

New variable name:

Ok
Cancel

In the Variables menu click on ‘Make a Variable’ and call it ‘choice’ We shall use this variable to store a random number later.

## Step 3: Using the variable



From the Variables menu drag the ‘set item to 0’ block to the coding area. Now click on the arrow next to ‘item’ and change it to ‘choice’.

## Step 4: Random number generator



From the Random menu, drag ‘pick random 0 to 4’, and change ‘0 to 4’ so it reads ‘0 to 1’. Then connect it to “set choice to” block. Lastly place these blocks inside the ‘on shake’ loop.

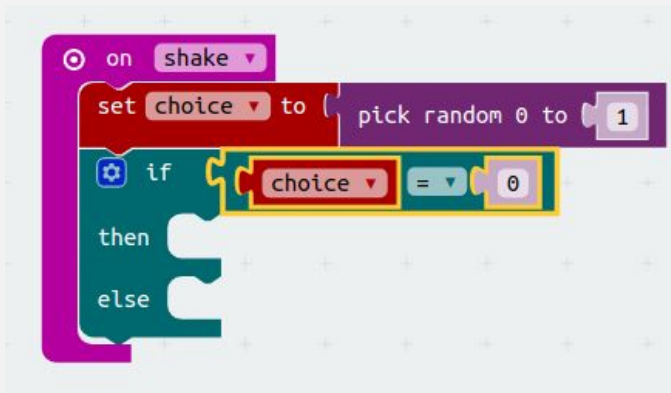


## Step 5: Logical comparisons



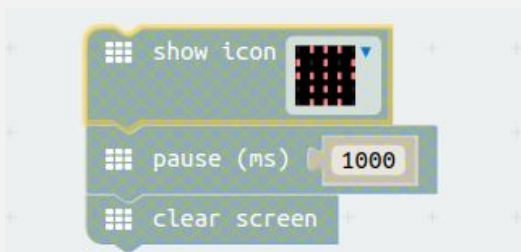
Go to the Logic menu and drag an 'if true..then..else' block, and place it inside the loop, but under our code. This block will run code if a condition is met.

## Step 6: Decisions...



Our first block is from Logic and it is '0 = 0' drag this and connect it to the 'if'. Next from Variables drag 'choice' and place it where the first '0' is.

## Step 7: Skulls!



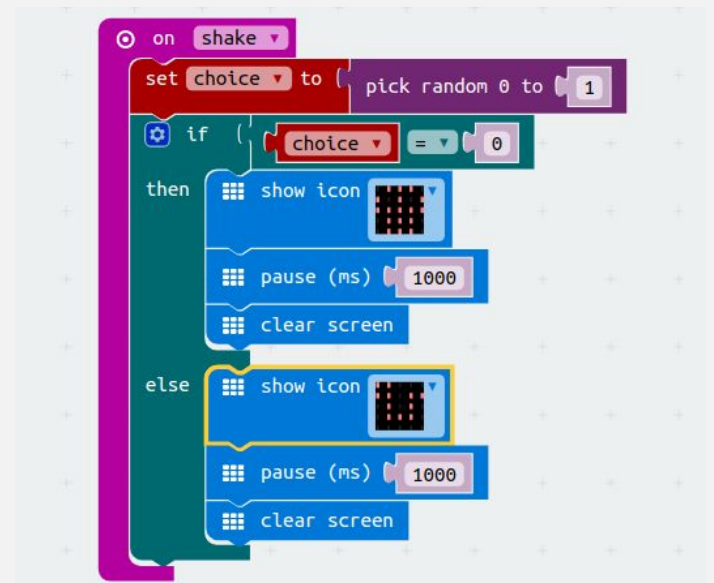
All of the blocks in this step are from Basic: first we'll use the 'show icon' block to display a skull. Then we pause the image for 1 second before clearing the screen.

## Step 8: Snakes!



Here we repeat step 7, but have a snake appear on the screen. Remember the clear screen block can be found under Basic → More.

## Step 9: Final code



Our skull block of code is placed inside the 'then' part of the logic, and the snake is placed inside 'else'. All of the code for this project should now look like this.

That's it, now download the code to your micro:bit and when it has finished, give the micro:bit a shake to see either a skull or a snake appear on the LED grid!

