

# Adventures with the



## CO<sub>2</sub> Dock

FOR MICRO:BIT

HARDWARE v1A



Instructions Document v1a

# TABLE OF CONTENTS

Introduction.....	2
CO2 and Health.....	3
Getting Started.....	4
Experiment 1. Displaying CO2 Readings.....	5
Experiment 2. CO2 Meter.....	7
Experiment 3. CO2 Alarm.....	8
Experiment 4. Data Logging to a File.....	9
Experiment 5. Data Logging over USB.....	12
MakeCode Extension.....	14
Troubleshooting.....	16
Learning.....	17
MonkMakes.....	18

## INTRODUCTION

The CO2 Dock is a true CO2 sensor, combined with temperature and relative humidity sensors designed for use with the BBC micro:bit.

The board will work with micro:bit version 1 and 2 boards.

This booklet includes five experiments complete with code in MakeCode blocks.

# CO<sub>2</sub> AND HEALTH

The level of CO<sub>2</sub> in the air we breathe has a direct influence on our well-being. CO<sub>2</sub> levels are of particular interest from a public health point of view as, to put it simply, they are a measure of how much we are breathing other people's air. We humans breathe out CO<sub>2</sub> and so, if several people are in a poorly ventilated room, the level of CO<sub>2</sub> will gradually increase. As will the viral aerosols that spread disease.

Another important impact of CO<sub>2</sub> levels is in cognitive function – how well you can think. The following quote is from the National Centre for Biotechnology Information in the USA:

*“at 1,000 ppm CO<sub>2</sub>, moderate and statistically significant decrements occurred in six of nine scales of decision-making performance. At 2,500 ppm, large and statistically significant reductions occurred in seven scales of decision-making performance”* Source: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3548274/>

The table below is based on information from <https://www.kane.co.uk/knowledge-centre/what-are-safe-levels-of-co-and-co2-in-rooms> and shows the levels at which CO<sub>2</sub> can become unhealthy.

Level of CO <sub>2</sub> (ppm)	Notes
250-400	Normal concentration in ambient air.
400-1000	Concentrations typical of occupied indoor spaces with good air exchange.
1000-2000	Complaints of drowsiness and poor air.
2000-5000	Headaches, sleepiness and stagnant, stale, stuffy air. Poor concentration, loss of attention, increased heart rate and slight nausea may also be present.
5000	Workplace exposure limit in most countries.
>40000	Exposure may lead to serious oxygen deprivation resulting in permanent brain damage, coma, even death.

# GETTING STARTED

## Connecting

The CO2 Dock receives its power from the BBC micro:bit. This will usually be via the micro:bit's USB connector.

Connecting a BBC micro:bit to the CO2 Dock is just a case of plugging the micro:bit into the CO2 Dock as shown below.



Note that the ring connectors at the bottom of the CO2 Dock are connected to the ring connectors of the micro:bit, allowing you to attach other things to your micro:bit.

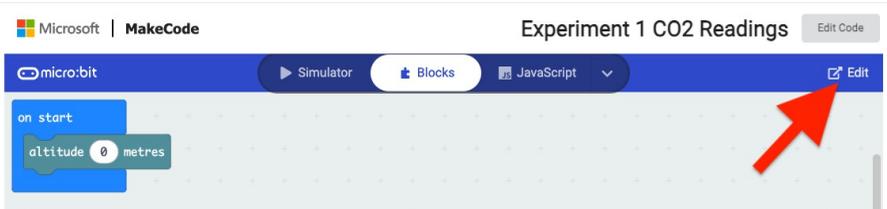
If the micro:bit is powered, then an orange LED in the CO2 Dock's MonkMakes logo will light to show that it is powered.

