WHAT'S INSIDE HANDWASH TIMER | MIDI WITH MICRO:BIT | HUSKYLENS REVIEW

micro :mag

Issue 8
August 2020

THE UNOFFICIAL MICRO:BIT COMMUNITY MAGAZINE

micromag.cc

Contents

News

Features

Cover Feature

Makes

Reviews

Final :Bit

DESIGN AN ADD-ON
Learn how to create your very own add-on board for the BBC micro:bit

Design an add-on | P18

Digital Tech Olympics | P10

Creating a micro:bit product | P13

Hello World

Kerry Kidd
Editor In Chief

Welcome to Issue 8 of micro:mag, we have another fun-packed issue for you including a tutorial on how to create your own micro:bit add-on boards and an update on the newly released MakeCode editor.

We also have an awesome make this month on how to make a non-touch hand washing timer which is very appropriate within the current circumstances. As usual, we also have another awesome micro:hit where Les takes us through how to make a step counter. We would also like to take the time here to thank Les for his continued support with micro:hit over the last 8 issues we are sorry to say that he is no longer going to be writing for us due to his new job at Tom's hardware taking up his time. We think the micro:hit section of micro:mag is critical to beginners so if any of you would like to step into Les's shoes and write a short tutorial for micro:hit please get in touch with us at hello@micromag.cc.

We're also really looking forward to the future of micro:mag. This will be our last PDF publication as we transition to a brand new format and we encourage you to read the last two pages of the magazine to learn more about what we have in store. That aside, we hope you enjoy Issue 8!

Meet the team

Kerry Kidd
Editor In Chief

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

Follow me:
@RaspiKidd
raspikidd.com

Joshua Lowe
Senior Editor

Josh is a young coder & creator of the Edublocks tool for micro:bit He has delivered lots of workshops & talks around the world.

Follow me:
@all_about_code
edublocks.org

Contributors

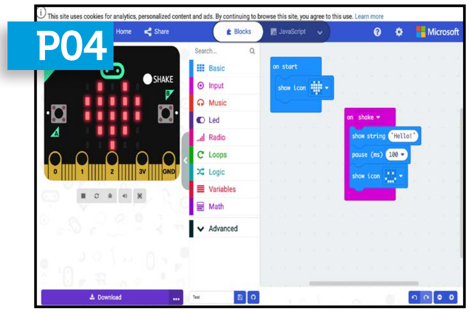
- » Les Pounder
- » Micro:bit Foundation
- » Australian Computer Academy
- » Amanda Frasure
- » Debra Ansell
- » Seluck Yusuf Arslan
- » David Perks
- » Daryl Croke
- » Rob Wilson
- » G. Michael Wood
- » Kitronik LTD
- » Xinabox
- » Nicole Parrot

Contribute!

Got a cool project, news story or make? Why not write an article for the new micro:mag website?

Sign up at:

micromag.cc/contribute



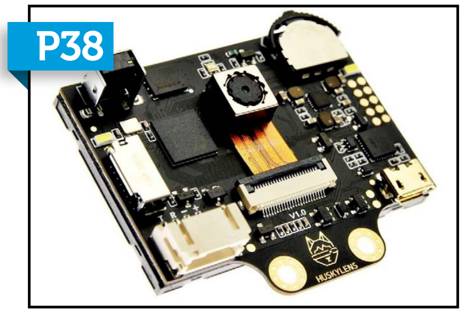
```
on start
  show icon

on shake
  show string "Hello!"
  pause (ms) 100
  show icon
```

MAKECODE 2020 UPDATE

Recap of our favourite features within the MakeCode 2020 Update.

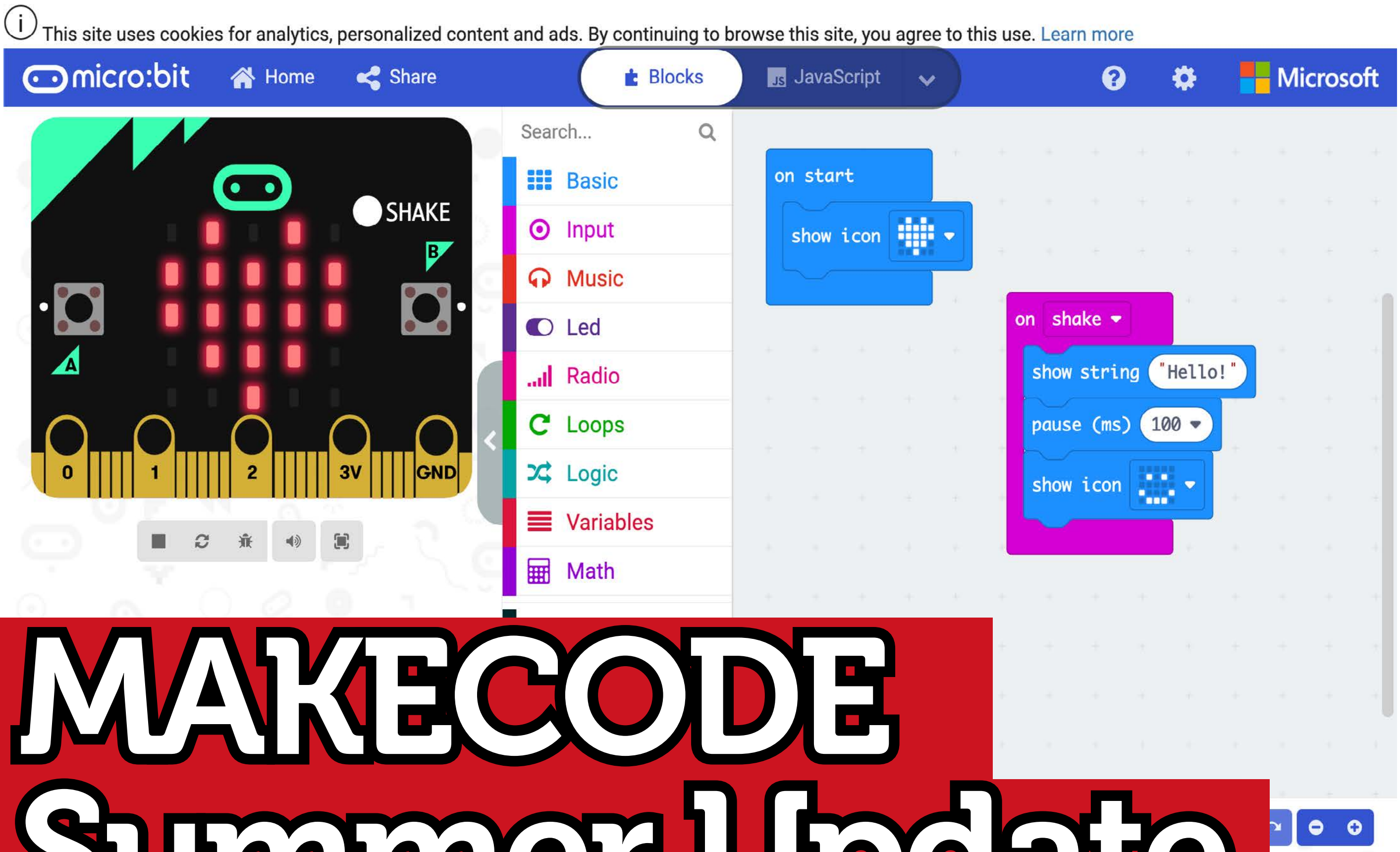
CODING WITH 5TH GRADERS

A teachers experience with using micro:bit in her classroom.



HUSKYLENS - REVIEWED

Add AI + Machine learning to your micro:bit projects with HuskyLens.



This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use. Learn more

```
on start
    show icon
on shake
    show string "Hello!"
    pause (ms) 100
    show icon
```

MAKECODE Summer Update

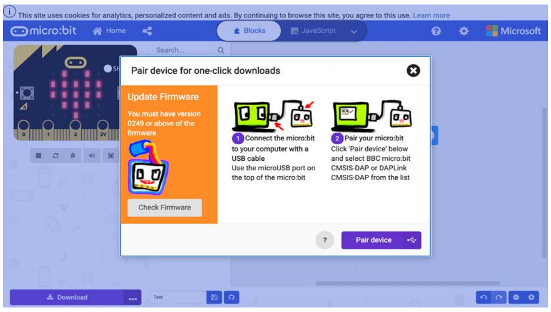
MakeCode has received its annual 2020 summer update. **Kerry Kidd** takes a look at what has been added.

About MakeCode

Microsoft's MakeCode editor is the perfect way to start programming and get creating with the BBC micro:bit. The colour-coded blocks are familiar to anyone who's previously used Scratch, and yet powerful enough to access all the features of this tiny computer. You can also switch to JavaScript and Python to see the text-based code behind the blocks.

Makecode, the most popular editor for the BBC micro:bit has now received its annual summer update. This update brings a whole host of new features to enhance what the editor can do. Let's take a look at some of our favourite features in this new release. Hopefully you can find a new feature you like and can make use of in your project!

Improved webUSB support



This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use. Learn more

Pair device for one-click downloads

Update Firmware

You must have version 0249 or above of the firmware

Check Firmware

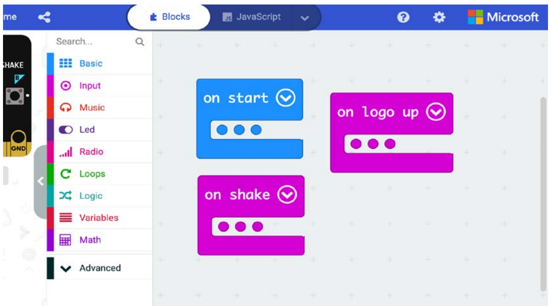
1 Connect the micro:bit to your computer with a USB cable
Use the microUSB port on the top of the micro:bit

2 Pair your micro:bit
Click 'Pair device' below and select BBC micro:bit CMSIS-DAP or DAPLink CMSIS-DAP from the list

Pair device

The 2020 update includes improved WebUSB support; this is where you can have your micro:bit connected within a chrome-based browser and download code straight to the micro:bit instead of dragging hex files from your computer to the micro:bit.

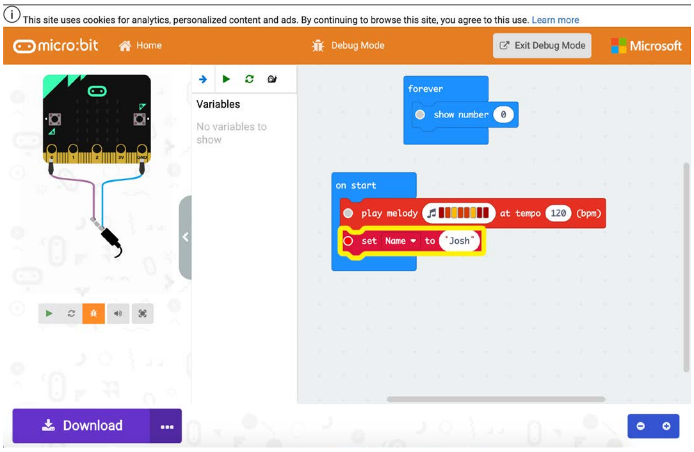
Collapsable Blocks



```
on start
on logo up
on shake
```

We can now collapse long blocks of code, making our programs more readable within MakeCode. So if you have a program with a lot of functions you can collapse the functions down to just show the name of the function, then you can expand it if you want to know more about what the function is doing.

Code Debugger



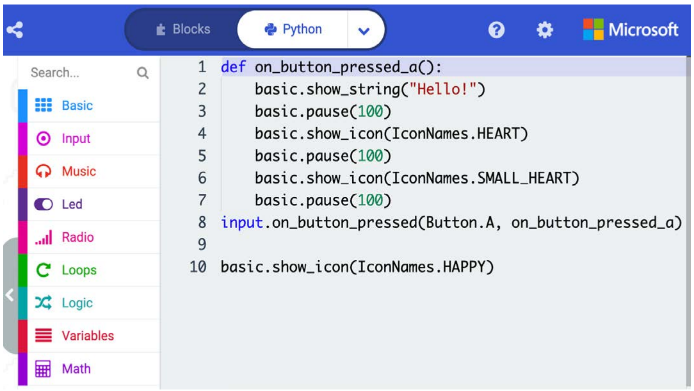
This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use. Learn more

Variables

No variables to show

Debuggers are critical tools to help students understand the state of their program execution flow and are often included in code editors. The Bug icon in the simulator lets you enter a Debugging mode, where you can step through the program looking for issues. You can add a break point, that lets you view the variables available to the program at that point.

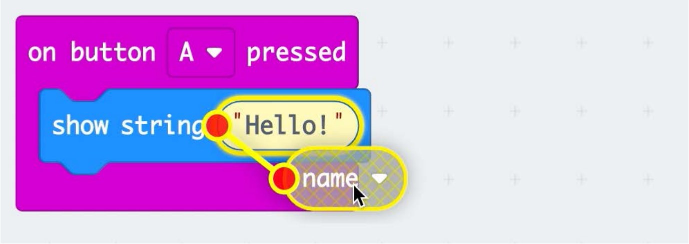
Python Language Support



```
def on_button_pressed_a():
    basic.show_string("Hello!")
    basic.pause(100)
    basic.show_icon(IconNames.HEART)
    basic.pause(100)
    basic.show_icon(IconNames.SMALL_HEART)
    basic.pause(100)
input.on_button_pressed(Button.A, on_button_pressed_a)

basic.show_icon(IconNames.HAPPY)
```

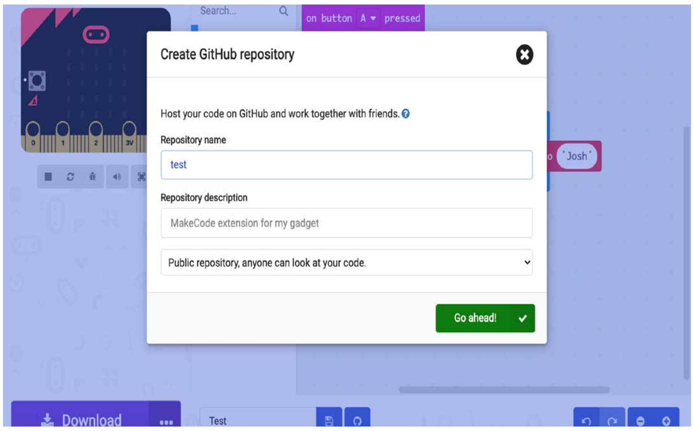
As well as being able to see the JavaScript view of your code, you can now see it in Python. This is a MakeCode variant of Python and not the same as MicroPython. So beware, any Python code you create within MakeCode will not work in any of the other Python editors.

Dot to Dot Connectors



Inline blocks now have red dots on the side and as you drag them into an input, a yellow line will appear so it's clear as to where the block is going to land. This is a really helpful feature for knowing exactly where an inline block is going to land.

GitHub Repository Support



Create GitHub repository

Host your code on GitHub and work together with friends.

Repository name: test

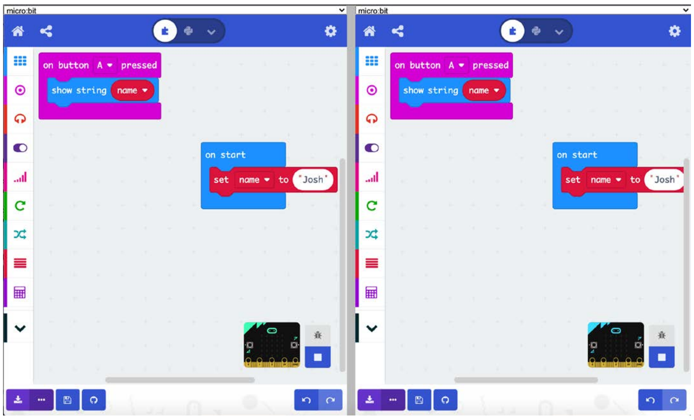
Repository description: MakeCode extension for my gadget

Public repository, anyone can look at your code.

Go ahead!

if you have a GitHub account (available for those 13 years and older), you can now store MakeCode projects in a GitHub repository. This is great for group projects, version control, and teaching students about open source software development.

MakeCode Multi Editor



The multi-editor provides a way to simulate the true experience of running radio programs on the micro:bit. The multi editor provides two separate editor windows to program your transmitter and your receiver respectively. Create, modify, and test out two programs at the same time.

Use the new MakeCode Features

MakeCode for micro:bit: makecode.microbit.org

Multi Editor: makecode.com/multi

Read about other features: go.micromag.cc/mc20

Digital Kids Camp

OpenUK in partnership with Pimoroni & Huawei have launched “Open Kids Camp” based around the micro:bit.

ABOVE: Digital Kids Camp is run by the OpenUK Organisation.

In partnership with Pimoroni and Huawei, Open UK have launched the “Open UK Digital Kids Summer Camp”, an opportunity for 3000 kids to get hands-on with micro:bit and the MiniMU Glove Kit, the kit made by Pimoroni with Helen Leigh and Imogen Heap. The camp features a course which consists of 10 animated lessons that are 10 minutes long. In each lesson, participants of the camp will take part in a range of fun activities with the MiniMU Glove Kit and the micro:bit. OpenUK provided 3000 kits to kids in the UK for free in order for the competition to go ahead.

The lessons are also free of charge for anyone to access on the OpenUK Kids Camp youtube channel, so if you have a MiniMU glove already and are wanting to take part in the activites this is a a great way to do so! Alongside the videos is an Ezine with articles from members of the OpenUK and micro:bit communities. These provide insights into how to get starting with programming and industry tips from leaders such as David Whale, Femi Owolade-Coombes and many more.

To find out more about what OpenUK are doing and stay tuned for more competitions and activities like this in the future, visit their website.