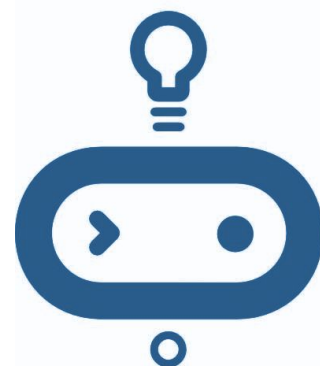
**Well done! You've made your own decision making micro:bit!**

## Gareth Stockdale

Former head of operations at BBC learning, which lead the original micro:bit project. Now, CEO of Micro:bit Educational Foundation.



[@stockers1001](#)



# A Message of support

From the CEO of the Micro:bit Educational Foundation

Here at micro:mag, we have been thrilled to have support from the community and the Micro:bit Educational Foundation since the start, and are very grateful for all the support we have got so far. So, here is a message from Gareth Stockdale, the CEO of the foundation, to give his message of support for micro:mag.

We're delighted to support the micro:mag team with their new venture and we're always pleased to hear about events and goings-on within the micro:bit community.

We continue to build on the strength of the original BBC project which empowered students, 90% of which say the micro:bit helped show that anyone can code. This would not be possible without the support from our Community, for which we are immensely grateful.



The BBC micro:bit continues to inspire students across the world as well as helping teachers deliver Computer Science lessons.

Picture: Micro:bit Educational Foundation

"It has been amazing to see the global appetite for the BBC micro:bit and its ability to widen participation in digital creativity at such low cost" said Gareth Stockdale, CEO. "2018 is an exciting year for the Foundation and we're looking forward to sharing our latest news with micro:mag and their readers"





# Getting to know the micro:bit foundation

We thought it would be nice for micro:mag readers to get to know the Foundation team a little better, so we posed them the question “What is your favourite burger?”

**Gareth** - Mushroom double Swiss they used to do at Burger King, end of an era when they stopped doing it. Also a burger that I had at the weekend in Croatia, no idea what was in it but at least 2 different types of meat!

**Jonny** - Steak and Honour Mushroom & Cheese (veggie, of course)! (Steak and Honour are a burger van in Cambridge)

**Kavita** - My new favourite is a beetroot burger 🍷 smothered in hot sauce.

**Howard** - I had a good burger from a van outside the big B&Q in Cribbs Causeway the other day.

**David** - Five Guys burgers win for me, most of my best resources were written sitting in Five Guys in Harlow!!

**Mark** - All the burgers, in the spirit of 'Inclusivity' or a Halloumi burger from the van outside Penryn B&Q

**Waris** - Yakisoba Pan, translated as noodle in bread/bun.



**Emma** - My favourite is a simple beef, cheese and sauce!

**Rachel** - 3) Reuben burger 2) Lamb burger with Caramelised onions and Rocket 1) Ultimate burger is a double patty, with cheese, bacon, burger sauce, and gherkins! You have to have gherkins with a burger. The bun needs to not be too sweet (brioche rarely impresses me with a burger), the beef patties have to be juicy, and please ask me how I want it cooking.. it's medium rare btw. If I'm served a side of coleslaw with my burger you get an extra point. So, basically, just cook the burger well, and make sure the beast as a whole has good structural integrity, if I have to use a knife and fork you've made it wrong.

Tina is an International Teacher of ICT and Computer Science, currently teaching at The British School in Caracas, Venezuela. Proud Mum, Grandma and Self Confessed Techaholic.



[@TinaFountain7](#)

# micro:bits Come to the British School Venezuela

Read about Tina's experience of finding out about the micro:bit then taking this new found tech to her new job in Caracas, Venezuela.