# EXPERIMENT 1. DISPLAYING CO2 READINGS

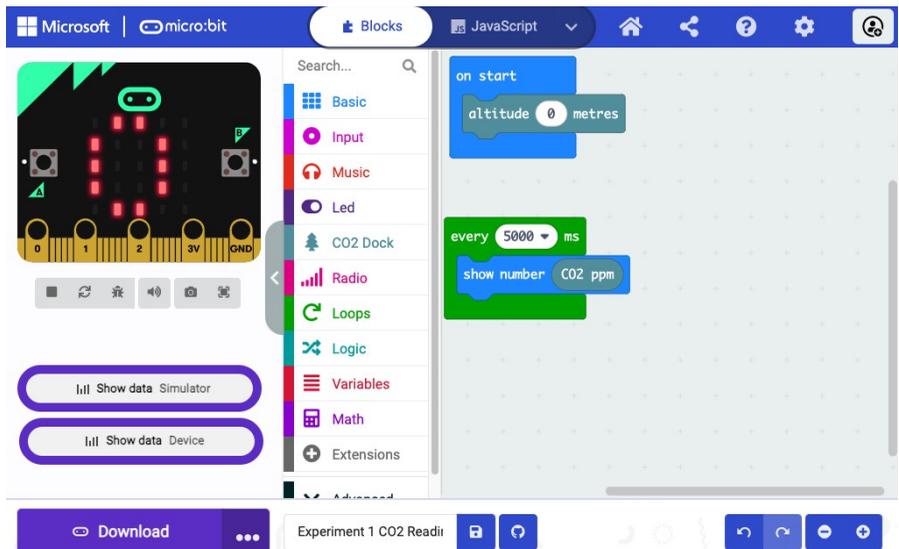
**MakeCode Link:** [https://makecode.microbit.org/\\_A3D9igc9rY3w](https://makecode.microbit.org/_A3D9igc9rY3w)

This program displays the CO2 reading in parts per million, refreshing every 5 seconds.

When you click on the code link at the top of the page, the MakeCode system will open a preview window that looks like this:



You can preview the program, but you can't change it or, more importantly, put it on your micro:bit, until you click on the **Edit** button indicated. This will open the normal MakeCode editor and you can then upload the program onto your micro:bit in the normal way.



When the program first starts, you may see unlikely readings of CO2 level. This is normal. The sensor used by the CO2 Dock takes a few minutes for readings to

stabilise.

Once the readings have stabilised, try breathing on the CO2 Dock to increase the CO2 readings. Note that it will take some time for the CO2 readings to increase, and even longer for them to fall back down to the room's CO2 level. That's because the air that finds its way into the sensor's chamber will take some time to mix with the air from outside the sensor.

The code is pretty simple. The **on start** block contains the block **altitude**. This block is useful if you live somewhere high up (more than 500 meters) then you should change the value from 0 to your height in meters, so that the sensor can compensate for the reduced atmospheric pressure that alters the CO2 measurement.

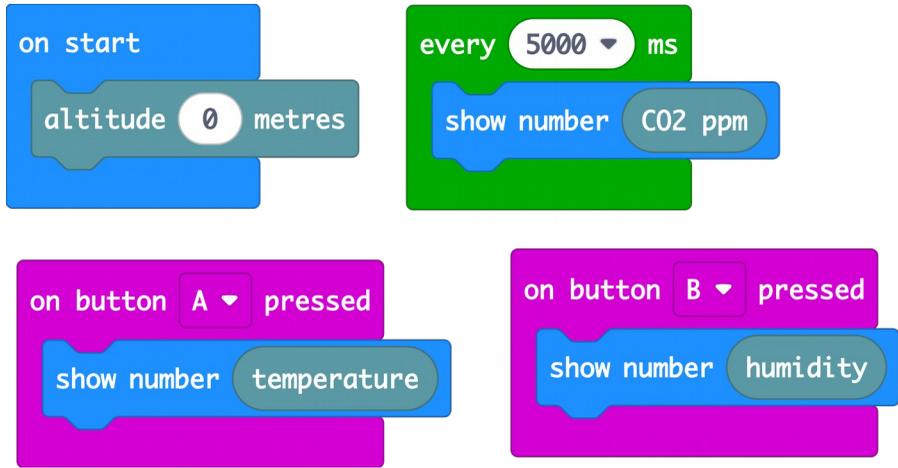
The **every 5000ms** block contains code that will be run every 5 seconds. You can find this useful **every** block in the **Loops** section of the blocks palette. This **every** block contains the **show number** block that takes the CO2 **ppm** block as its parameter to be scrolled across the micro:bit's display.

If you have any problems getting this to work, see the **Troubleshooting** section at the end of these instructions.

## EXPERIMENT 2. CO2 METER

**MakeCode Link:** [https://makecode.microbit.org/\\_9Y9Ka2AWjHMW](https://makecode.microbit.org/_9Y9Ka2AWjHMW)

This program builds on the first experiment so that, when button A is pressed, the temperature in degrees Celsius is displayed and, when button B is pressed the relative humidity is displayed as a percentage.