Find out more

OpenUK Kids Camp
https://openuk.uk/openkidscamp

OpenUK Kids Camp YouTube
https://go.micromag.cc/ytkidscamp

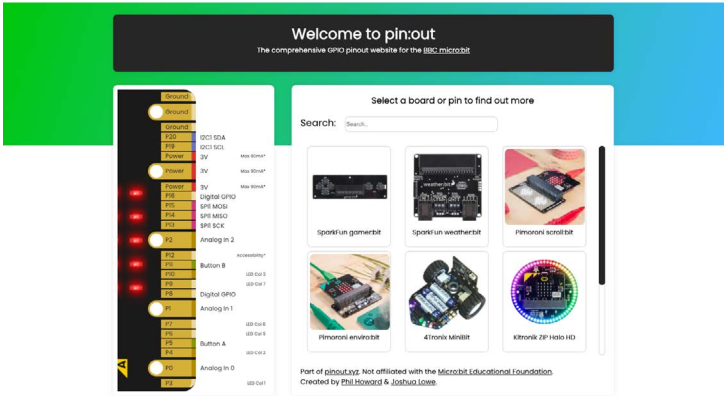
pin:out for micro:bit

Are you a manufactuer of micro:bit add-on's or just love a certain add-on for the micro:bit? Ever wanted to know which pins a micro:bit add-on board uses and had to search through pages of documentation to find what you're looking for?

Pinout.xyz, a popular website for finding out pinout information for the Raspberry Pi now has a website for the micro:bit which our own Senior Editor, Joshua Lowe, worked on. It's a great place to go and find detailed information about every pin on the micro:bit as well as pinout information for popular micro:bit accessories. Pages for every pin are now populated with information but the website is now looking for companies and community members to add in pinout information for popular add-on boards to pin:out. It's really simple to do and it only requires a few steps which are easy to follow.

It would be great to see this website take off in the same way it has for the Raspberry Pi so head on over to the pin:out website to take a look at this awesome resource!

microbit.pinout.xyz



Welcome to pin:out

The comprehensive GPIO pinout website for the BBC micro:bit

Select a board or pin to find out more

Search: Search...

Part of pinout.xyz. Not affiliated with the Micro:bit Educational Foundation.
Created by Phil Howard & Joshua Lowe.

New Accessories SHOWCASE

We take a look at some of the brand new micro:bit accessories that have just hit the shelves!

SnowPi RGB KICKSTARTER

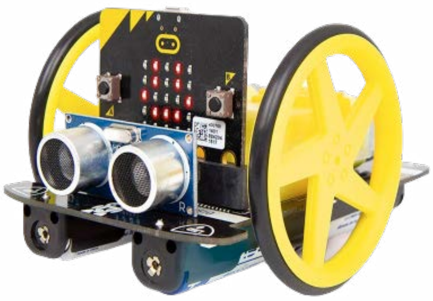
SnowPi RGB is a wintery add-on for your Raspberry Pi or micro:bit with adaptor. In the shape of a Snowman, it features 12 RGB LEDs which builds on the original SnowPi board released in 2015. It's a great way to use neopixels with the micro:bit for the first time! You can back the project now on Kickstarter.

Price: £9 **Back it on KS:** snowpi.xyz

4tronix Really Useless Box

The Really Useless Box (RUB) is a fun and quick assembly project that nevertheless allows for a lot of learning during coding. The RUB kit comprises of 7 circuit boards (PCBs) that screw together with the fixings provided. Make the RUB react to toggling the switch. Move the switch to the ON position and the Really Useless Box can switch it off again.

Price: £12 **Buy:** go.micromag.cc/RUB



Kitronik :MOVE Motor

The Kitronik :MOVE Motor for the BBC micro:bit provides a fun introduction to buggy robotics. Learn about movement, how to utilise light and sound and obsticle detection + avoidance. Attached to the chassis are two bi-directional DC motors with variable speed control. The wheels have rubber tyres and are a simple push-fit onto the motor shafts.

Price: £25.14 **Buy:** kitronik.co.uk/5683

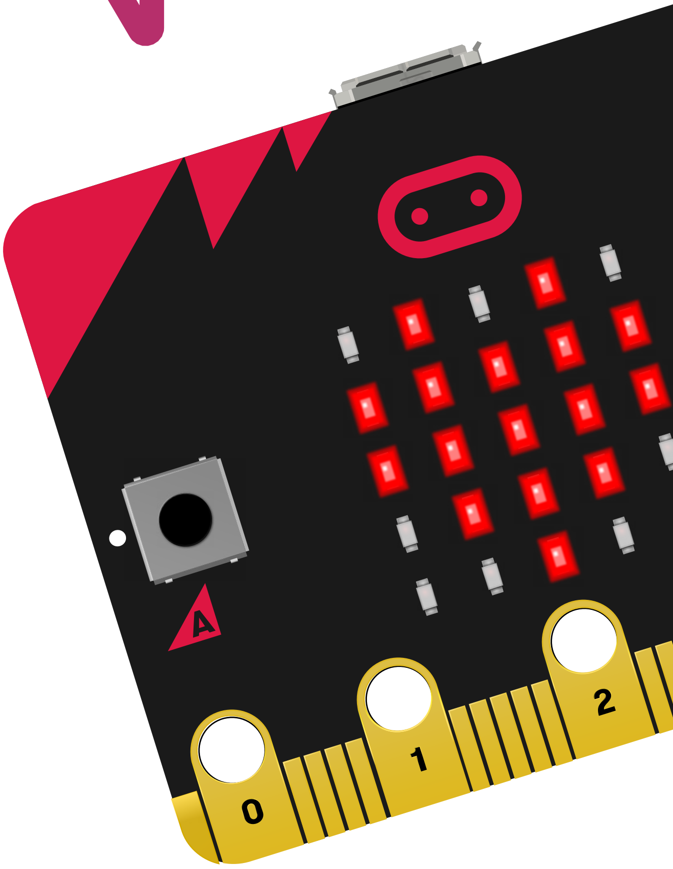
micro:bit home learning

Digital creativity for families!

To support families at home we've launched short video guides, perfect for informal learning.

Every project is built around programming and hardware concepts.

Downloadable activity sheets support young coders and give them further ideas to make the code their own.



Start here: microbit.org/homelearning

micro:bit

DIGITAL TECH OLYMPICS

The Australian Computing Academy ran a Digital Technology Olypics based on micro:bit. By **Nicola O'Brien, Naomi Cammayo & Sujatha Gunja**

About the Author

The Australian Computing Academy provides primary and secondary educators with classroom resources and training to help implement the Australian Digital Technologies Curriculum effectively.

Twitter: @AusCompAcademy

The Australian Computing Academy (ACA) partnered with the Western Australia Department of Education and Teachers Can Code (TCC) to hold a mini Digital Technologies Olympics in Perth, Australia. Over 150 teachers gathered for a unique competition as part of a two-day professional training workshop. Teachers had fun and learned valuable new digital technologies (DT) skills through hands-on learning and collaboration with their peers.

The micro:bit is a fantastic tool to raise engagement and interest towards DT given its interactive qualities, and so the ACA team challenged the teachers to work together in teams and create micro:bit projects that integrate computing with sports and other learning areas which can link back to the Health and Physical Education Curriculum. On the eve of the DT Olympics, there was a hive of planning, prototyping and designing after a hands-on workshop conducted by the ACA on algorithms and programming basics.

As the teachers pushed towards the completion of their group projects, they were able to reflect on the frustrations and challenges that their own students encounter in working with tech. All teams showed astounding levels of creativity and so much was achieved after just three short learning sessions!

Above:

ACA's academic director Dr James Curran and TCC's Bhavneet Singh takes aim at the "Cupids' Arrow" project

Among the impressive projects were:

- The “micro:bit dance challenge” which uses the radio to give the participant an easy or difficult dance sequence to follow;
- The “cupid’s arrow” that challenges participants to test their archery skills;
- The “balancing act” wherein micro:bit accelerometers are used to detect balance and stability in yoga stances;
- The “push yourself,” which involves a micro:bit-powered car in the middle of two individuals who need to do push-ups to move it towards themselves to win

You can have a taste of the DT Olympics by trying out the awesome ACA Milk Bottle Challenge.

While there was no shortage of competitiveness between teams, there was a great atmosphere of support and positivity as they presented their work and tested each other’s projects. Everyone was enthusiastic in giving and receiving valuable insights on how to design, code, test and implement solutions. Many groups also shared how their particular activity could be differentiated for specific classroom needs, thus making learning applicable to a diverse range of contexts.

See the amazing teachers in action:
youtu.be/hqkLK7wGHFk

Above:
The awesome teachers behind the “push yourself” project

CODING WITH 5TH GRADERS

Have you ever attended a conference and thought, "You could do what they are doing"? By **Amanda Frasure**

About the Author

Amanda is a 5th grade reading teacher at Central Elementary, Vinton County Local School District.

Twitter: @afrasure6

Nearly two decades into their career as an educator, Amanda has been the participant of many conferences, professional developments, and assemblies. So many times, Amanda has had that exact thought – that they could do what they were doing and that Amanda could probably do it better.

In August of 2019, Amanda submitted a proposal to be a presenter at OETC (Ohio Educational Technology Conference, in Columbus, Ohio in February 2020). The general idea was this: She would teach four 5th graders how to write code and use a micro:bit throughout the school year. They would use their knowledge to create and invent new things. Along the way, Amanda would take small video clips of them and create a video documenting their learning journey. Then, they would join her at the conference and be presenters along with her. Amanda thought that if they could show educators from all over Ohio how she taught 5th graders to code and use the micro:bit, and if her students were there to help them learn, they would be inspired to take what they learn and use it in their own classrooms and school districts.

A few weeks into the school year, Amanda selected four students: Chase, Blake, Ava, and Mady. They met every day during their specials and Amanda's plan time. This gave them about 30 minutes each day. At first, she taught the students to write basic code through makecode.

This took about six weeks. Next, Amanda began to teach them about the micro:bit and how to program it. They also began looking for ideas for what they wanted to create. Once her students were determined what they were going to create, they began meeting an extra hour after school every Monday and Wednesday.

Chase decided to create a cartoon robot. Blake chose to make a fruit piano. Ava made a Genie Ball, and Mady created a Frustration Game. Amanda took the students through the design process, built prototypes, and learned a lot through trial and error. Her students learned about patience, perseverance, problem-solving, and teamwork.

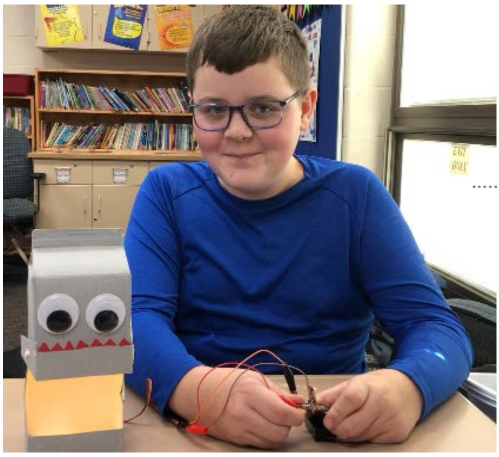
Throughout the school year, they started fundraising and began selling Candy Grams. This was an EntreEd project that taught Amanda's students the value of earning money and creating a budget for supplies and materials.

Left:

Mady French: Frustration game

See, they had become Central's Coding Crew and they decided that they needed t-shirts for OETC. They also decided that to help the participants of their session, they wanted to be able to take crates of materials for them to build and create something new. In addition to that, Amanda felt the only true way that would allow other educators to take what they had learned and begun implementing it immediately would be to give away micro:bits at the conclusion of our session. Along with the Candy Grams, they did a whole-school fundraiser where they had a building goal dollar amount and if they met the goal, a colleague of Amanda's (and a chaperone for OETC) was going to Kiss a Pig – they earned over $1,000! They had enough money to purchase t-shirts, fill eight crates with miscellaneous materials for new inventions, and then purchase 25 micro:bits to give to session participants. (In addition to these funds, Central Elementary's general fund paid for our hotel stay and food.)

At the beginning of February, their new creations were done, their video was completed, crates were assembled and ready to be used. Amanda held a general assembly at school where they showed all the students (Preschool-5th Grade) and teachers what they had been working on during the school year and what their money from Candy Grams had been for. They left on Monday, February 10, 2020, after school. The first day of the conference Amanda scheduled a visit to the Convention Center. There were hundreds of booths that showcased all sorts of technology. Some of their favourites were seeing a Mars Rover, and getting to meet BB8 (en entirely 3D printed working robot), and meeting and operating R2D2. Since they wore their t-shirts, they also had a lot of conversation about their upcoming session on February 12th.



Left:

Chase Rose: Carton robot

The day of their presentation, they were up and ready early. Amanda's students had programmed a micro:bit with their name continuously scrolling across the LEDs, an image to appear when the A button was pressed, and a tune to play when the B button was pressed. These were assembled into name badge displays that were positioned in front of her students at their table. Slowly, as it neared their session time, people started to arrive. After a brief introduction of who they were, where they were from, and what their session was about, Amanda played the video documenting the student's learning journey. Afterwards, participants began to program the micro:bits and use materials to make their own inventions and her students were able to circulate around helping with questions, showing how to program and download code to the micro:bit, and even showing how to attach speakers for audio or motors for movement. The session concluded with 25 people winning a micro:bit to use in their own classrooms. Amanda is certain that the students and herself accomplished their mission and inspired other educators!

Below:

Presenting at OETC

CREATING A MICRO:BIT PRODUCT

Debra Ansell shares her path to creating a micro:bit wearable product.

About the Author

Debra Ansell blogs and tweets as GeekMomProjects online. Her belief that LEDs improve everything inspires much of her work.

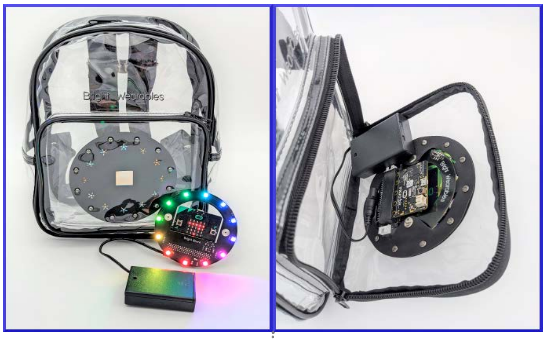
Twitter:

@BrightWearables

Debra credits micro:bit with merging two of her disparate passions, LED wearables and coding education, into her new tech wearable business, BrightWearables.com. For many years, she had enjoyed volunteering to introduce elementary school children to introductory coding concepts. Her formerly distinct hobby/obsession of integrating LEDs into clothing and accessories kept her active in the Maker community. Some of her Maker and instructional activities are documented on her blog at GeekMomProjects.com.

Debra's volunteer work introduced her to the micro:bit almost four years ago while she was researching educational physical computing platforms. After evaluating the versatile, powerful, but easy to code board, she was hooked. She had planned to use it in coding lessons, but feels it was inevitable that she would eventually find a way to incorporate it into her LED projects.

Intrigued by the potential synergy of tech education and LED wearables, Debra set out to create a fashion accessory that is easily programmed to change LED colours and patterns but also durable enough to withstand the stresses of being worn.



Above:

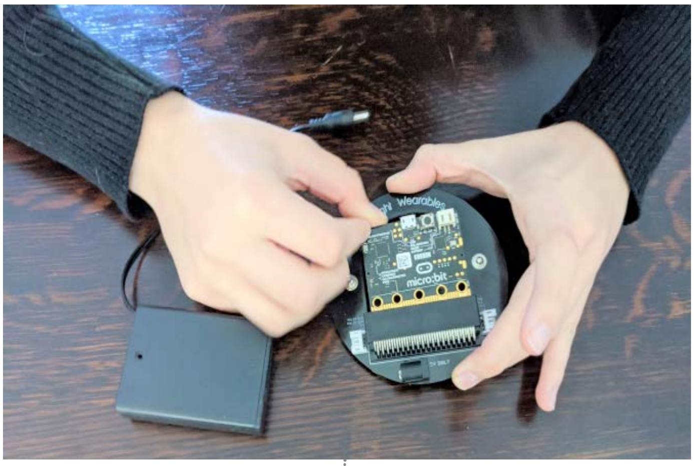
Bright Wearables is a modular system of PCBs and specifically designed bags.

Two years, dozens of prototypes and one US patent filing later, she launched her new product line, Bright Wearables, in December of 2019. The first product offering is a modular system consisting of different designs of illuminated bags and backpacks (Bright Bags) designed to enclose and display a micro:bit accessory board (Bright Board) showcasing a ring of addressable RGB LEDs. Debra explained her goals in bringing programmable fashion accessories to market this way, "While Bright Wearables bags are a fun and attractive fashion accessory for anyone, my hope is that the Bright Wearables line will attract students who might not be engaged by more traditional physical computing environments, such as electronics kits and robotics, to coding."

Debra describes her journey of developing a new micro:bit-based product as both challenging and fascinating. Without a roadmap for the process, she relied on micro:bit's extensive educational ecosystem and hardware accessory market to provide her with direction along the way.

Developing the Bright Board PCB

Debra had electronics experience but had never designed a PCB, so she initially hired a consultant to turn her hand-soldered prototype into a manufacturable design. The Bright Board design had to satisfy several constraints. It required a slim profile, a connector to position the micro:bit with its LED grid centered inside a ring of RGB LEDs, and a system of integrated mechanical fasteners to attach the board to the Bright Bags. As with any wearable, all electrical connections needed to be sturdy and safe, and the high current draw of RGB LEDs made heat dissipation essential.



Above:

Though more costly to manufacture, the 80-pin micro:bitconnector makes it easy to connect the micro:bit to the BrightBoard

The slow turnaround with the consultant became frustrating, so Debra began to learn Eagle CAD herself. It was daunting, and she describes herself as "terrified" to submit her first order to a PCB manufacturer. She still remembers the reassurance she felt when an experienced maker informed her that even his PCB orders sometimes turned out useful only as "coasters." That encouragement, plus the low cost of ordering custom PCBs helped her to gain confidence in the process, experiment with design variations in different prototypes, and learn through a process of trial and error.