## Life Goals:

Tina Would like to travel, shop and create as much as possible (eventually find time to run a blog)

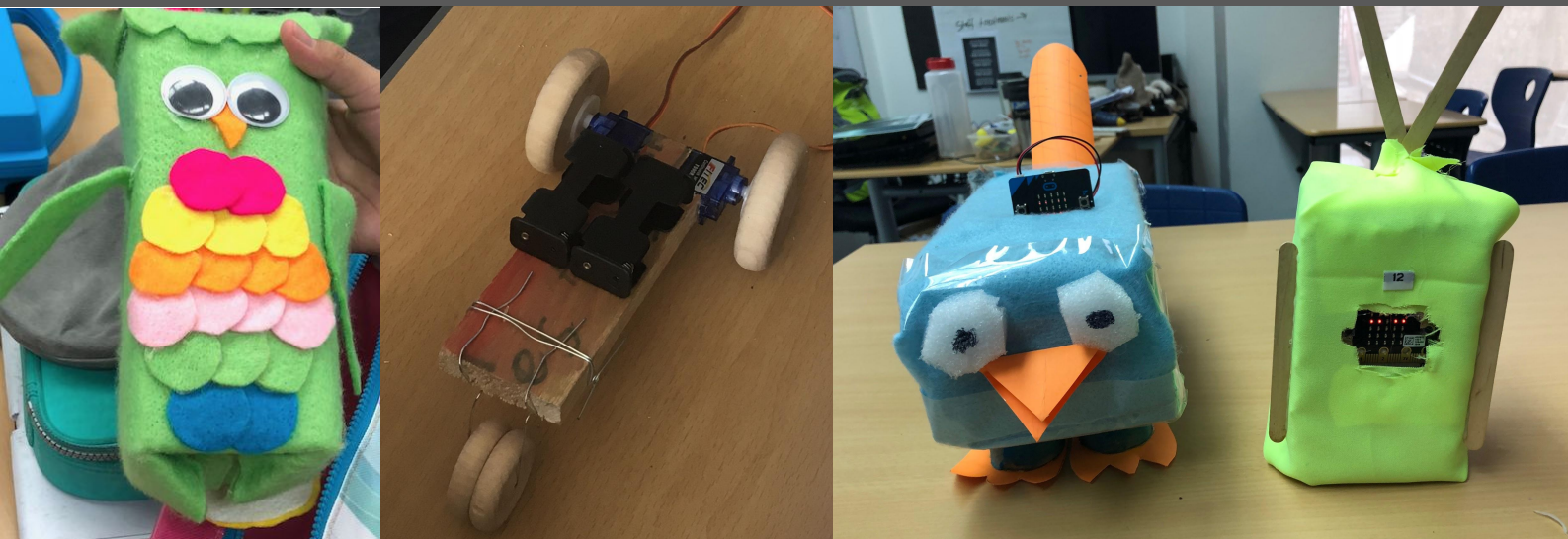
In June 2017 at Exabytes in Bradford, I was introduced to the micro:bit by Chris Sharples and Jon Witts and I was instantly hooked. I went home and emailed my soon-to-be head of school and asked for the money to buy 20 micro:bits. He said yes and within days I had a class set of micro:bits packed in my suitcase ready for my departure to Caracas.

Never did I imagine how much confidence the students would get from such a tiny little device. When I introduced year 9 to the fact that the first unit of study involved block programming I received a very negative response: they were all 'Scratched out',

yet after only one lesson the whole class was fully engaged - one pupil even went straight home and ordered his own micro:bit!

Within weeks the students were confident at programming the bits independently with the blocks editor and ready to transition to Python. They understood the concepts and felt confident with their ability, meaning the transition to text-based programming was much easier than I have experienced before. The wow factor of making something physical work takes away the fear of the programming.

There was a buzz in the lessons when the micro:bits where flashed and the code came to life. The web has so many amazing resources for the micro:bits it is impossible to run dry of fun ways to teach the concepts of programming.



Tina's students have created all manner of inventions with micro:bit - from noisy owls to robot buggies to virtual pets!

Picture: Tina Fountain

I really wanted the students to have some experience with robots but as a new department, wanting to introduce robotics for the first time, I needed to keep the costs to a minimum. I trawled the web for inspiration and found a couple of posts that showed two servos running from the 3v pin so decided to give it a go. On a short visit to the UK, I bought 20 servos, a couple of packs of crocodile clips and some pin wires, this was one of the best £60 I have ever spent.

I managed to make my first ever very basic two-wheeled micro:bit robot from an old plastic food container, 2 servos and crocodile clips. When the students saw my prototype they were incredibly excited to get started, the creativity was astounding. Working in pairs the students used whatever they could get their hands on to create a body and explored options for both powering the

bots and also controlling them. Not every bot worked as expected on the day of the wars but a lot of learning had taken place and the students were proud of what they had achieved.