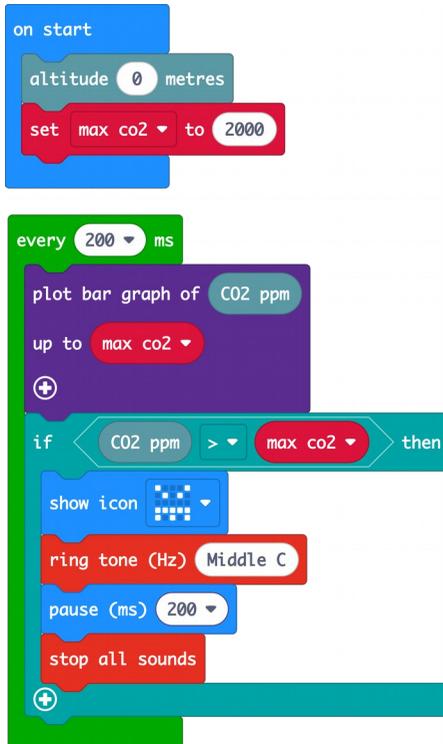
Install this program onto your micro:bit in the same way as you did in experiment 1, by using the code link at the top of this page.

When you press button A, the temperature in degrees C will be displayed once the current CO2 reading has finished displaying. Button B displays the relative humidity (how much moisture is in the air).

## EXPERIMENT 3. CO2 ALARM

**MakeCode Link:** [https://makecode.microbit.org/\\_EjARagcusVsu](https://makecode.microbit.org/_EjARagcusVsu)

This program displays the CO2 level as a bar graph on the micro:bit's display rather than as a number. Also, when the CO2 level exceeds a preset value, the display shows a warning symbol. If you have a micro:bit 2, or a speaker attached to P0 then the project will also beep when the CO2 threshold is exceeded.



```
on start
  altitude 0 metres
  set max co2 to 2000

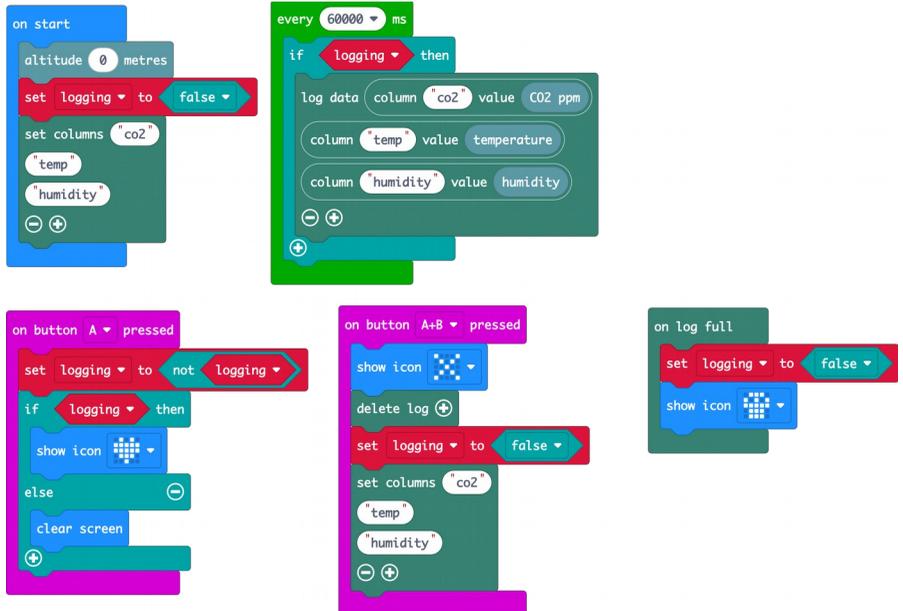
every 200 ms
  plot bar graph of CO2 ppm
  up to max co2
  +
  if CO2 ppm > max co2 then
    show icon [Warning symbol]
    ring tone (Hz) Middle C
    pause (ms) 200
    stop all sounds
  +
```

The code is written in a Scratch-like block-based language. It starts with an 'on start' block containing three sub-blocks: 'altitude 0 metres', 'set max co2 to 2000', and a plus sign. Below this is an 'every 200 ms' loop block. Inside the loop, there is a 'plot bar graph of CO2 ppm' block, followed by an 'up to max co2' block, and a plus sign. Then, there is an 'if CO2 ppm > max co2 then' block. Inside the 'if' block, there are four sub-blocks: 'show icon [Warning symbol]', 'ring tone (Hz) Middle C', 'pause (ms) 200', and 'stop all sounds'. The 'if' block is followed by a plus sign and a teal block that extends to the right, indicating the loop continues.

## EXPERIMENT 4. DATA LOGGING TO A FILE

**MakeCode Link:** [https://makecode.microbit.org/\\_YeuHE7R7zPdT](https://makecode.microbit.org/_YeuHE7R7zPdT)

This experiment will only work on a micro:bit version 2.