When Debra felt her prototype was ready for production, she hired an electrical engineer through Upwork to perform a design review and help her navigate the unfamiliar PCBA process. The engineer suggested several useful modifications and provided a second set of eyes at a critical step. She extols the benefits of that decision as follows, "Sending a product to be manufactured is nerve-wracking, and it is crucial to do whatever you can to reduce the possibility of production errors."

She found the most difficult design decision for the Bright Board PCB to be selecting the connector to attach the micro:bit. The standard 80-pin through-hole edge connector is a relatively expensive component and its through-hole mounting mechanism adds cost to the PCB assembly process. Debra experimented with other connection methods including pressure-based spring connectors (not robust enough) and screw connectors (too inconvenient for the end-user) before deciding that the sturdiness and benefit of allowing the end-user to easily pop the micro:bit in and out of the Bright Board made it worth the extra cost. Her hope is that Bright Wearables owners will find it handy carry a micro:bit with them, which they be able to pop out of the Bright Board to use in projects anywhere they go, and then snap it back in place to illuminate their Bright Bag when finished.

Above:

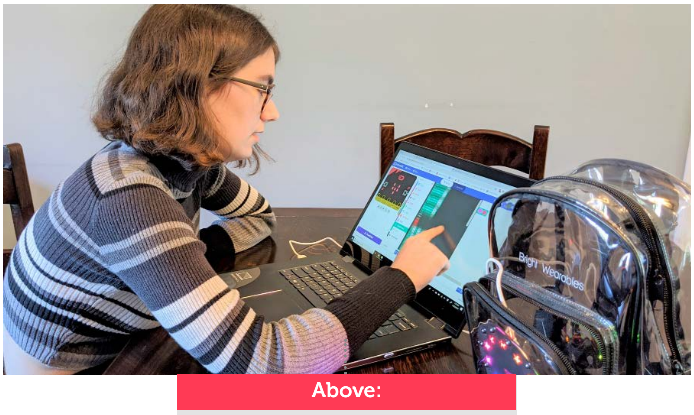
The 80 pin edge connector which is common on many add-on boards.

Coding a MakeCode Bright Board Extension

When she started Bright Wearables, Debra felt far more comfortable working with software than hardware. She quickly realized that micro:bit's coding website provided an existing, easy-to-use programming interface for the Bright Board, which would save her a great deal of time and effort. When writing the Bright Board library extension for MakeCode, she was delighted to discover that the development environment was clearly and extensively documented and well designed for extensibility.

As Debra says, “It is straightforward for anyone, not just professional developers to access and use the MakeCode extension developer tools. The integration of the MakeCode source and contributed extensions on GitHub gave me numerous practical examples for models.”

MakeCode has a number of useful tools and support channels for extension developers, including clear and extensive tutorials on designing code blocks, a MakeCode “playground” website in which to test them, and a forum for discussions and feedback on the micro:bit Slack channel. Debra feels that writing the Bright Board MakeCode extension was the only part of the development process for which she had some sort of roadmap, and she is grateful to the talented MakeCode creators and developers who provided it. She describes herself as having been “absolutely thrilled” to learn that her extension was officially approved and that it was searchable within the official MakeCode extensions list.



Above:

The extensibility of the MakeCode editor made it an ideal choice for the BrightBoard coding interface.

The Path Ahead

In starting Bright Wearables, Debra learned that the process of developing a wearable tech product extends far beyond just electronics and software. She had not anticipated the amount of research required simply to navigate United States business, trademark, and patent laws, and she claims that the story of sourcing Bright Bag production from Chinese manufacturers could fill its own lengthy article. In researching the market for educational wearables, Debra solicited feedback from informal focus groups of friends, their friends, and their friends’ friends. When setting up sales, she found that, despite her experience administrating her WordPress blog, building a WordPress-based shopping site is another animal entirely.

Debra opened sales on her website at the very end of 2019 and discovered that introducing the public to a novel product niche requires clear, concise, and effective explanation. While micro:bit is well known in the UK, it is much less familiar to the United States. According to Debra, the feedback from those who see the bags in person has been very good, but she is still working on the best way to make the concept of “codable wearables” clear online, and would like to add more documentation to the website.

As Bright Wearables enters its newest phase, Debra finds herself, once again, without a roadmap. There will undoubtedly be many twists and turns in the expedition to come, but she has enjoyed the adventure so far, and feels optimistic about the journey ahead.

Below:

The BrightWearables system lets its users take their micro:bit projects on the go

THE RISE of micro:bit in Turkey

Selcuk tells us how the micro:bit is playing a key part in education in Turkey

About the Author

Selcuk Yusuf Arslan is a computer science teacher in Ankara, Turkey. In 2019, he was selected as the Most Outstanding Young Person of the World Top 20 Honoree by JCI in the category of Humanitarian and Voluntary Leadership.

Twitter: @selcukyarslan

More Coding More Girls

The project More Coding More Girls that Selcuk has carried out at Ataturk Vocational and Technical High School made an impact both in Turkey and in Europe. The project, which was first awarded by STEM Alliance at Diversity in STEM Competition, was presented at micro:bit Live 2019 in Manchester and at the BETT Show in London last January. In addition, the project was reported as news both on the Ministry of Education of Turkey's website and national press. The most important feature of this project was that it enabled him to meet micro:bit. After this project, he continued to do a series of work with the micro:bit.

MCMG provided micro:bit education under the project of Code for Nature to 20 students who have not taken any education before and this was funded by Turkey's leading airline company Pegasus and the Toplum Gönüllüleri Foundation(TOG). These 9-10 years old students were studying in a disadvantaged school and they received coding education for the first time. It was really efficient to start coding with micro:bit and also the easy programming of the micro:bit, integrated sensors and good price/performance were our prefered reasons for this project.

Teacher Trainings

As a teacher, Selcuk often organises workshops. One of these was micro:bit workshop that was held in Turkey's southern Hatay at the National Education Exchange Forum. All teachers were given a certificate of participation at the end of the workshop, where approximately 15 teachers participated. Selcuk opened another teacher workshop for Altındağ Science and Art Center teachers. In this 2-hours workshop, he explained the basic practices to the teachers. Selcuk continued these studies during the epidemic. Could a microbit workshop be organized with distance education? Why not! He tried to teach micro:bit by making a YouTube live broadcast for teachers from the city of Mugla which was located the west of Turkey. More than 1000 teachers joined this session. The high participation rate made him very happy. Selcuk is also organized another workshop at the National Interdisciplinary Science Education Teachers Conference on the 4th-5th of July in 2020. Also, he continues to tell people about the projects that he made with the micro:bit to the teachers and candidate teachers in the experience sharing presentations in Turkey.

Rise of micro:bit

Doku Schools also use the micro:bit, which is the other school Selcuk teaches courses at and his students are extremely happy to code with micro:bit. The feedback he received from the students shows that the micro:bit is an extremely entertaining microcomputer. After the work that Selcuk did with the teachers, he voluntarily provided micro:bit training for 100 students in 5 sessions with his colleague Merve Özer in a shopping mall that located in the capital city of Ankara. Finally, Selcuk can not pass without mentioning that he has started a distance micro:bit education with Turkey's leading distance education company Novarge. In this way, micro:bit's ecosystem will grow more in Turkey. He also sees that more teachers and students use micro:bits after these sessions. Selcuk tried to be a pioneer about the usage of micro:bit in Turkey and this movement continues to grow fast.

Above:
Selcuk delivering a micro:bit workshop in Turkey.

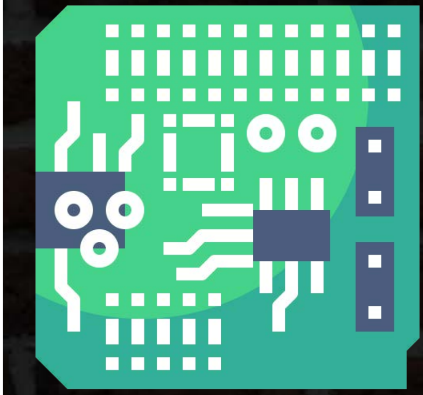
DESIGN AN ADD-ON

Introduction

One of the awesome features we outlined about the micro:bit in our last issue was the edge connector at the bottom. This is great for providing expandability to your micro:bit and increasing the number of things you can do with it.

The market is full of interesting add-on boards you can connect to your micro:bit to enable it to do a whole host of different things, but sometimes, there isn't an add-on board for everything. So why not make your own?

In this issue, we'll be taking a look at how we can make our very own micro:bit add-on board using a free piece of software called Fritzing and how to take it to the manufacturing stage in a few simple and easy to follow steps. Let's get started!



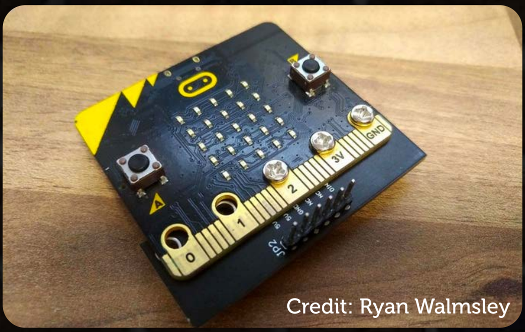
PCB Idea

First off, we need to think of an idea for our add-on board. If this is your first time creating a PCB, it's good to think of something simple.

Many micro:bit add-on boards make use of the full edge connector by using an edge connector socket on the add-on. However, to make things simple, we'll just be using the 5 main pins of the micro:bit and attaching the micro:bit to our PCB via screws and standoffs.

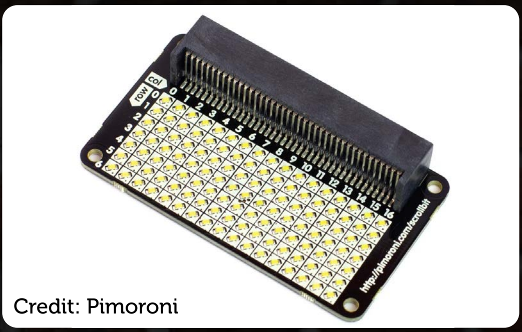
The PCB we will be designing is a simple traffic light system with 3 LEDs, however, we encourage you to follow what you've learnt in this tutorial and apply it to an idea of your own.

5 Pin Board



Credit: Ryan Walmsley

Edge Connector



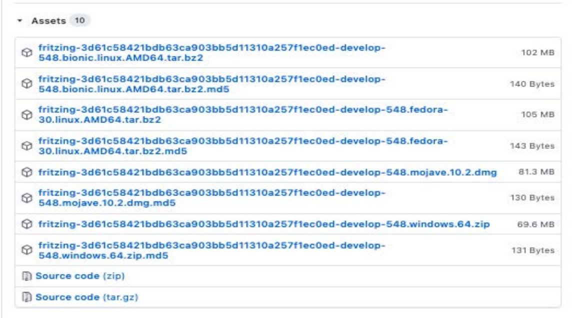
Credit: Pimoroni

Install Fritzing

The software we are going to be using to create our PCB is Fritzing. You may have used it before for creating circuit diagrams. It's really simple to use and requires little to no learning at all. It's important to note that Fritzing is not the "industry standard" for creating PCBs, but it's a great place to get started. If you're looking for something more advanced, you should look at alternatives like KiCad and Eagle.

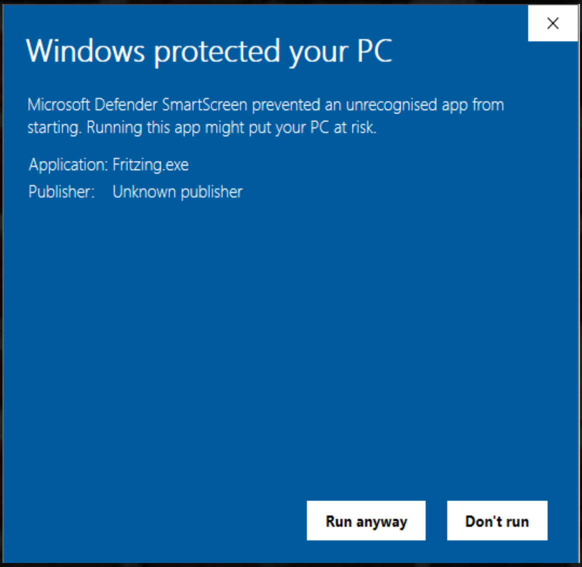
Fritzing is an open-source project maintained by a group of community members. Fritzing can be downloaded via their website with a donation of £8 and if you can donate, we encourage you to do so, especially if you're a regular user of the software. However, the Fritzing team release a free version via GitHub which we'll show you how to download.

Head on over to https://github.com/fritzing/fritzing-app/releases/tag/CD-548 where you'll find the latest release of the software. At the bottom of the page you'll see a range of download links. The names are quite confusing but to find the right one look towards the end of the link where you'll see the name of your operating system.



▾ Assets 10

fritzing-3d61c58421bdb63ca903bb5d11310a257f1ec0ed-develop-548.bionic.linux.AMD64.tar.bz2	102 MB
fritzing-3d61c58421bdb63ca903bb5d11310a257f1ec0ed-develop-548.bionic.linux.AMD64.tar.bz2.md5	140 Bytes
fritzing-3d61c58421bdb63ca903bb5d11310a257f1ec0ed-develop-548.fedora-30.linux.AMD64.tar.bz2	105 MB
fritzing-3d61c58421bdb63ca903bb5d11310a257f1ec0ed-develop-548.fedora-30.linux.AMD64.tar.bz2.md5	143 Bytes
fritzing-3d61c58421bdb63ca903bb5d11310a257f1ec0ed-develop-548.mojave.10.2.dmg	81.3 MB
fritzing-3d61c58421bdb63ca903bb5d11310a257f1ec0ed-develop-548.mojave.10.2.dmg.md5	130 Bytes
fritzing-3d61c58421bdb63ca903bb5d11310a257f1ec0ed-develop-548.windows.64.zip	69.6 MB
fritzing-3d61c58421bdb63ca903bb5d11310a257f1ec0ed-develop-548.windows.64.zip.md5	131 Bytes
Source code (zip)	
Source code (tar.gz)	



Windows protected your PC

Microsoft Defender SmartScreen prevented an unrecognised app from starting. Running this app might put your PC at risk.

Application: Fritzing.exe
Publisher: Unknown publisher

If you're on Windows, open the zip file and extract it. This might take a few minutes depending on the speed of your computer. Once extracted, search for the file "Fritzing.exe" and double click to open it.

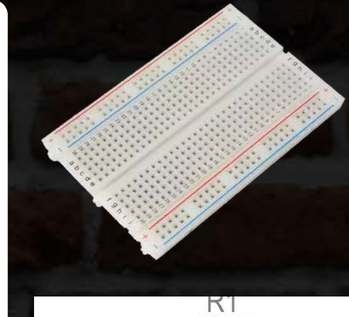
You may be presented with a warning from Windows saying that "Microsoft Defender prevented an unrecognised app from starting". We can ignore this warning by pressing the "More Info" link which will bring up a "Run anyway" button at the bottom of the dialog.

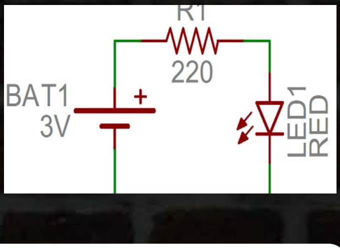
Once you've selected that, Fritzing will load up and you'll be presented with the welcome screen.

Using Fritzing

There are a few things that Fritzing can do. The first is to create basic graphical breadboard diagrams. This can be really handy for writing tutorials for micro:bit, Raspberry Pi and Arduino where you want to show how to put a circuit together. It can also be used for planning out the circuit for a PCB in a simple and graphical way. The next feature of Fritzing is schematics, these are similar to the graphical breadboard diagrams except a schematic drawing shows how a circuit is made through circuit symbols. The last feature of fritzing is the PCB designer, this is where you can drag and drop components onto a board to create a Printed Circuit Board. This can then be exported and set off for manufacture.

In this tutorial, we'll start by planning our circuit in the breadboard view, building the circuit in real life to test it out and then design a PCB.





Before we get started, we need to download the Add-On Board Template file that we have created. Download it at go.micromag.cc/template and open it in Fritzing by going to File > Open.

You'll also need the micro:bit part which can be downloaded at go.micromag.cc/mbf.

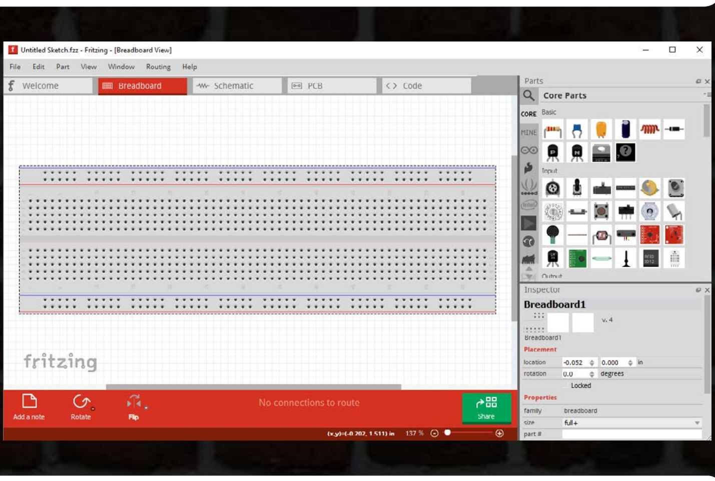
Let's start by loading up the breadboard view. We only need to create a rough diagram here in order to be able to test our circuit. There are a few things we need to know about the breadboard view in order to start using it.