Since then Year 7 have created the most amazing virtual pets utilising so many of the input sensors, they have hacked up old headphones and used servos to create moving body parts, one girl even created her own power pack to allow projects to run for longer. They have shown no fear in their endeavours, are very confident in writing and reading the code and have produced excellent annotations showing they understand the concepts. More importantly, the micro:bits have given *me* more confidence to try out new ideas and be creative in my classroom. I can't wait for next year - and for all the new equipment I have ordered to complement the micro:bits to arrive!

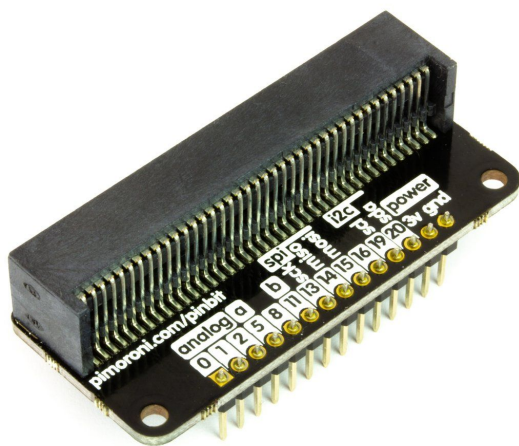


# Pimoroni

## pin:bit

The first of Pimoroni's range of playful micro:bit add-ons lets you build bigger and better projects faster.

# 8/10



The pin:bit arrives pre-soldered and ready for use

pre-soldered and is ready to use for projects straight out of the box.

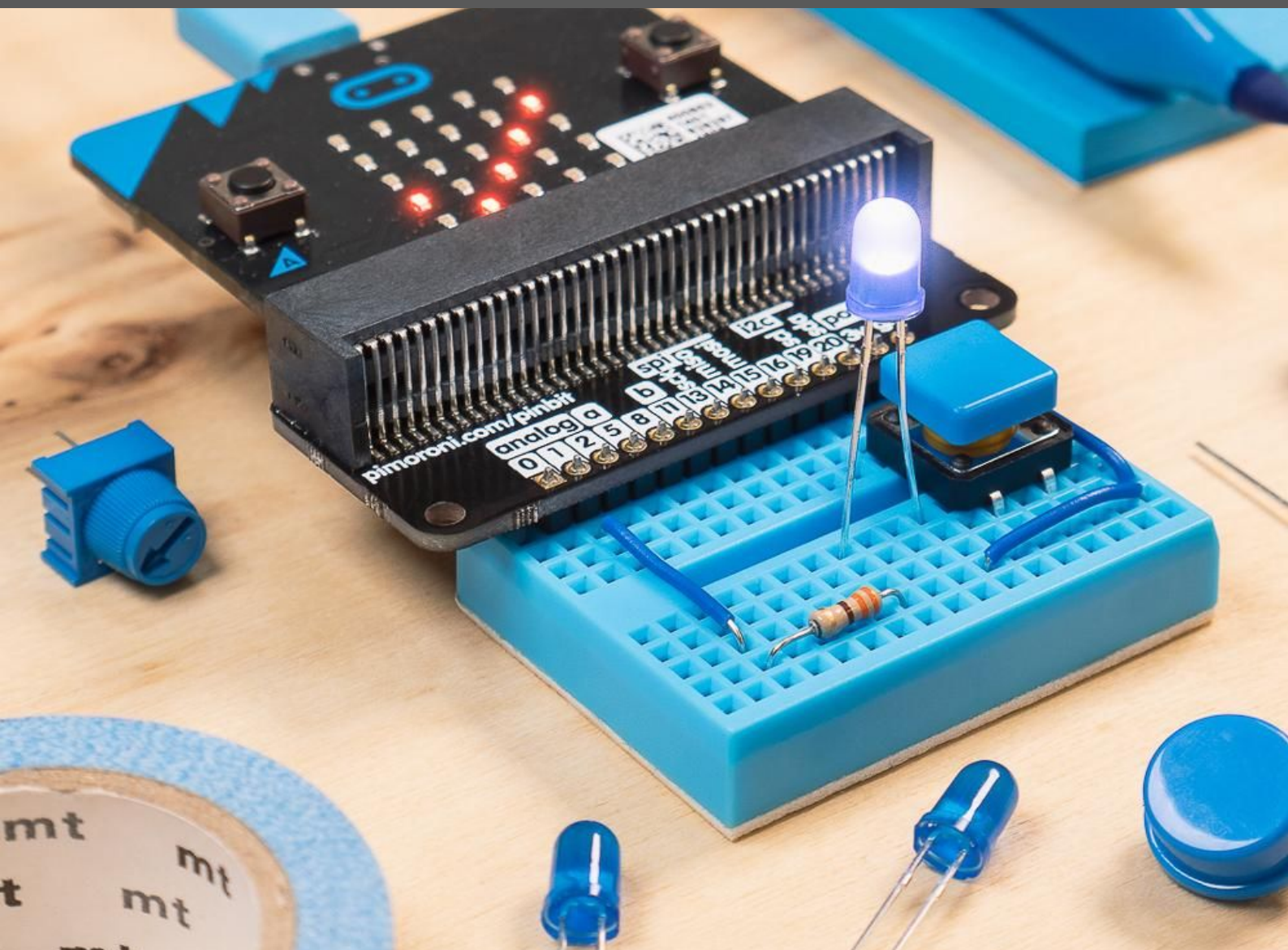
It's worth noting that, unlike most of its competitors, the pin:bit doesn't break out all the pins on the edge connector of the micro:bit; but focuses on a handy selection (leaving out those used for the built-in LED matrix). This makes the board arguably easier to use for beginners - and also makes it compatible with smaller 140-point breadboards, which are commonplace in schools and makers' toolboxes. The pins themselves are helpfully labelled in a friendly way, and the design is lovely and playful.