To use the program, press button A to start data logging – you will see a heart icon to show that all is well. Sampling is set to 60000 milliseconds (1 minute) – ideal for running the experiment overnight. But if you want to speed things up, change this value in the **every** block. Reducing the sampling time will mean that more data is collected and you will run out of memory sooner.

When you want to finish logging, press button A again. You can delete all the data by pressing buttons A and B at the same time.

If the micro:bit runs out of flash memory in which to store the data, it will stop logging and show the 'skull' icon.

The data is written into a file called MY\_DATA.HTM. If you go to the MICROBIT drive on your file system, you will see this file. The file is actually more than just the data, it also contains mechanisms for viewing the data. If you double-click on MY\_DATA.HTM, it will open in your browser and look something like this:

## micro:bit data log

[Download](#)[Copy](#)[Update data...](#)[Clear log...](#)[Visual preview](#)

This is the data on your micro:bit. To analyse it and create your own graphs, transfer it to your computer. You can copy and paste your data, or download it as a CSV file which you can import into a spreadsheet or graphing tool. [Learn more about micro:bit data logging.](#)

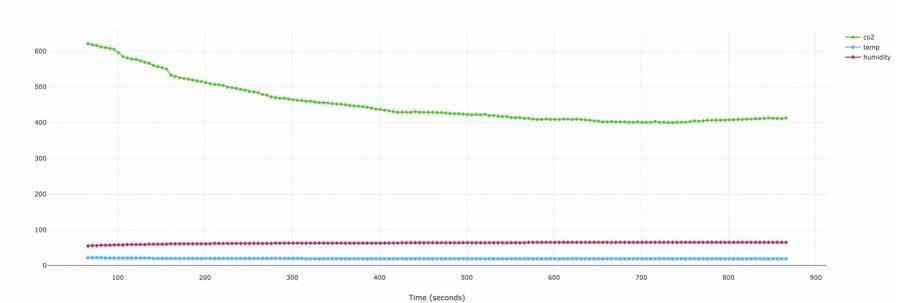
Time (seconds)	co2	temp	humidity
65.82	621	22	55
70.82	618	22	56
75.82	616	22	56
80.82	612	22	57
85.82	610	21	57
90.82	608	21	57
95.82	605	21	58
100.82	596	21	58
105.82	585	21	58
110.82	581	21	59
115.82	578	21	59
120.82	577	21	59
125.82	573	21	59
130.82	569	21	59

If you click on the Visual preview button, a simple plot of the data will be displayed.

## micro:bit data log

Download Copy Update data... Clear log... Close visual preview

This is a visual preview of the data on your micro:bit. To analyse it in more detail or create your own graphs, transfer it to your computer. You can copy and paste your data, or download it as a CSV file which you can import into a spreadsheet or graphing tool. [Learn more about micro:bit data logging.](#)



This project only works on version 2 of the micro:bit because it uses the *Data Logger* extension, that is itself only compatible with the micro:bit 2.

The Data Logger extension has a **set columns** block that allows you to name the columns of data that you are recording. When you want to write a row of data to the table, you use the **log data** block.

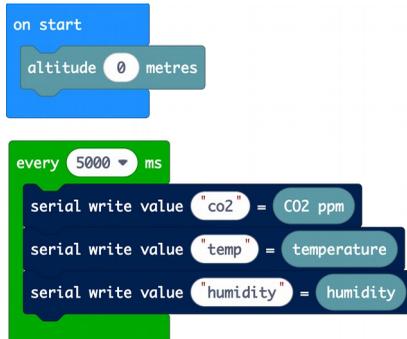
The Data Logger extension also has a **on log full** block that will run the commands inside it should the micro:bit run out of space to store the readings.

# EXPERIMENT 5. DATA LOGGING OVER USB

**MakeCode Link:** [https://makecode.microbit.org/\\_fKt67H1jwEKj](https://makecode.microbit.org/_fKt67H1jwEKj)

This project only works on micro:bit version 2 and works best using the Google Chrome browser. Even so, you may find that the web USB feature of Chrome does not always work reliably. This is also a project, where the micro:bit must be attached to your computer with a USB lead.