The breadboard view is split up into 3 sections:

Designer

Parts

Inspector



Untitled Sketch.fzz - Fritzing - [Breadboard View]

Breadboard1

v. 4

Breadboard1

Placement

location -0.052 0.000 in

rotation 0.0 degrees

Locked

Properties

family breadboard

size full+

part #

No connections to route

(x,y)=(-0.202, 1.511) in 137 %

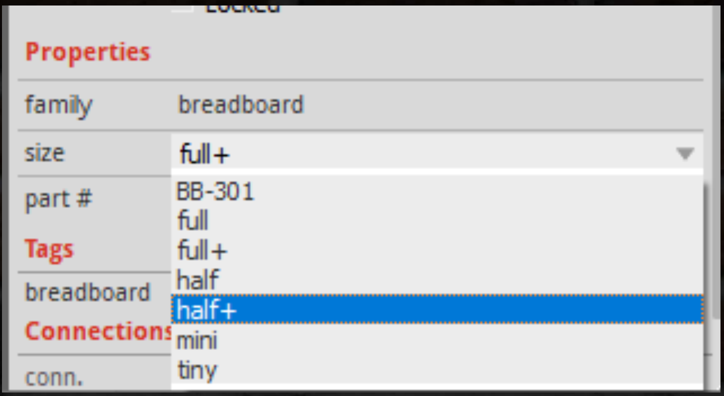
The designer section currently has a graphic of a breadboard, this is where we will place our components to create the circuit diagram, just like you would if you were building a circuit in real life. The parts section at the top right has a library of different parts in which we can drag into the designer section to create a circuit diagram. Finally, the inspector view in the bottom right has the properties of each part. If you click on a part within the designer view you can change properties like rotation and colour here.

Start by going to File and then Open in the top menu and navigate to where you saved the micro:bit part earlier on in this tutorial. Open up the .fzpz file to add it to your parts library.

Prototype

Now we're ready to start creating our circuit diagram!

You'll notice that there is already a breadboard in the designer view, however, it's a bit too big for our simple circuit. To make it smaller, we need to click on the breadboard, then in the inspector view under properties, you'll see a "Size" dropdown. It will currently be set to Full+. Click on this dropdown and change it to "Half+".


Properties

family breadboard

size full+

part #

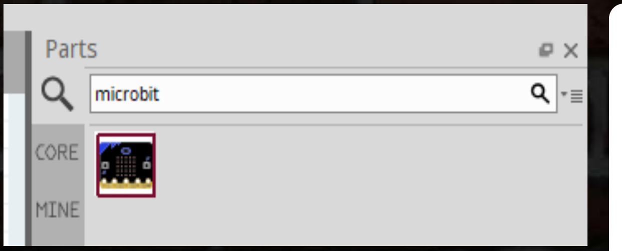
Tags

breadboard

Connections

conn.

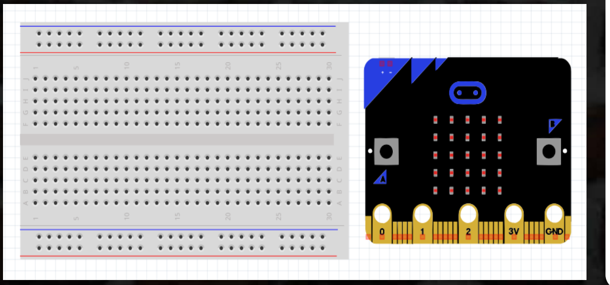
BB-301
full
full+
half
half+
mini
tiny


Parts

microbit

CORE

MINE

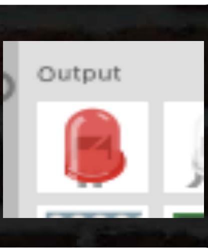


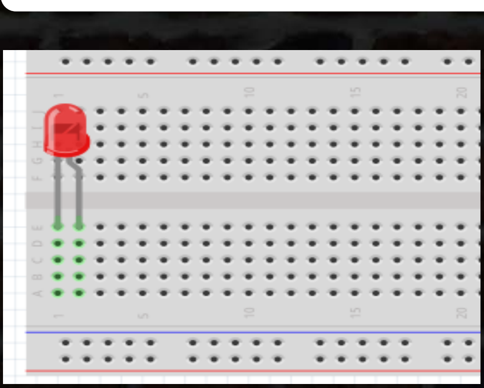
Now it's time to start adding some components! Components can be added to the designer view through the Parts section of Fritzing. You'll see that the parts view is split into 2 sections. The left section has a list of categories and the right section has a list of parts under that category.

The first part we need will of course be a micro:bit. Click the search icon in the parts view and Search for "micro:bit" in the search bar. After that, you'll see the micro:bit part. To add it to the designer view, simply drag and drop the micro:bit thumbnail next to the breadboard.

Your designer view should now look like the one on the left.

Our final Circuit will consist of 3 LEDs and 3 resistors. Let's get started with adding an LED onto the breadboard. Go to the CORE category and scroll down to "Output". You'll see a red LED icon that looks like the one on the right



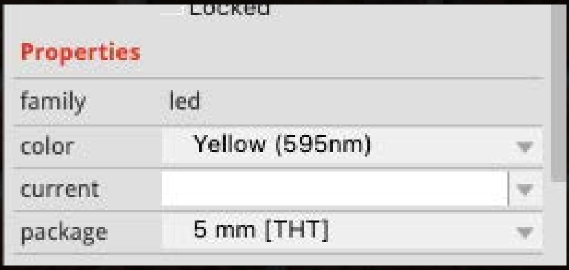


Our breadboard is split into two halves, each half has 5 rows. You'll see that there are "coordinates" on the breadboard to mark the holes. Drag the Red LED from the parts view into the top two holes of the bottom half like the image on the left. There we go! You've just dragged your first component onto the breadboard. Now, right click on the red LED and click dupilcate. Skip two holes and on the same row drop the duplicated LED.

Prototype

The circuit we are aiming to create is a traffic light so we'll need to change the colour of this second LED. To do that we can use the inspector view in the bottom right. Click on the second LED and locate "Color" under properties in the inspector view. You'll see that it will currently be selected to "Red (633nm)". We want to change this to "Yellow (295nm)". To do this, click on the colour dropdown menu and select a new colour. You'll see that the colour of the LED in the designer view has changed to yellow.

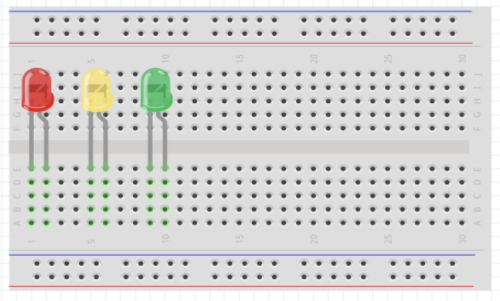
Next, duplicate and drag another LED, again missing two holes and placing it to the right of the yellow one and change the colour of this LED via the inspector view to "Green (555nm)". Our breadboard should now look like the one on the right.



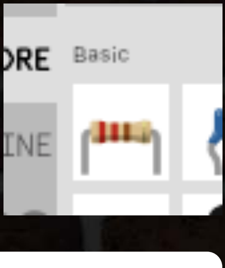
Locked

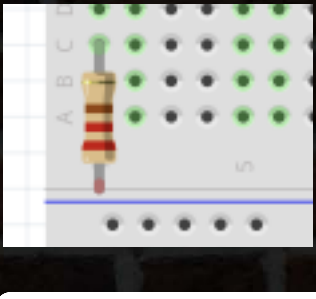
Properties

family	led
color	Yellow (595nm)
current	
package	5 mm [THT]



There we have it, our LEDs are now in place. The next thing we need to do is add some resistors. Scroll to the top of the CORE category and you'll find a resistor icon. Drag and drop it underneath the breadboard.



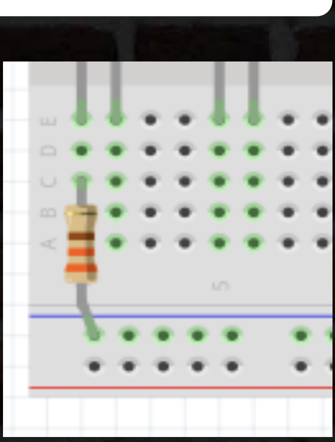


Next, what we need to do is rotate the resistor so that it's vertical rather than horizontal. Select the resistor and at the bottom of the Fritzing window, you'll see a rotate button. Click it once to rotate the resistor so that it's vertical. Now once you've done that, drag the resistor so that the top leg is in the second hole underneath the short leg of the Red LED

The resistor value is currently 220 Ohms but we want to use a 330 Ohm resistor. To change the value, select the resistor and find the "resistance" dropdown under properties within the Inspector view. Click it and select 330 Ohms.

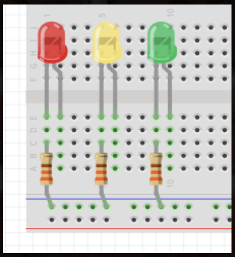
We want the bottom leg of the resistor to go into the bottom blue ground rail. To do this we will need to extend the leg. This is fairly simple to do, simply hover your mouse over the bottom of the resistor leg and drag the dot into the first hole of the blue ground rail to extend it.

Now duplicate the resistor twice by right-clicking on it and pressing duplicate. Place the two resistors in the same position for the yellow and green LEDs as you did for the red one.

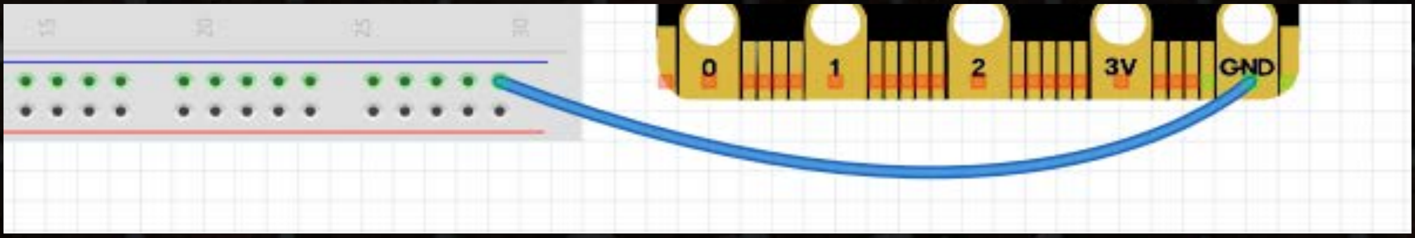


Prototype

That's all of our LEDs placed on the breadboard alongside 3 330 Ohm Resistors. Fritzing makes it really easy to plan out our circuits before we prototype them, which is why we're designing our PCB circuit on the Breadboard designer first. It's good practice to always carefully plan out PCB designs and test them to make sure the PCB will work when manufactured.



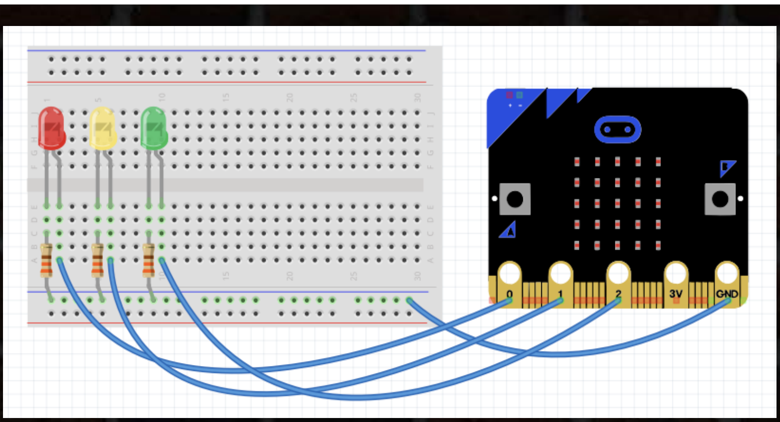
That's all of our components placed on the breadboard, good job! Now to connect them to our micro:bit. Adding jumper wires to Fritzing is simple to do. Let's create our first wire for the blue ground rail. We will be connecting the last hole to the right of the blue ground rail to the GND pin on our micro:bit. Click on the last hole and drag your mouse over to the small orange dot of the GND pin on the micro:bit. You'll notice that the wire is now covering up the other pins. We can bend the wire by holding down CTRL on our keyboard and dragging the middle of the wire down.



Now we need to connect up the LEDs. Start with the red LED, connect a wire from the last hole underneath the longer leg of the Red LED and connect it to Pin 0. Bend the wire like before so that the diagram is tidy. Now do the same for the other two LEDs, Yellow to Pin 1 and Green to Pin 2. This will complete our circuit diagram! It should look exactly like the one below:

It's time to prototype our circuit! Grab 4x Male to Male Jumper wires, 1x Red LED, 1x Yellow LED, 1x Green LED, 3x 330 Ohm Resistors, 1x Breadboard, 4x Crocodile Leads and a micro:bit. For prototyping, you'll need to connect a crocodile lead to one of the ends of each jumper wire to connect it to the micro:bit's edge connector.

Download the test code at go.micromag.cc/testcode to test out your circuit.

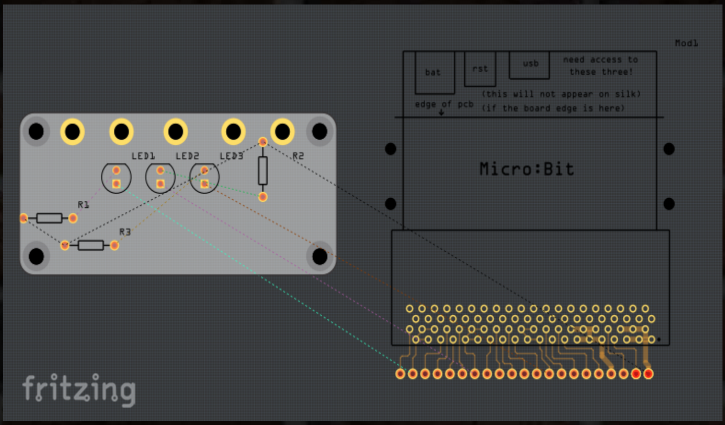


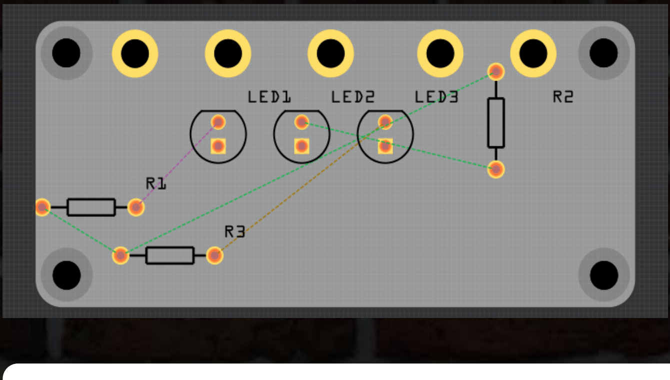
PCB Design

Now it's time to actually create our PCB. To do this, we need to click on the PCB tab at the top of the Fritzing window.

You should now see all the components from the breadboard tab in this new PCB tab. You'll also be able to see the PCB board with 5 screw holes for the micro:bit.

The first thing we need to do is remove the micro:bit from the PCB view. To do this, simply click the micro:bit text and hit delete on your keyboard.


Mod1
bat
rst
usb
need access to these three!
(this will not appear on silk)
(if the board edge is here)
edge of pcb
Micro:Bit
LED1
LED2
LED3
R1
R2
R3
fritzing



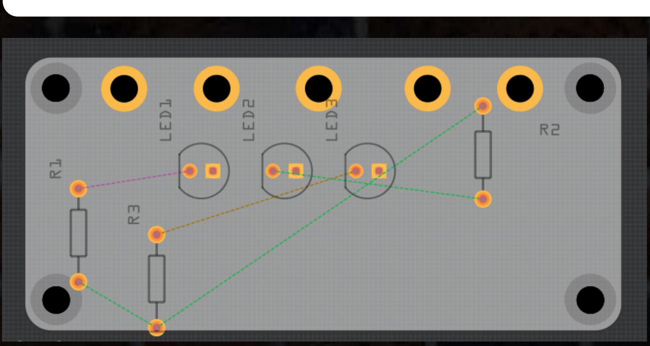
There are a few important things to note about PCBs before we start the layout of ours. The PCB we will be creating is a "Two-Layer Board" which means it has two sides in which you can have traces. It's good practice to put all the traces on the back of the PCB if possible so it looks nicer when produced. PCBs also have a layer called "Silkscreen" in which you can put text and images. They also have a "Soldermask" layer which is the colour of the PCB.

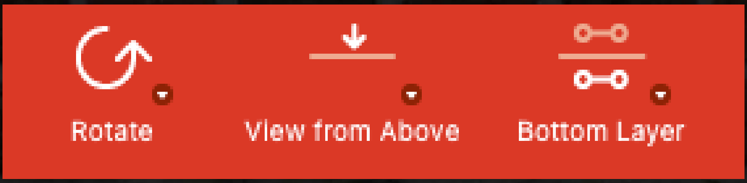
Let's create the layout for our PCB. The first thing we want to do is ensure we are editing on the "Bottom Layer". To do this, click the "Layer" button in the bottom toolbar until it says "Bottom Layer".

Now we need to ensure the components are the correct orientation. We want the flat side of each LED (The circular components) to be towards the left. As we did in the breadboard view to change the rotation. Select the component and until it's the correct orientation. Do this for each LED.

We also want to make sure that the resistors are the correct orientation. Make sure that they are all vertical. You can do this by pressing the same rotate button.