We tested pin:bit with an Adafruit NeoPixel Ring, some pushbuttons and a microphone - and it made prototyping quick and easy, allowing us to use all different pins for more advanced projects. You could also use the pin:bit to hook the micro:bit up to more permanent projects, by soldering directly to the pins or using a piece of stripboard to make connections.

*(Image courtesy of Pimoroni)*

Maker giant Pimoroni has recently launched the first few members of its new family of micro:bit accessories. Alongside add-ons for environmental sensing, sound output and display, the simplest board in their armoury is pin:bit, which converts the edge connector on the bottom of the micro:bit to a set of breadboard-compatible, 0.1"-spaced pins.

The board costs a fairly reasonable £5 and is available from Pimoronis own website - with re-sellers expected to stock the product soon. It arrives



There aren't currently any learning resources for pin:bit on the Pimoroni website - but there is an excellent guide to the micro:bits pins on the micro:bit Foundation's website at [go.micromag.cc/pins435fd](https://go.micromag.cc/pins435fd) which can be used with pin:bit. The pin:bit also hangs down a little if placed on the edge of the breadboard - but in practice, this isn't really a problem and the connections are sound.

## Where can I buy one?

Pimoroni shop:  
[go.micromag.cc/pinbit](https://go.micromag.cc/pinbit)

Overall, the pin:bit is a great way to get started with using more pins of the micro:bit for more advanced projects - and the friendly design and ease-of-use make it ideal for use with kids, as long as you have learning materials to hand.

## The competition....

4Tronix make a wide range of micro:bit breakouts at

[4tronix.co.uk](https://4tronix.co.uk)

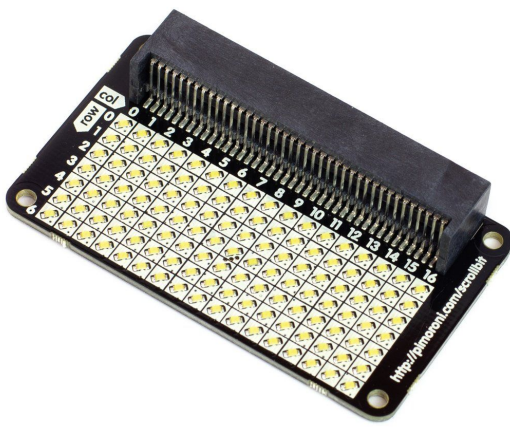
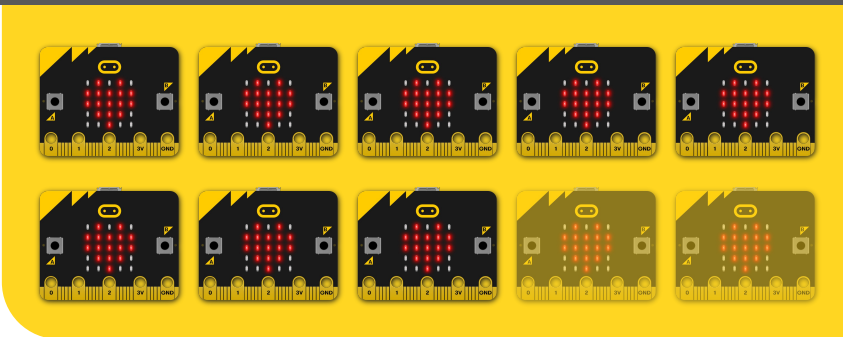
Kitroniks edge connector breakout is available from