Your PCB should now look like the image below

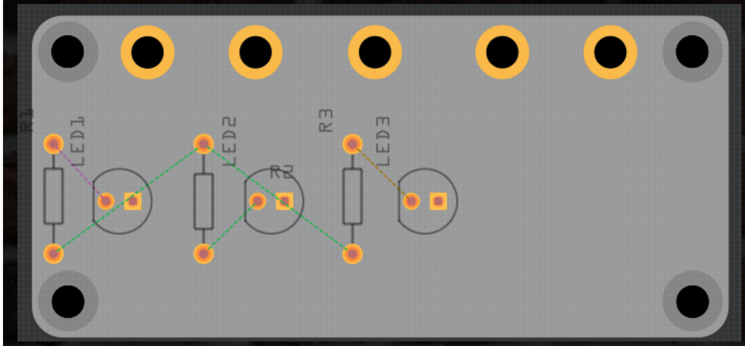


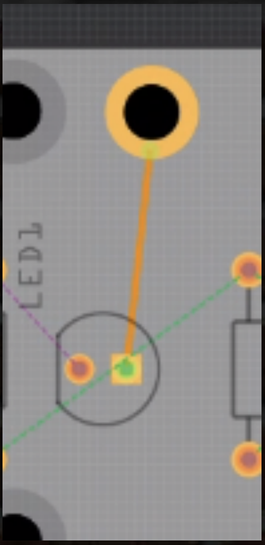


PCB Design

It’s now time to lay our PCB out properly. We eventually want to have an image on the right side of the PCB, so in this example, we will be putting the LEDs towards the left of the board with a resistor next to each LED. We suggest putting the LEDs in the middle/towards the bottom of the PCB to allow enough room for the micro:bit when it’s connected to the screw holes.

To move a component simply click and drag it to the desired position. The layout we went for is on the right.





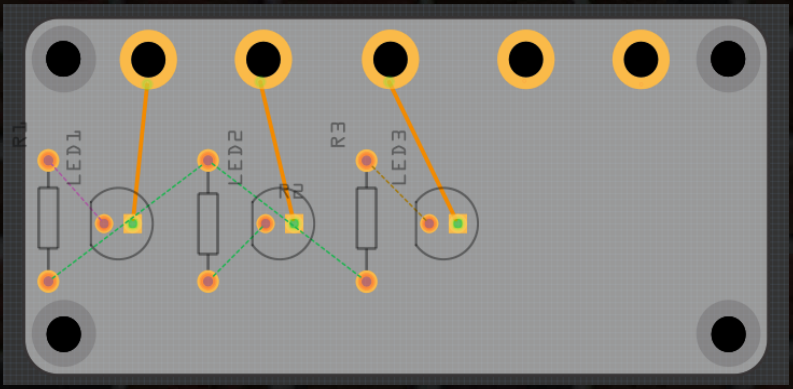
Now we have the PCB layout done, it’s time to create some traces! Each screw hole on our template has a small connector. If you zoom in on a screw hole you’ll be able to see it. Each component also has holes in which the components legs will go through, we can use these connectors to draw a trace from the micro:bit to the component.

It’s important that before we draw traces, we make sure that the view setting on the bottom toolbar is set to “View from Above” so you connect the correct components to the correct micro:bit pins!

Let’s create the first trace, connecting the first LED towards the left to Pin 0. You’ll notice that each LED component has a flat and a curved side. The flat side is negative and the rounded side is positive. The positive leg will go to the pin whilst the negative leg will go to ground.

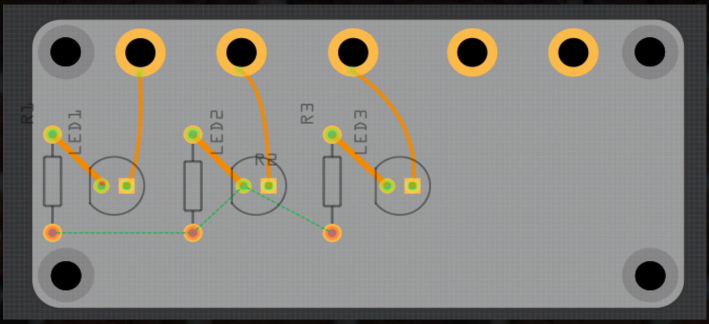
To create a trace between the positive leg of the first LED and Pin 0 (the first screw connector at the top on the left) click the positive leg hole of the Led and drag it towards the bottom of the pin 0 screw connector until the orange trace highlights blue. You should now have a trace connecting the two together.

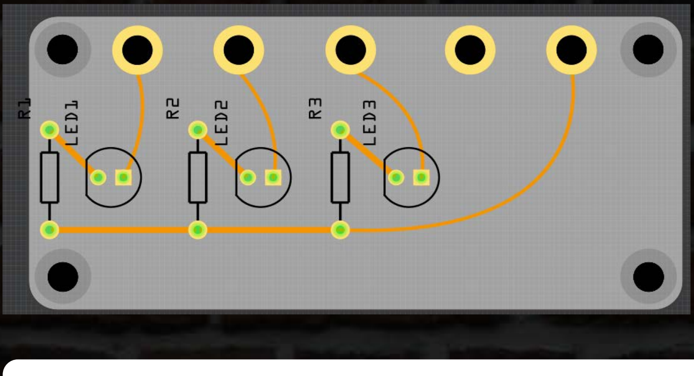
Now do the same with the positive leg of each LED with the second LED going to Pin 1 and the third LED going to Pin 2. You should now have a PCB that looks like the one on the right. To make the traces look a bit nicer, you can curve them the same way as in the breadboard tab. Simply hold down CRTL and drag the middle of the trace. To make the traces go at sharper angles, just drag at any point of the trace.



PCB Design

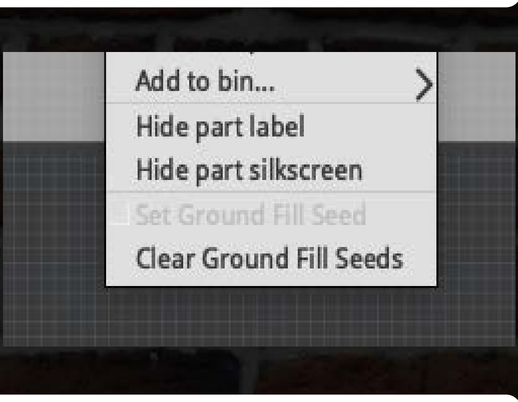
Now we need to connect the negative leg of each LED to the resistor to the left of it. To do that, create a trace the same as you did before and connect the negative leg of each LED to the top connector of the resistor it is next to. The image on the right shows you how to do this.

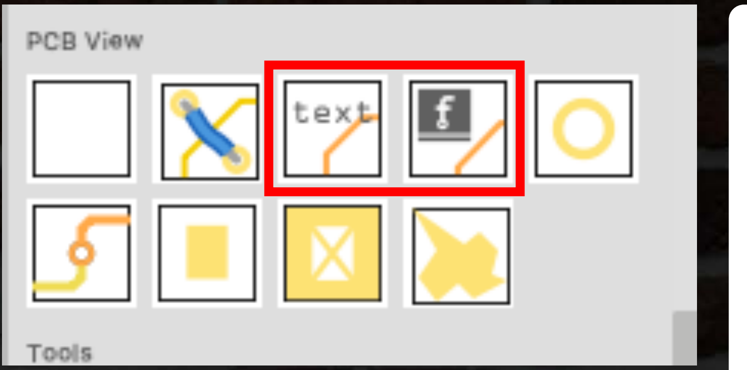




The last step of wiring up our PCB is to connect all the resistors to the GND pin of the micro:bit. Create traces between each bottom connector of the resistors and the wire the one on the right to the last pin on the micro:bit as shown below. Once that is complete you have finished routing your PCB!

The last thing we need to do is add some silkscreen. Currently we already have some silkscreen elements in the form of component outlines and labels. It's up to you whether you want to keep these, we personally like to remove the labels and keep the component outlines. To remove any of these, right-click on the component and select "Hide Part Label" or "Hide Part Silkscreen".



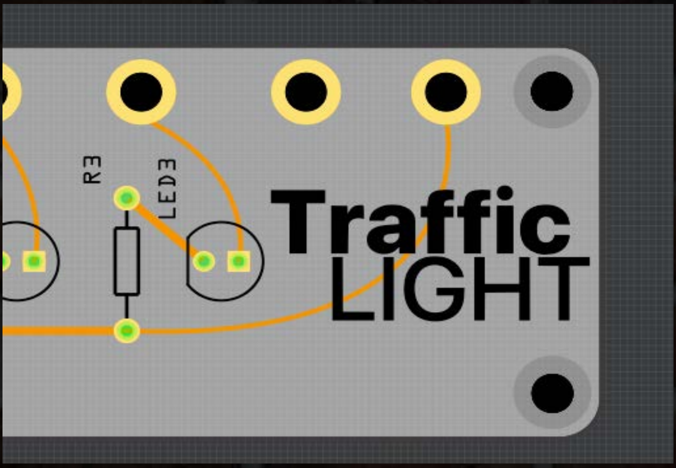


To make our board more interesting we can add text or images. To find the parts for text and images, select the CORE category in the parts view and then scroll down to the bottom to find "PCB View". In here you'll find the text and image components. To add one to the PCB, simply drag the one you want onto the PCB board.

Depending on where you want to add the image/text, you'll need to change the Layer in the bottom toolbar before dragging. In this example, we'll be adding an image to the top layer so we'll press the layer button to switch to the top before dragging in the image component.

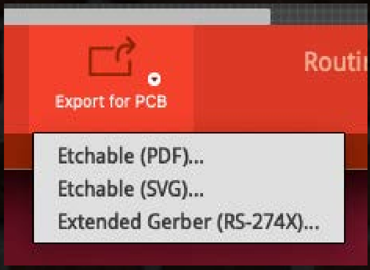
PCB Design

For an image to show up on the PCB, it needs to be a black, transparent SVG. To change the image from the default setting within Fritzing, in the inspector view go to properties and then click the Load Image File. We recommend SVGs for the best results however you can use PNG files instead. The image you load might be quite large so you may need to zoom out and use the "Handlebars" in the 4 corners of the image to resize it. Once you've got it to the size you want, drag it into the desired position on the PCB.



Manufacture

You've successfully created your own PCB, congratulations! Now we need to actually get it made. In order to do this, we need to export some files. PCB Manufacturers accept GERBER files, luckily Fritzing can generate these and it's as easy as putting them in a zip file and uploading them to the PCB manufacturers website. To do this, click the small arrow within the "Export for PCB" button in the bottom toolbar. From the dropdown, select "Extended Gerber" and select a folder to export.



Once you've zipped up your Gerbers, it's time to send them off to be made. It's good to order around 3-5 for a protoype run so below we've listed 3 of our reccommended and affordable places to get your PCB made.

OSHPARK

Slightly more expensive than others but oshpark are very friendly and active within the maker community.

oshpark.com

AllPCB

Fast PCBs with a variety of shipping options all around the world. AllPCB offers a range of offers for first time orders/customers.

allpcb.com

JLCPCB

JLCPCB is currently a popular choice amongst makers and enables you to get prototypes for a very low cost.

jlcpcb.com

We hope you've enjoyed and learnt something in this tutorial. We can't wait to see what PCB creations you make!

By **Joshua Lowe**

CALL FOR CONTRIBUTIONS

WRITE AN ARTICLE FOR THE NEXT MICRO:MAG

We want to hear about your awesome micro:bit powered project, event or story

FILL IN THE CONTRIBUTIONS FORM AT MICROMAG.CC/CONTRIBUTE

Even if you've never written for a magazine before, our team are here to help you get your article in micro:mag.

micro:bit

Keep on the move with micro:bit

By Les Pounder

About the author

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

@biglesp

bigl.es

You will need...

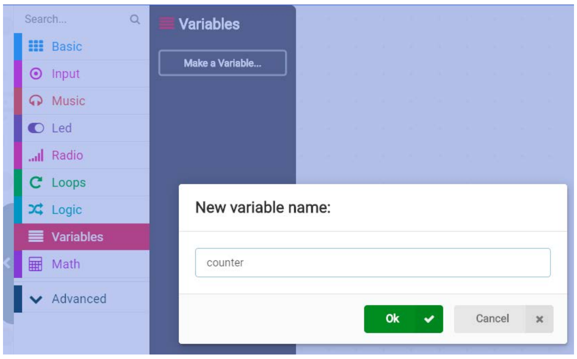
- » BBC micro:bit
- » USB Cable

Keeping fit is very important and it is common knowledge that we should walk 10,000 steps a day to keep ourselves healthy. But how do we count the steps? Well for that we need a step counter, and with the micro:bit we have one in just 9 easy to follow steps! This tutorial assumes that you are already familiar with the micro:bit MakeCode editor, which can be found at

https://makecode.microbit.org

Step 1 - Where do we store data?

Variables are containers used to store data. We create a variable by going to the Variables section, and then click on Make a Variable... Call the variable counter.

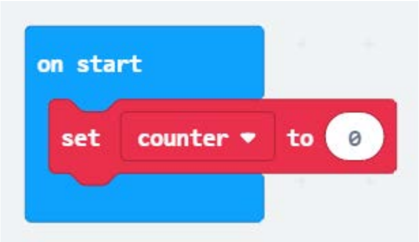


New variable name:

counter

Step 2 - On Start

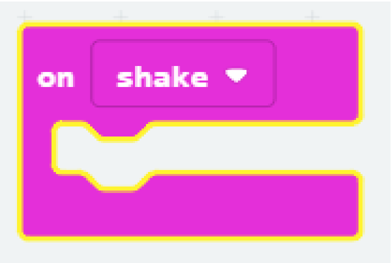
When the micro:bit starts, we want the counter variable to reset to zero. So go to Variables and place the set counter to 0 block inside on start.



```
on start
    set counter ▾ to 0
```

Step 3 - Taking a step

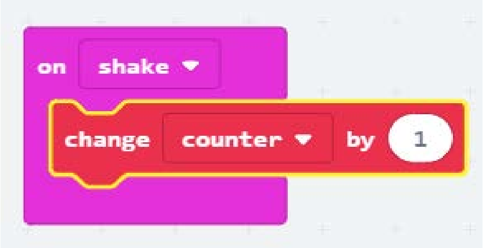
When we take a step, the micro:bit will shake, and we use this as a trigger. From the Input section drag On Shake and place it into the code.



```
on shake ▾
```

Step 4 - Add to the counter

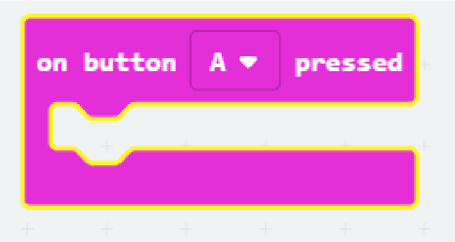
Each time we take a step, we need to save that data to the counter variable. From the Variables section drag change counter by 1 and place it inside the on shake loop.



```
on shake ▾
    change counter ▾ by 1
```

Step 5 - Push Button Data

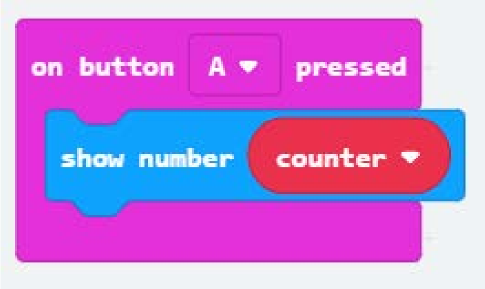
The step count is not displayed by default. To see the screen we need to press the A button. From Input drag on button A pressed and place it into the code.



```
on button A ▾ pressed
```

Step 6 - Show the step count

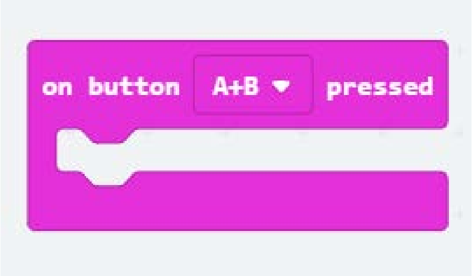
To show the step count we need to drag the show number block from Basic and then drag the counter block from Variables. The counter block stores the step count.



```
on button A ▾ pressed
    show number counter ▾
```

Step 7 - Two button presses

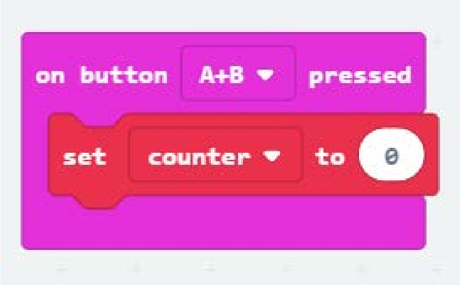
From the Input section drag on button A pressed into the code. Change the dropdown to show A+B so that the code contained within only activates if both buttons are pressed.



```
on button A+B ▾ pressed
```

Step 8 - Reset the counter

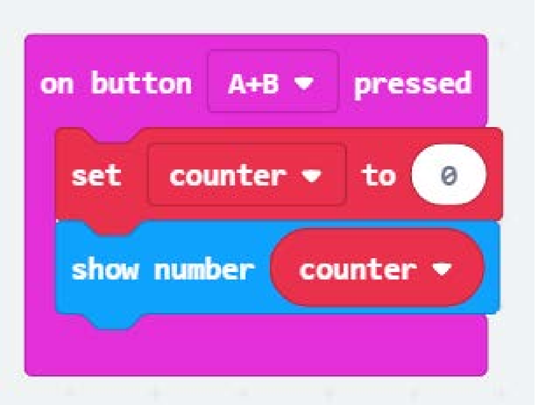
When both buttons are pressed we want the counter to reset to zero. From Variables drag the set counter to 0 block and place it inside the loop.



```
on button A+B ▾ pressed
    set counter ▾ to 0
```

Step 9 - Giving feedback

When both buttons are pressed we want the counter to reset to zero. From Variables drag the set counter to 0 block and place it inside the loop.

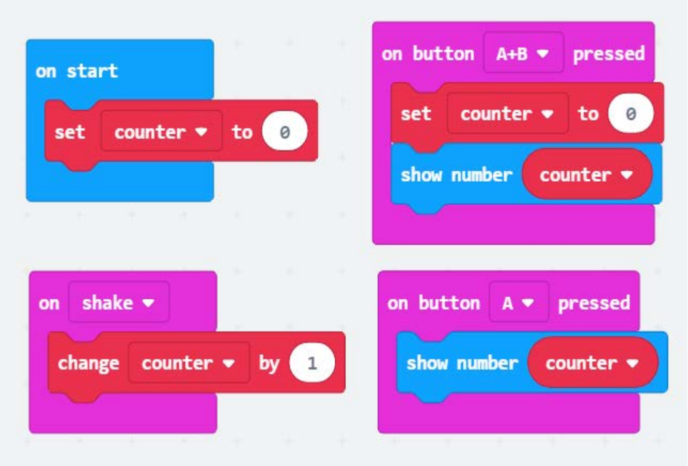


```
on button A+B ▾ pressed
    set counter ▾ to 0
    show number counter ▾
```

With the code now complete, click on download to download a HEX file to your Downloads folder, and then flash the code to your micro:bit by connecting the micro:bit to the computer, and dragging the HEX file to the micro:bit drive in your file manager. Once flashed, remove the micro:bit and power it from a USB battery or using the 2 x AAA battery holder that came with your micro:bit.

Now go for a walk, keep fit and above all, keep safe!

Full Code



```
on start
    set counter ▾ to 0

on button A+B ▾ pressed
    set counter ▾ to 0
    show number counter ▾

on shake ▾
    change counter ▾ by 1

on button A ▾ pressed
    show number counter ▾
```

We need a new writer for micro:hit!

We'd like to congratulate Les on his new full-time Job at Tom's Hardware!
We're on the lookout for a new writer for the micro:hit section of the micro:mag. If you're intrested, please email:

hello@micromag.cc

Download the project:

You can try out this project on the micro:bit makecode website, take a look here

makecode.microbit.org/_gWe76bVcWWLK

MIDI WITH MICRO:BIT

BY David Perks

micro:bit meets Musical Instrument Digital Interface. Access the power of professional music equipment.

About the Author

David is an artist-maker who has added electronic junk and microprocessors to his palette. Currently making a lot of noise in his garage.

Twitter: @pdbperks

What is MIDI?

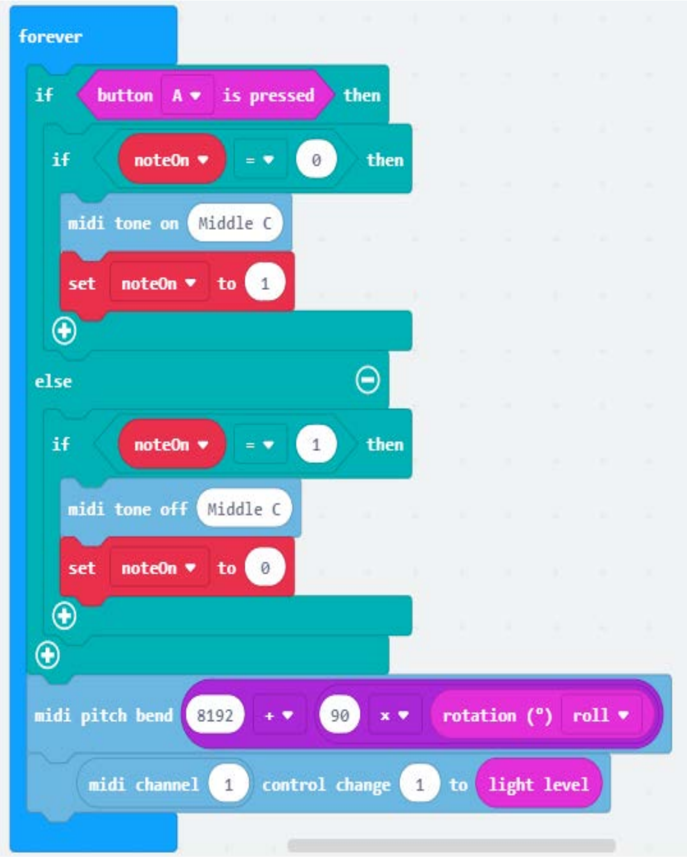
The Musical Instrument Digital Interface is an industry-supported protocol created in 1983 to allow digital electronic instruments to communicate with each other using a 3-wire serial connection. All MIDI messages consist of an eight-bit status byte which is generally followed by one or two data bytes: for example, the sequence 144, 60, 127 (hex 0x90, 0x3C, 0x7F) means turn note on, middle C, full velocity. MIDI's primary use is in audio but it is also used to control lighting. The original 5 pole DIN cabling has been supplemented by USB, WiFi, and Bluetooth. The MIDI Association (www.midi.org) exists to support and promote its continued technical and creative use.

Midi & micro:bit

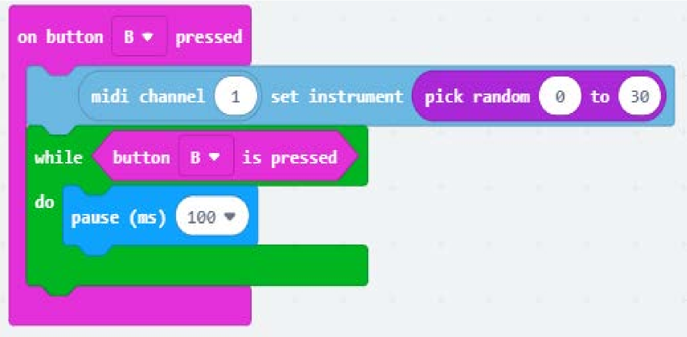
MakeCode provides good support for MIDI: you need to navigate to Advanced/+Extensions, search for and install the midi interface. To use the hardware circuit, redirect serial TX to P0 and set the baud rate to MIDI standard 31250. Use raw serial for the MIDI messages. David is going to keep the main program loop quite simple. They want a controller that will add missing effects to their musical keyboard but also have potential as a creative instrument.

Pressing button A will turn on note Middle C. Variable noteOn is used to track the state of button A and turn off Middle C when it is released. This allows David to control the duration of the note.

Two features missing from David's musical keyboard are pitch bend and modulation wheels. The pitch bend effect ranges from 0 to 16,383 with 8192 representing no change. Since the micro:bit roll sensor returns a value in degrees positive or negative from level, multiplying that value by 90 will approximately fit that range. He could use the Math/map function to give a precise correlation but they have found that slight imperfections in a MIDI instrument can add interesting character. For this reason, David uses light level (0 – 255) as a control change parameter for controller 1 (modulation wheel, 0 - 127) since this will give them a better response in poor light and any value over 127 will simply start again at 0. These effects will work on the note played by button A or on notes played by the musical keyboard. Pressing button B will change the instrument. David has chosen to make a random selection. The maximum number of instruments in a sound bank would be 127, but some may contain as few as 20 so adjust this value as required.



```
forever
  if button A is pressed then
    if noteOn = 0 then
      midi tone on Middle C
      set noteOn to 1
  else
    if noteOn = 1 then
      midi tone off Middle C
      set noteOn to 0
  midi pitch bend 8192 + 90 x rotation (°) roll
    midi channel 1 control change 1 to light level
```



```
on button B pressed
    midi channel 1 set instrument pick random 0 to 30
  while button B is pressed
  do
    pause (ms) 100
```

MIDI over USB

Some microcontrollers (the Circuit Playground Express or Arduino Micro) offer true USB MIDI interfaces; a computer recognises these boards as MIDI compatible allowing them to communicate directly with software synthesizers just like most modern MIDI music keyboards. Although this is not the case with the micro:bit, it is possible to convert messages sent as serial into MIDI by using a cross-platform software program such as Hairless MIDI. Just remove the serial redirect block from the MakeCode example for the MIDI messages to be sent by USB serial at 115200 baud. On Windows 10, David has to use an extra virtual MIDI bridge called loopMIDI before they can connect to their favourite free synthesizer - NanoStudio1 (www.blipinteractive.co.uk/nanostudio1/).

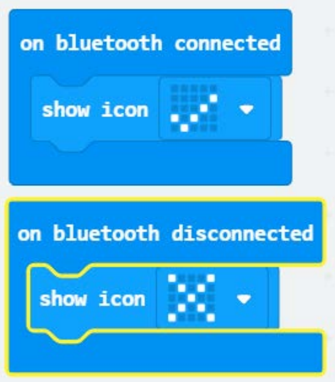
On Linux, David has used Hairless MIDI via AconnectGUI to amSynth, so if you have a high spec RaspberryPi you might get it working on that. If you have Chrome web browser you could try https://nicroto.github.io/viktor/, a synthesizer that uses the HTML5 webmidi API. To enable webmidi type chrome://flags/#enable-web-midi.

Setting up the MIDI software bridges is more complicated than establishing a DIN cabling connection but in contrast, there are no hardware requirements other than the USB cable.

MIDI over Bluetooth

This is a great use for micro:bit Bluetooth and in some respects the easiest MIDI configuration to establish. It must be noted, however, that Bluetooth uses a lot of memory and David encountered problems when trying to add extra code. The first two methods could be written using MicroPython but this is not an option for Bluetooth.

In MakeCode blocks navigate to Advanced/+Extensions and install the Bluetooth-midi interface. It will warn you that this will disable radio functions. You can use the same forever loop but change the on start routines to include on Bluetooth connected. The extension currently only supports iOS



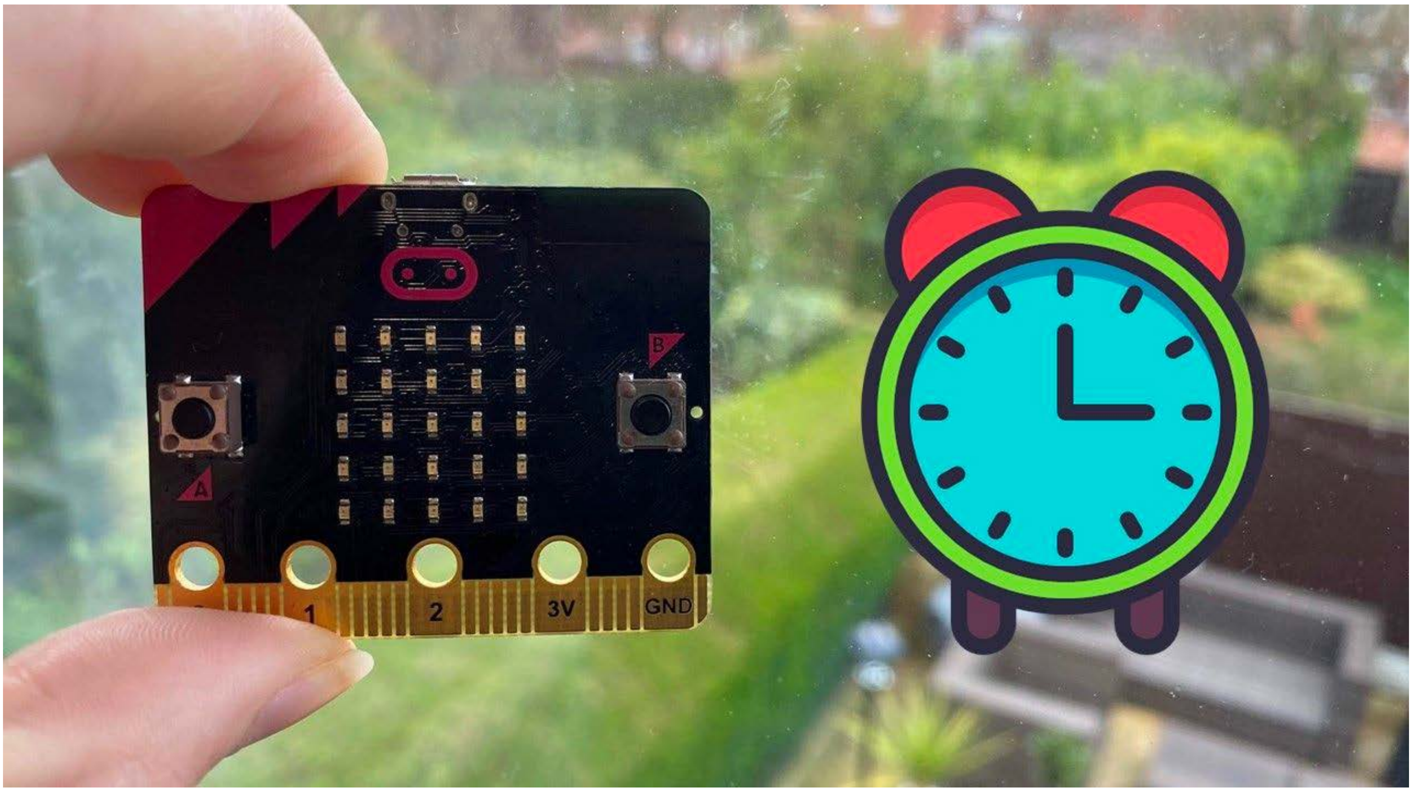
```
on bluetooth connected
  show icon
on bluetooth disconnected
  show icon
```

Left:
The code for the bluetooth connection is straight forward.

David used the iOS micro:bit app to pair the board with their iPad. In the app, tap Choose micro:bit then tap Pair a new micro:bit. For the initial pairing you need to hold down the micro:bit buttons A and B, then press and release the reset key on the back. Wait until the lights have filled the screen and follow the instructions provided by the app. You will need to go to Settings/Bluetooth and connect the BBC micro:bit under MyDevices: He finds that they have to repeat this every time they want to connect to the iPad.

Left:
The Minimoog Model D.
The world's first portable synthesizer.

David has successfully used several synthesizer apps with the micro:bit and would recommend the Minimoog Model D app or Audikit Synth One. In the Minimoog app, click the settings cog icon in the top right-hand corner, select MIDI option and click the Bluetooth Midi button to list and connect your device.
In the Minimoog MIDI preferences screen, there is an option to Map CCs. This allows us to link sensors on the micro:bit to the Minimoog: by using the midi control change block we can adjust specific controls in the software synthesizer. This MIDI learn feature is found in most software synthesizers and allows us to create inventive controllers for professional quality digital instruments. Have fun.



NON TOUCH HANDWASH TIMER

BY Daryl Croke

Handwashing is more important than ever with the pandemic, learn how to make a timer with micro:bit.

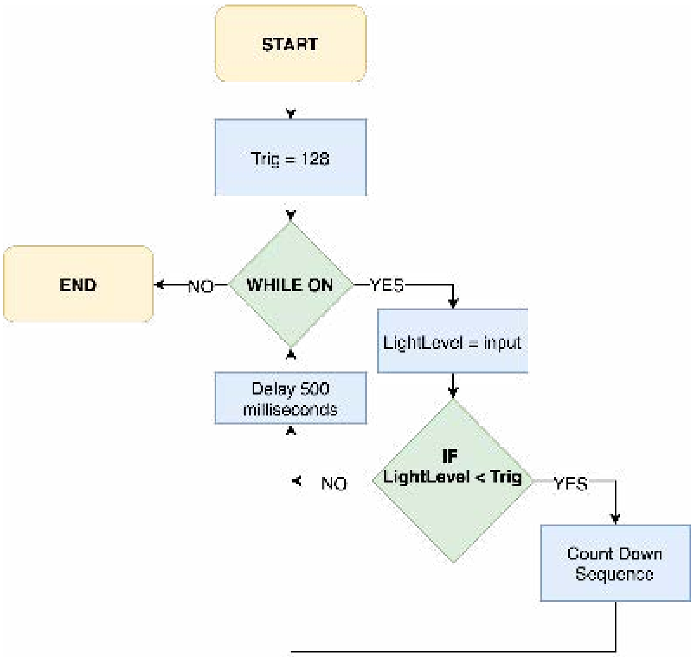
At the time of writing this article Melbourne, the city where Daryl lives and works is still firmly in lockdown. All schools are shut, and most students are learning remotely. History will be the judge if the relatively low Australian Covid-19 death toll was a result of decisive government action, or good luck. One thing we can be certain of is to avoid a second wave of infections our behaviours need to change.

One behaviour that needs to change is our attitude to hygiene especially regular hand washing. Governments around the world have been correctly promoting the need for thorough hand washing for at least 20 seconds.
As a high school teacher, Daryl envisaged a 20-second no-touch timer that could be easily deployed in school toilets to encourage students to wash their hands for the required length of time. They wanted to develop a solution that didn't require using an external sensor, wiring or perhaps the need for separate power supplies. It had to work out of the box, simply download a program, install batteries and attach to wall or mirror. It also has to work, without maintenance, for a couple of weeks.

He chose the micro:bit as his weapon of choice because:
1. had some lying around
2. currently teaching a unit on the micro:bit
3. It has an LED matrix
4. There are built-in sensors

The question then became what built-in sensor could we use? Using a button would be simple enough but would require the user to physically touch the micro:bit; not very hygienic. The same could be said for accelerometer and compass. Forget about the temperature sensor unless you were prepared to hold your hand near the micro:bit for an indefinite time. Then like Paul on the road to Damascus, Daryl saw the light or more correctly figured they could use the Ambient Light Sensor to detect a human hand waving near the micro:bit sensors.

Well the solution was simple right? Just set a threshold variable and when ambient light level drops below that threshold, caused by your hand covering the micro:bit, trigger a 20-second countdown timer.