[kitronik.co.uk](https://kitronik.co.uk)

# Pimoroni scroll:bit

A fun display with a lot of pixels! It's 119 bright white LEDs are perfect for scrolling messages with your micro:bit, or for animations, graphs, and more!

8/10



The scroll:bit arrives pre-soldered and ready for use

(Image courtesy of Pimoroni)

The Sheffield pirates Pimoroni have just released their first four micro:bit add-ons and we have been lucky enough to get our hands on the pin:bit (reviewed on the previous page) and the scroll:bit.

The scroll:bit is made up of 119 white LEDs arranged in a 17x7 matrix. Each LED can be controlled independently of each other. scroll:bit connects to the micro:bit using the edge connector. The scroll:bit is priced at a reasonable £13.50 and is available from Pimoronis own website.

The scroll:bit comes pre-soldered and

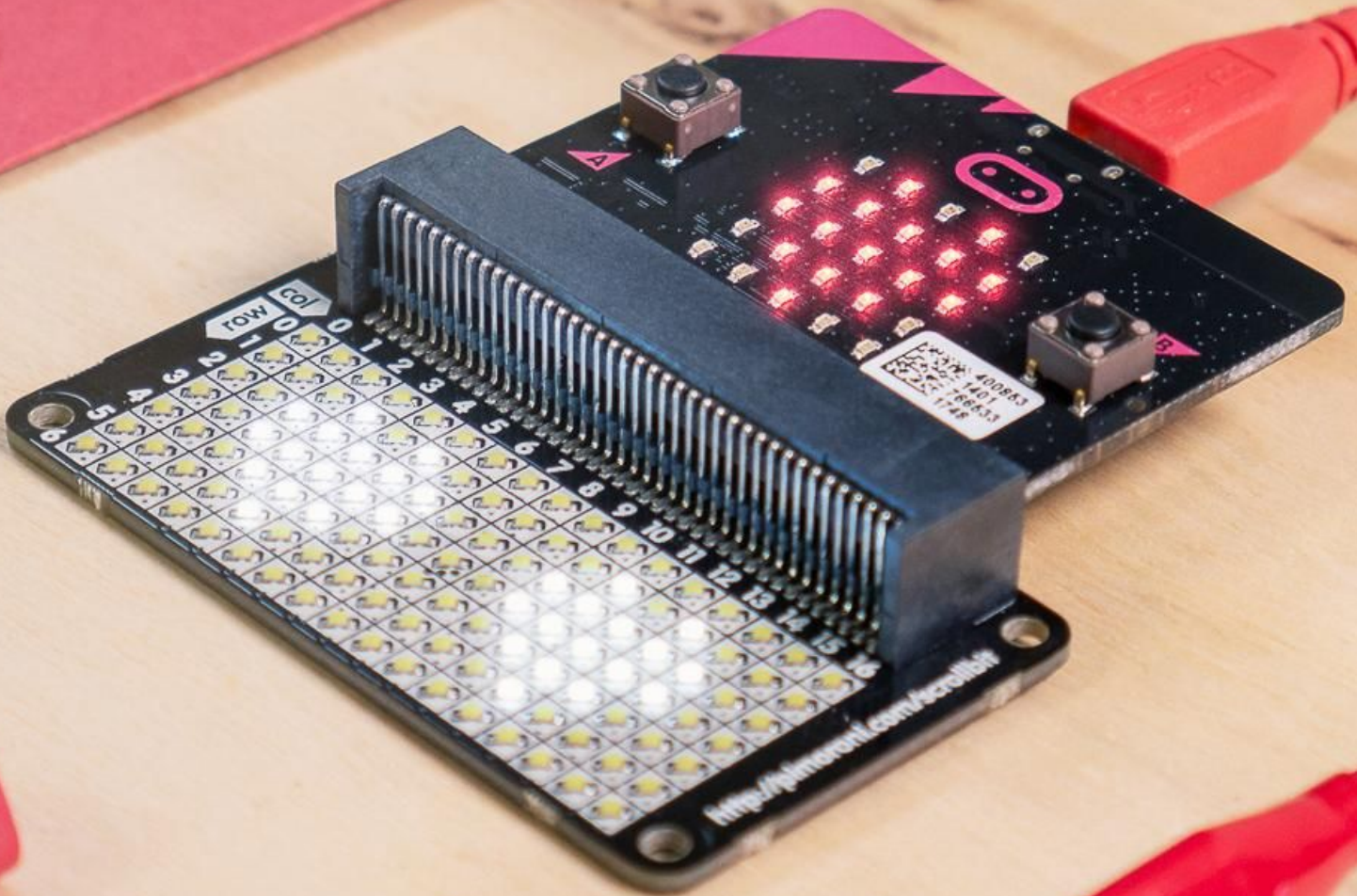
ready to go, but before you start coding you need to install the scroll:bit library if you are using MakeCode this is quite simple by clicking on the cog at the top of the window which opens a menu. From the menu click on add package and type in scroll:bit in the text field and press enter, below the text field a picture of the scroll:bit appears click on it and the scroll:bit library is now downloaded and ready to use.