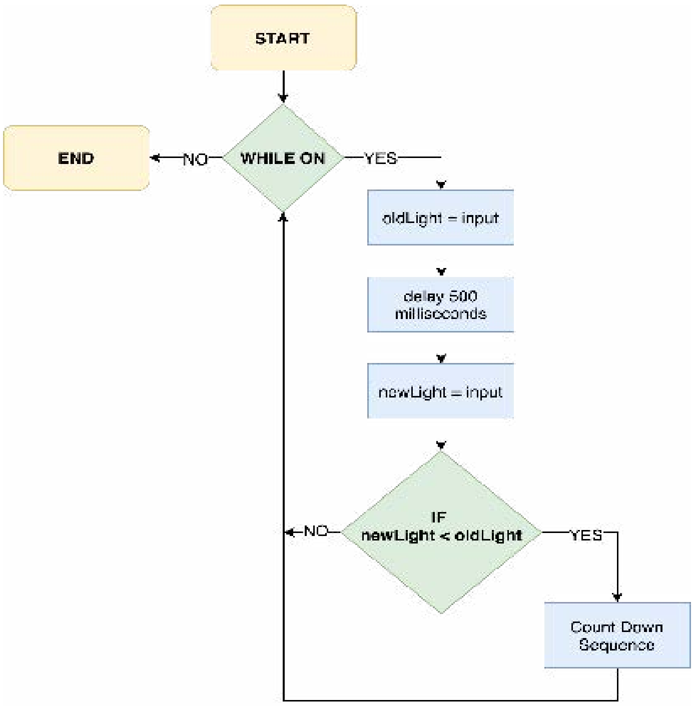


Problems

Ambient light levels change from room to room and light levels change throughout the day. It wouldn't be feasible to re-program the micro:bit to suit every bathroom in a house, office or school. Every different situation would require you to monitor and set a different trigger level for each room. The ambient light level in the Year 8 boys' toilet might be 67 in the morning. The light level in the Year 10 girls' toilet might be 87 in the morning. Light levels might change again in the afternoon or on a cloudy day.

Version 1

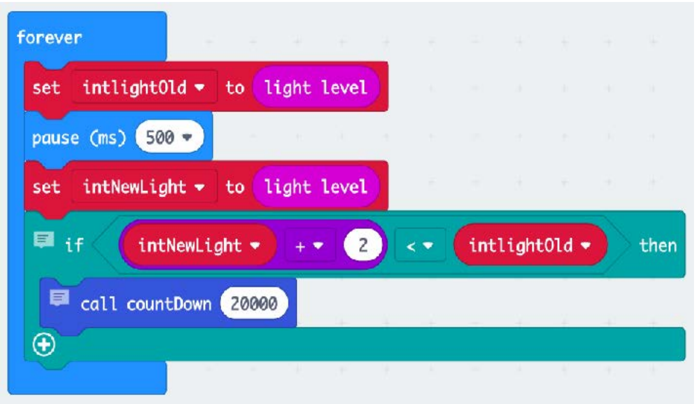
Rather than setting a generic trigger level, we need to measure a drop in light levels over a short period of time. In the algorithm below Daryl set an "oldLight" level to the ambient level then delay for 500 milliseconds before setting a "newLight" level. If there is drop from old to new, caused by a hand blocking light, then the countdown sequence starts.



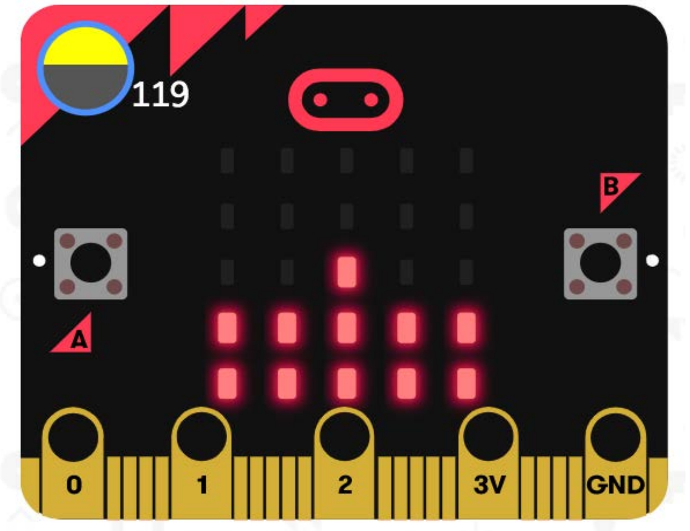
Daryl successfully developed this solution in MakeCode and used it at home in their bathroom for a couple of weeks. It worked without a hitch and it did prompt a change in their behaviour and they got into the habit of thoroughly washing their hands every time, no cheating. So, deployed in a school it should have a positive effect on personal hygiene.

I created a Youtube explanation here: https://www.youtube.com/watch?v=L6vPFa2oQiA&t=113s

The MakeCode link is here:
https://makecode.microbit.org/28197-50014-62398-74184



```
forever
  set intlightOld ▾ to light level
  pause (ms) 500 ▾
  set intNewLight ▾ to light level
  if intNewLight ▾ + ▾ 2 < ▾ intlightOld ▾ then
    call countDown 20000
  ⊕
```



Problems

Although it worked well in Daryl's bathroom at home, they can see problems in high use situations.

For a start, it will continue to run when it is not needed at night. Version 1 checks changes in light levels 7200 times an hour. Daryl developed a "low-power version" that will only check for changes in light levels if the room light level is above 4. That is when light is in the room. It checks if the light is in the room every 30 seconds. This will reduce the number of loops at night from 7200 to 120 times an hour. Hopefully this will prolong the battery life.

Secondly by default micro:bit sets the LED output to maximum brightness. In a dimly lit room, it is not necessary to for the LED to be at maximum brightness to be seen. Daryl tweaked the program so that if the light level is under 64 set the LED brightness will be 128, half the maximum value. Hopeful this will also prolong battery life.

Thirdly on advice from a colleague Daryl reduced the complexity of the countdown sequence, so the timer now only loops 5 times instead of 40. Not as elegant but batteries will last longer.

Low Powered Version.
Link to version 2.
https://makecode.microbit.org/_R90WYXLHcJ6T

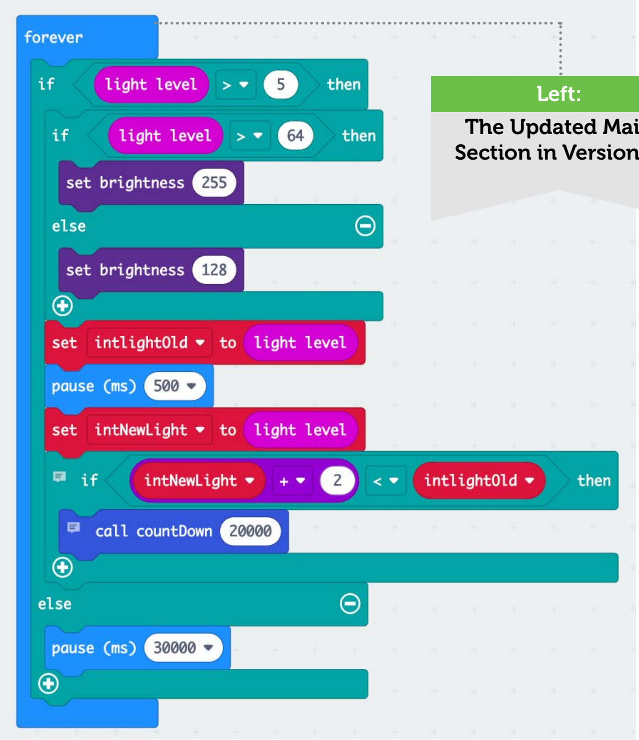
Version 3

At the time of writing, Daryl considering creating a version that will self-calibrate. One possible development could be to run through a 10-second calibration loop every 30 minutes. The idea is to work out the difference between the highest and lowest light level or natural change over that period. Any drop in light level greater than the natural variation would have to come from a human hand blocking some of the light hitting the sensors. A consideration against doing this is that the program may become more difficult to explain and teach.

Proposed Deployment

Sooner or later we will come out of lockdown, schools and other large workplaces will operate while the virus is still in the community. Social distancing and handwashing will be more important than before the lockdown. Daryl proposes using existing class sets of micro:bits in school toilets and explains to students how the countdown timer works, including why it is important to use it properly. In other workplaces, Daryl would strongly suggest employers should cover the cost of a hygienic non-touch countdown timer.

Daryl's not sure of the best non-destructive methods to attach a micro:bit and battery to wall or mirror. They imagine a common adhesive picture-hanging strip would suffice. The micro:bit and battery could be housed in a clear box that is then attached to a wall or mirror. Once deployed, he hopes that the micro:bit would run for 2 weeks before its batteries need replacing. Hopefully by this time a change in handwashing behaviour has occurred. For it to be a sustainable long-term solution, rechargeable batteries and battery chargers should be used.



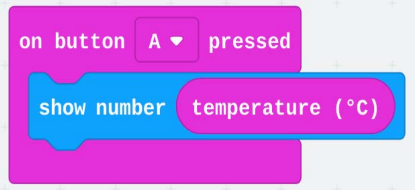
```
forever
  if light level > 5 then
    if light level > 64 then
      set brightness 255
    else
      set brightness 128
    set intlightOld to light level
    pause (ms) 500
    set intNewLight to light level
    if intNewLight + 2 < intlightOld then
      call countDown 20000
  else
    pause (ms) 30000
```

Left:
The Updated Main Section in Version 2.

Basic Weather Station

Use a micro:bit and Makecode to measure temperature. By **Rob Wilson**

A micro:bit can measure temperature all on its own - try it!



```
on button A ▾ pressed
    show number temperature (°C)
```

The microchip contains a simple sensor, but it's not very accurate - it might have an error of 4C.

Can we do anything about that?

You could compare your micro:bit's reading with another temperature sensor that you know is reasonably accurate. The add or subtract the difference between them, using one of the Maths blocks.

Make Measurements over one day

Now we will write code to make temperature measurements, and save them, every 15 minutes for a whole day. You can take the micro:bit anywhere to measure the temperature - but keep it dry!

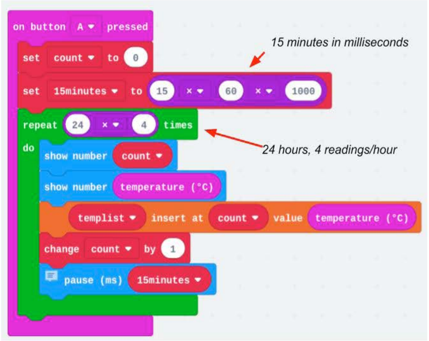
We will read the measurements the next day, at about the same time. We could use them to plot a graph. Our code will need a list, or array, to put the temperature readings in. We need to give it a name - let's use templist. You can think of a list as a table, where the first column shows the position (which we called count) of the temperature reading:

count	temperature
0	21
1	21
2	22
95	25

Start Coding!

Open MakeCode (makecode.microbit.org) to start coding.

Here is the code to make our measurements:



```
on button A ▾ pressed
    set count ▾ to 0
    set 15minutes ▾ to 15 × ▾ 60 × ▾ 1000
    repeat 24 × ▾ 4 times
    do
        show number count ▾
        show number temperature (°C)
        templist ▾ insert at count ▾ value temperature (°C)
        change count ▾ by 1
        pause (ms) 15minutes ▾
```

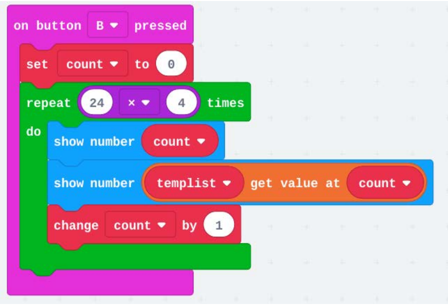
15 minutes in milliseconds

24 hours, 4 readings/hour

We are using a variable called 15 minutes - the micro:bit timer uses milliseconds (1 second/1000) rather than minutes, so we convert 15 minutes to 15 x 60 x 1000 milliseconds.

We make 4 measurements an hour, so we need to run our loop 24 x 4 times.

Code to read measurements



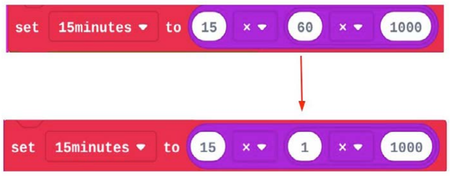
```
on button B ▾ pressed
    set count ▾ to 0
    repeat 24 × ▾ 4 times
    do
        show number count ▾
        show number templist ▾ get value at count ▾
        change count ▾ by 1
```

This will display the count number and temperature.

Enter both of these code blocks.

Test your code

You can speed up your code while you are testing by changing this block. Now, your 15-minute timer runs 60 times faster! *Don't forget to change back to 15 minutes and save it!



```
set 15minutes ▾ to 15 × ▾ 60 × ▾ 1000
```

```
set 15minutes ▾ to 15 × ▾ 1 × ▾ 1000
```

Test over one day

You must keep the micro:bit switched on until you have finished reading out the data - otherwise your measurements will disappear!

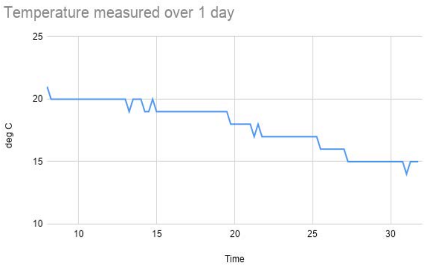
1) Start the measurements by pressing button A.

2) The next day (at least 24 hours later), press button B to see the data.

You can write down the temperatures to use later. There are other ways to record the data, including sending the readings to a computer...

Using your readings

You can save the data and convert it (using a spreadsheet, perhaps) to a chart, like this:



What Next?

We have shown how a micro:bit can be used to measure and record the temperature; we can also read out those measurements to use in a graph. In future articles, we can use WebUSB to send the readings to a computer, and add a different way to measure temperature, which can also measure humidity.

Husky Lens

Unlock your robots full potential by adding the husky lens for Artificial Intelligence. By **Kerry Kidd**.

AVAILABLE FROM
DF Robot

dfrobot.com/product-1922.html

What's Included?

- HuskyLens Mainboard x1
- M3 Screws x6
- M3 Nuts x6
- Small Mounting Bracket x1
- Heightening Bracket x1
- Gravity 4-Pin Sensor Cable x1
- Silicone Case

Price

$44.90 USD
€49.21 EUR
£49.50 GBP
Approx

The husky lens is an AI vision sensor for use with micro:bit, Raspberry Pi and Arduino over I2C, of course, Kerry has been testing it out with the micro:bit.

The husky lens is fairly easy to get started with as DFRobot have a great wiki taking you through on what the vision sensor does. This includes face recognition, Object detection, tag detection (like QR codes), colour detection, object tracking and line tracking.

To use the vision sensor you can either use mind+ coding environment developed by DFRobot or there is an extension for MakeCode which is the one we have been using due to MakeCode being more widely used within the micro:bit community. To get the extension on MakeCode you need to put in the Github link found on the DFRobot wiki rather than just typing "huskylens" as it's not within the official extensions for MakeCode at the writing of this review. One complaint I do have is that the extension has been updated since the writing of the wiki as the code blocks don't match within the code on the wiki and the actual code blocks within the MakeCode extension. This can cause confusion for beginner coders.

The husky lens is a great board for an introduction to AI vision and is great for beginners to get started.

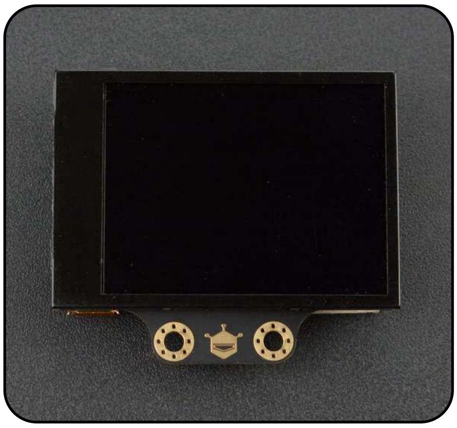
The husky lens would pair nicely with a robot that you want to add AI vision to as you can train it to detect and track objects as well as to follow lines.

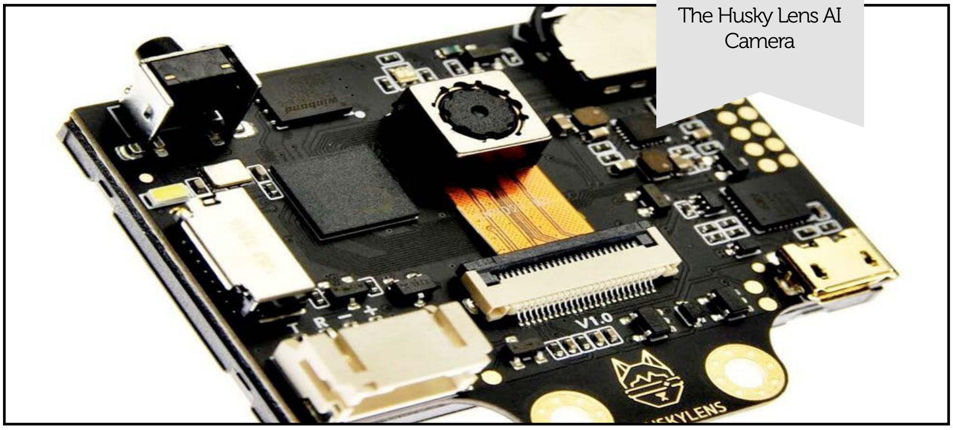
One thing we'd like to see in the future is MicroPython support for this board. As it's an advanced add-on, support for MicroPython seems like a must of a board like this.

If you're looking to do artificial intelligance with the micro:bit, the huskylens is a good option.

OUR RATING

9/10





Below: The Husky Lens AI Camera

Similar Products

Two other products to consider...