To get the Scroll:Bit to work with MicroPython is a bit more complex by having to download and copy the scroll:bit.py file across to your micro:bit along with your code file. As long as you follow the getting started guide written by Pimoroni you should not run into any problems.

We tested the scroll:bit by coding our own scrolling text conference badge. This is a great little board whether you want to create a conference badge or the game Snake. You can just create Snake using



# :review



the micro:bit LED Matrix, but by adding in the Scroll:Bit you have more LEDs to play with and can make your game bigger.

Pimoroni give you a good starting point with a JavaScript reference guide on their GitHub page for MakeCode so you can get started easily. When you download the MicroPython library to use within Mu there is also some examples

within that GitHub repository to help you get started with scroll:bit. There are some really good getting started resources over on the Pimoroni Learn site too!

[go.micromag.cc/pimlearn](http://go.micromag.cc/pimlearn)

Overall this is a great micro:bit add-on if you want more LEDs to play with and make bigger games than you can make on the built-in LED matrix on the micro:bit.

## Where can I buy one?

Pimoroni Shop:  
[go.micromag.cc/scrollbit](http://go.micromag.cc/scrollbit)

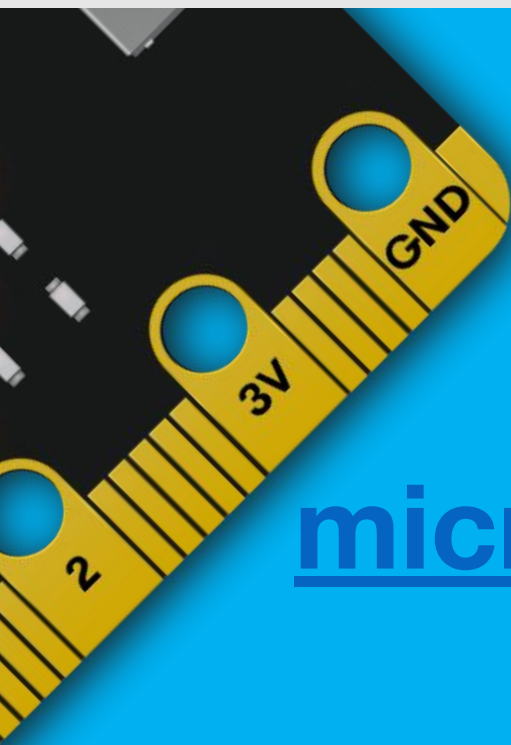
## The competition....

There are no other micro:bit add-ons like the scroll:bit!

# Help us cover the costs of micro:mag



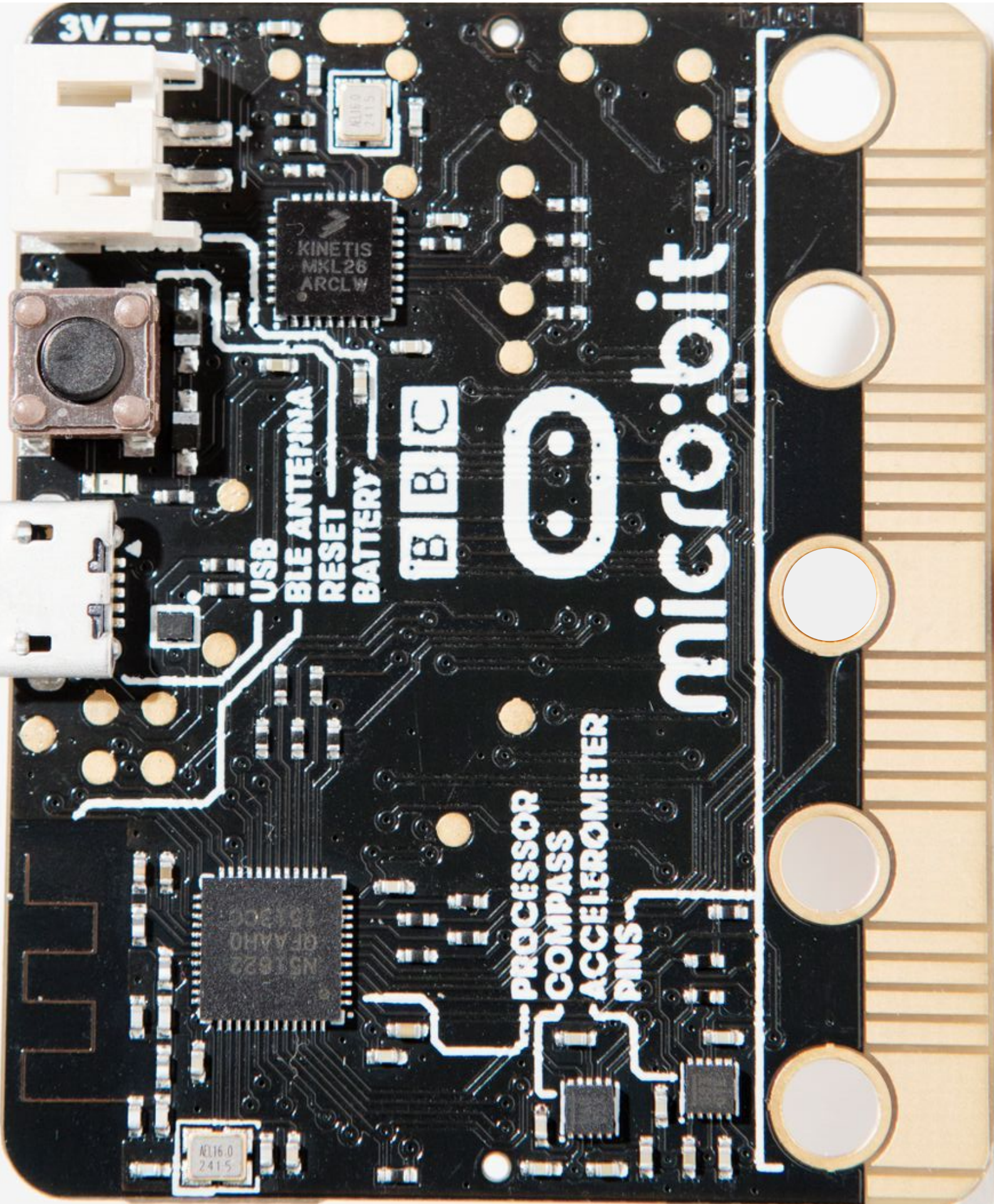
micro:mag is run by a team of dedicated volunteers - but we still need to cover our costs. Donating helps us continue to deliver the community micro:bit magazine free of charge.



Donate on  
**PayPal:**

[micromag.cc/donate](https://micromag.cc/donate)





micro:mag is released under the [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 \(CC-BY-NC-SA 4.0\)](https://creativecommons.org/licenses/by-nc-sa/4.0/) International License.