01 Kittenbot AI Module

A similar sort of idea as the HuskyLens but slightly more expensive at $69 and has less features.

go.micromag.cc/koi

02 M5Stick V AI Camera

Not compatiable with the miicro:bit, but the M5StickV is also a good option for AI.

go.micromag.cc/stickv

4Tronix Mars Rover

Build your own BBC micro:bit powered Mars Rover with this kit from 4Tronix. By **Joshua Lowe**

AVAILABLE FROM
4Tronix

go.micromag.cc/mars

What's Included?

- Drive:Bit Board

Price (When released)

$157 USD
€133 EUR
£120 GBP
Approx

We see a lot of micro:bit robots here at micro:mag but this one is different. Ever wanted to own your own working Mars Rover? Well, now you can with the 4Tronix Mars Rover kit.

At £120, this kit isn't the cheapest micro:bit robot at all but it's different to every other micro:bit robot. The majority of micro:bit robots follow the same formula of 2 wheels, a caster and a PCB chassis, whilst the Mars Rover from 4Tronix breaks this mould.

The rover has 6 wheels, of which 4 are DC motor powered and the other two are powered by a combination of servos and DC motors.

The robot takes around three hours to build depending on how skilled you are at putting these kits together. We managed to build our kit with little issues and it was overall an enjoyable few hours. Some parts are fairly tricky but luckily 4Tronix have provided detailed build instructions with images.

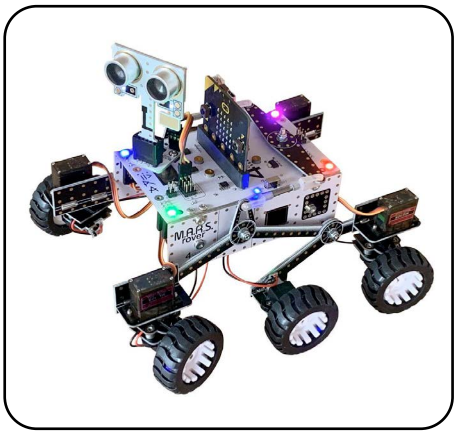
This robot is excellent for climbing over obstacles due to it being six wheel drive and the way the robot's legs are constructed allows for the it to drive over rocky terrain with ease.

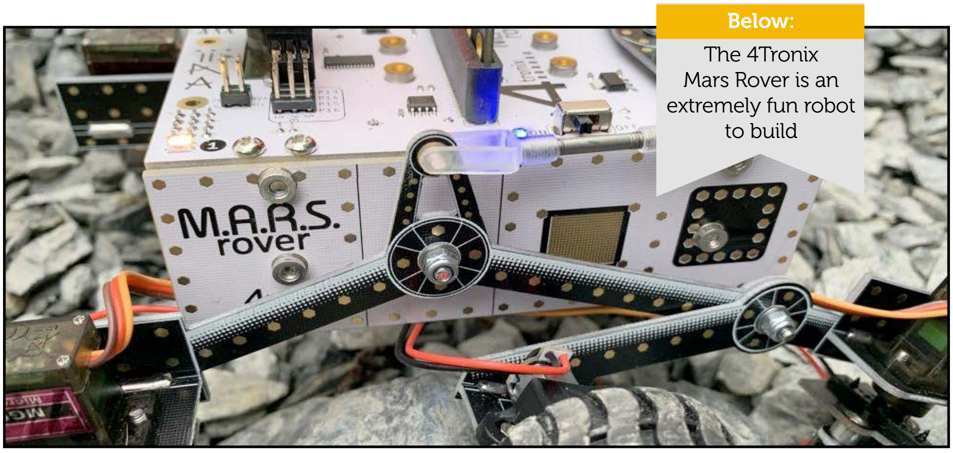
The robot also includes an ultrasonic sensor which sits on top of the front of the robot and the sensor can tilted from left to rght to control line of sight.

4Tronix have provided an easy to use MakeCode extension which allows you to control all of the features of the rover, including the 4 neopixel LEDs. There's no official MicroPython library yet, and this is something we'd like to see in the future, especially for a robot this expensive! However we feel that for what you get, the price is very fair!

OUR RATING

9/10





Below: The 4Tronix Mars Rover is an extremely fun robot to build

Similar Products

Two other products to consider...

01 4Tronix mini:bit

One of our favourite budget robots if you can't justify the price of the 4Tronix Mars Rover.

go.micromag.cc/minib

02 4Tronix Bit:Bot XL

A favourite robot amongst members of the micro:bit community and a solid option.

go.micromag.cc/bitbot

Waveshare 1.8″ LCD

Add a tiny colour LCD to your micro:bit for just under £20.
By **G. Michael Wood**

AVAILABLE FROM
Amazon

go.micromag.cc/18lcd

What's Included?

- 1.8inch LCD for micro:bit

Price

$25 USD
€21 EUR
£18.99 GBP

Approx

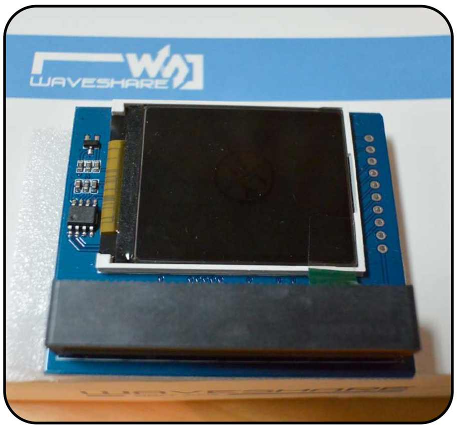
It can, after a while, get tiresome reading numbers and words scrolling across the 5-X-5 LED matrix onthe micro:bit, so at that point, one starts looking for options available to provide visual output.

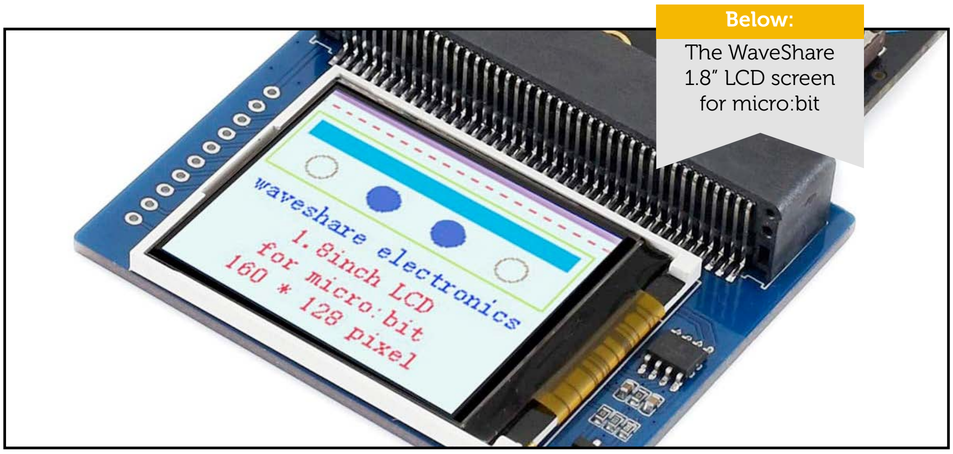
One option is the Waveshare 1.8″ diagonal LCD display with 160 x 128 pixels that can display 65k colours. It has a plug-in micro:bit adapter and the display can show both text and (as we will see) simple images.

Waveshare is a company that has made many accessories for the small computer world and this is their first (and so far, only) display for the micro:bit. The 1.8″ colour display arrived in a small box a few weeks after being ordered and paid for online. It was carefully wrapped in bubble wrap and a piece of closed-cell foam. I suspect the damage-in-transit rate is very low.
When the micro:bit is inserted into the Wavebit LCD Display the latter is located "below" the Micro:bit....and the default (and unchangeable) text output means that any use of text requires that this orientation be the one that is used.

The Waveshare LCD display has a built-in character set which allows 10 lines of 22 characters each of what appears to be a Courier non-proportional 7-pixel wide font that is reasonably easy to read. Overall the display is sharp and edges of graphics and characters are clear. Although the colour of any text string can be easily set, there is no option to change the size or style of the text (so no bold or italics options). Depending on the character/number symbol design the actual characters can be up to 7 pixels wide (e.g., "m" or "u") but most have a 1-pixel space buffer before and after each character. However, the width of some characters means that there is no visible gap between letters when they are beside each other (such as with the word "Humid" where the "u" and "m" touch on the Waveshare display). The alphanumeric characters are up to 8 pixels in height. However, letters with "descending" parts, such as lower-case "y" require 2 pixels below the base "line" for the characters. In addition, there is a 1-pixel buffer allocated above and below the tallest and lowest letter, so the overall line "height" of text lines (with the 1-pixel buffers) is 12 pixels. On a display 128 pixels high this allows for 10 lines of text with a buffer at the top and the bottom of the display (or elsewhere if one desires as any text line is located by display-level x/y coordinates). The display overall is bright and clear (artefacts and ghosting in the images are the results of the difficulty of taking a digital picture of a digital source).

Note that designing a display for even just text output can be time-consuming because each line needs a pixel-designated





Below: The WaveShare 1.8″ LCD screen for micro:bit

X & Y start so one must take into account the height of each line and any pixel buffer (if desired) at the top.

The graphics options offered for this display are straightforward but somewhat limited allowing one to draw a point, a line, a rectangle, and a circle with a small range of 4 line thicknesses and in the last 2 the ability to “fill” the object with a range of colours (that match the line colour if unfilled). It is not possible to have the line defining the object outline in one colour and the fill of the object in another colour.

The colour, of text or drawn objects, can be set using one of fourteen pre-set, named colours OR using a slider (that provides 65k colours). Any of these objects, or text or numbers, can be overlapping.
Even with text, however, the display “refresh” rate is not rapid. For instance, if one changes the colour of a letter on the bottom line it takes more than a second for the letter to change colour after the code refreshes the screen. This delay isn’t affected by order of the code either. If one codes for changing a letter at the bottom to a new colour and then one at the top to a new colour (in that order) the screen refresh changes the colour from top to bottom irrespective of the order it is coded in if one is only using a single “Send Display Data” command at the bottom. One can force a sequence by adding a “Send Display Command” after each text colour change but that consequently makes the entire process take longer because the entire display is refreshed each time. For some users (particularly in games) the slow speed of refresh might be a problem, but for others (such as numeric output from sensors with a lower sampling time) it likely will not be. There is, however, a code block that can somewhat address this “speed of refresh” issue.

The whole display “refresh” option -- enacted by the “Send Display Data” code block -- occurs in a visible fashion and with either text or graphics the user can actually watch the refresh roll down the screen. However, there is a code block – “Show windows display data” - that allows you to refresh only part of the display. In practice, this code block with a designated X/Y starts/stop causes significantly faster screen refreshes than the code block “Send Display Data” for both graphics and text changes on the display because it refreshes only the designated part of the whole screen. This can make updating an area of text or graphics much quicker.

The Waveshare display uses a number of the available GPIO pins, which can also be accessed either through the adapter or by wiring the micro:bit to PCB access points, but others (pins 0, 3, 4, 5, 6, 7, 9, 10, 11, 12, 19, 20) are still available if an adapter board is used to give access to them. Note that the 5-X-5 LED matrix on the micro:bit can also be used in conjunction with the Waveshare display.

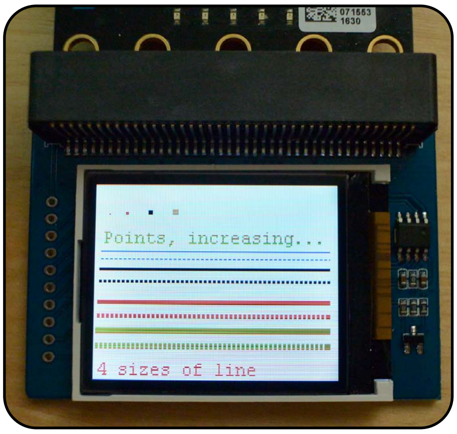
In conclusion, The Waveshare LCD display is a handy piece of kit that made some aspects of using the micro:bit more useful (particularly since I work with science teachers and I am interested in how the micro:bit can be used to collect data in classroom science activities. To get around the shortage of available pins (and access to them) I use one micro:bit to collect the data and then send it over the “radio” to another micro:bit with the Waveshare display attached. The slow refresh rate, even with the block-specified refresh area, would limit the display for many uses but there are few other options allowing graphics available for the micro:bit right now and this display is reasonably straightforward to use and is affordably priced.
The Waveshare 1.8” LCD display for the micro:bit can be purchased from the manufacturer at https://www.waveshare.com/1.8inch-lcd-for-micro-bit.htm for $15.99US (+ shipping), but it is also available from other distributors. My order (to Canada) took about one month to arrive.

OUR RATING

8/10

This is a preview!

Our new website format will allow us to do more in-depth reviews like this one. We hope you like this preview as to what these new reviews look like!



MAJOR CHANGES FOR MICRO:MAG

After nearly three years publishing a PDF format magazine, it's time to change how micro:mag works. By **Josh & Kerry**.

When the idea of micro:mag was born back in March of 2018, we never thought we'd get to where we are today. We wanted to create a community magazine for the micro:bit which gave members of the community a place to showcase their projects and share their work and we like to think we have succeeded with that goal.

Our intention was to style micro:mag as a normal magazine but publish it in a free PDF format. It's fair to say, this has brought along many challenges for us. When we started the magazine, we had no funds, so we used free software that allowed us to create Issues 1-4. After this, the magazine really started to take off which enabled us to upgrade our software to Adobe InDesign and the Creative Cloud suite. This enabled a better layout and also the print copies we have been producing for the last year. However, we've been reflecting on our format and taking feedback from our readers on what we can do to improve what we offer. We've come to a conclusion as to what the future of micro:mag looks like.

The COVID-19 pandemic has taught us that virtual content is the future. The current print format that we produce takes up 80% of our time that we could be spending on other things like creating more interactive content and other types of media for micro:mag. Therefore, we're happy to announce that micro:mag is transitioning into an online only website format.

Let us explain what we mean by this. Going forward, we'll be publishing the same content that we already do now but instead of putting it in a print format every few months, we'll be posting articles on a new website that we have built. Then, at the end of the month we'll be publishing a monthly round up newletter linking to our favourite posts from that month.

What this means is, micro:mag will be easier to create for us and it will be a lot easier for you, our readers, to read, via a mobile optimised website. This also means that we can publish regular weekly content for you to read and enjoy so you're always up to date with what's going on in the world of micro:bit.

We feel that this is a better future for the magazine and will help it to grow even further. We want micro:mag to become **the** place people go to for regular news, makes and reviews from the micro:bit community and we can't wait for the future.

You'll likely have lots of questions about this move and we've already spoken to a few people about our plans and they're just as excited as we are for the future! Hopefully we can answer some of the most popular questions below!

FAQs

Will you still be releasing PDF format content?

This is unfortunately the end of our PDF issues, however, we have plans to create maybe one or two print publications a year. Stay tuned for those! Each article published on the new website will be print friendly so you'll be able to download a PDF file and print that out if you'd like! We'll also be creating a round up email at the end of each month.

Can I still submit an article?

Yes! Nothing has changed about the way we do things other than the format we publish micro:mag. We still need your articles and we encourage you to visit our website to find out how to submit an article to us.

What happens to previous magazines?

Don't worry, we'll still be keeping our back catalogue free to download for everyone. They're going nowhere and we're proud of what we've created over the past few years!

What happens to print copies?

As we will no longer be producing the PDF magazine, we will stop production of our print magazines also. We've really enjoyed being able to print the magazine and send it out all over the world. However, we feel that a virtual future is the way forward for micro:mag!

I bought a print subscription. What happens now?

Because we are stopping the production of print and PDF magazines, print subscriptions will stop also. You should have received an email this week via the email address you used to buy your subscription on refund details for the issues we haven't sent out to you yet.

Will the content be free?

Absolutely! All articles will be free on the new website and nothing will be put behind a paywall. We are looking into membership options for the website and we'll be launching these shortly. Memberships will include early access to content a few days before they go public.

Will there be ads on the new website?

There will be ads on the new site however only a few spots will be available and these will all be custom ads that will be micro:bit related and not annoying.

When is the new website launching?

We're still working on a few things, but our new website will be replacing our current website in September alongside new branding and a new logo.

What types of content will you be publishing?

A website format brings lots of new opportunites for us! We'll be publishing the same content written by the community (News, Features, Makes & Reviews) as well as interactive content with embeddable makecode editors alongside some new video content created by us and the micro:bit community. We can't wait to share some new and exciting content with you that has never been possible in our current format.

How often will content be published?

We're aiming for a similar setup that popular tech news websites have currently and we'll be be ramping up production to hopefully achieve daily articles soon. We're really excited about this new format as it allows us to get from the writing to publishing stage a lot quicker than we currently can with the PDF magazines.

How will the new website be funded?

We're lucky to have generous advertisers at the magazine and we hope to keep this the same as our main source of income. However, we also rely on community donations to fund the magazine and we expect costs to go up with this new format. If you'd like to contribute, head over to micromag.cc/donate to help us out!

I have more questions!

We'd love to answer your questions about the changes. You can contact us via email at hello@micromag.cc or via Twitter @micro_mag.

Closing Statement

We'd like to close out the final PDF issue of micro:mag by saying "Thank you" to the whole micro:bit community for reading, contributing, donating and supporting us along the way.

We're really excited about the future of micro:mag with our new format and we hope you'll join us for the journey ahead.

We can't wait to start putting out more of the same great content for you to enjoy and hopefully this will help micro:mag grow and make it a better magazine and resource for everyone. Whether you're a beginner or pro, we hope you'll stick with us and make use of the new content we'll be putting out. Make sure to watch on our Twitter page or subscribe to our mailing list to be the first to know when the new format micro:mag launches, you don't want to miss it!

micromag.cc

£5.99