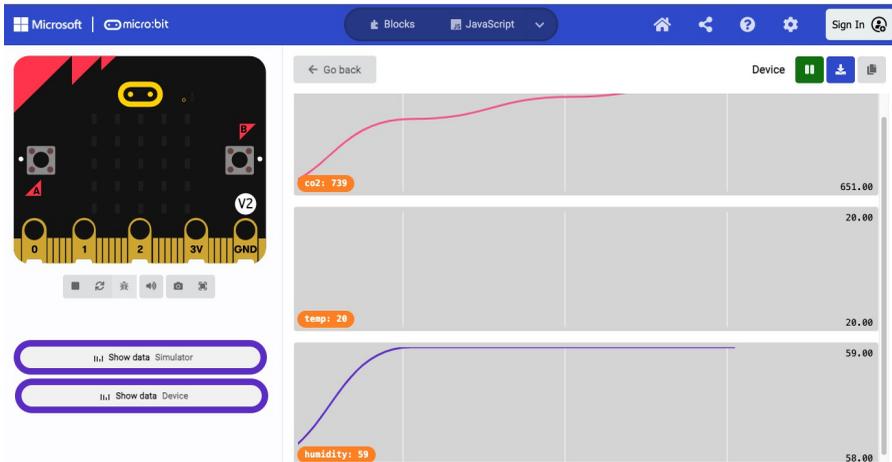
Instead of logging data to a file, like we did in Experiment 5, you will be logging data to your computer in real-time over the USB connection.



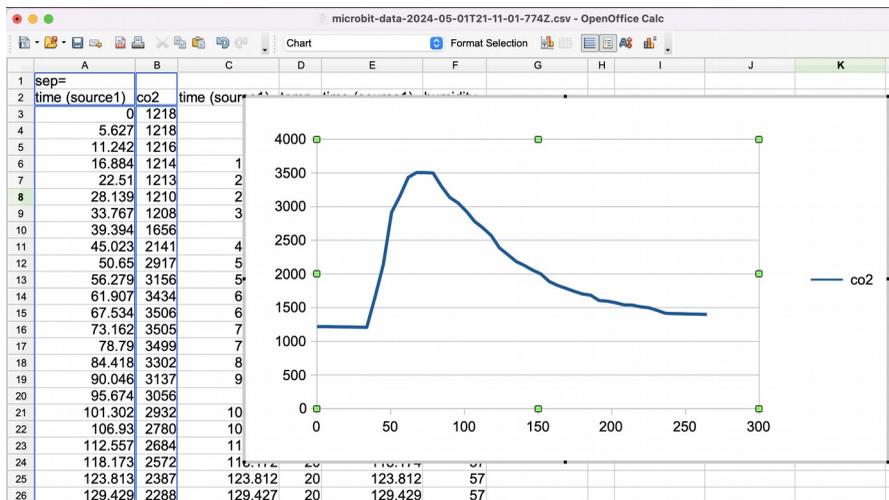
```
on start
  altitude 0 metres

every 5000 ms
  serial write value "co2" = CO2 ppm
  serial write value "temp" = temperature
  serial write value "humidity" = humidity
```

Once the program is uploaded, using a paired micro:bit, click on the **Show data Device** button and you will see something like this.



Having captured the data, you can then click on the blue *download* icon to save it as a CSV file that can be imported into a spreadsheet, where you can plot charts.



Because the three readings are actually logged at slightly different times, there will be a separate time column, in the CSV file, for each reading type. When creating a chart, just pick one of the time columns for the x-axis – it doesn't matter which.

This project uses the **serial write value** block that you will find in the Serial category of blocks. This sends the reading over the USB connection to the makecode editor running in your computer's browser.

# MAKECODE EXTENSION

The CO2 Dock uses a MakeCode extension to provide a set of blocks to make programming simple. The previous example programs already have the extension installed but, if you are starting a new project, you will need to install the extension.

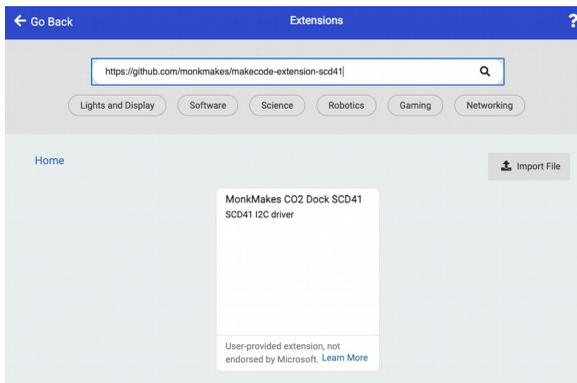
To do this:

\* Go to the MakeCode for micro:bit website here: <https://MakeCode.microbit.org/>

\* Click on **+ New Project** to create a new MakeCode project – give it whatever name you like

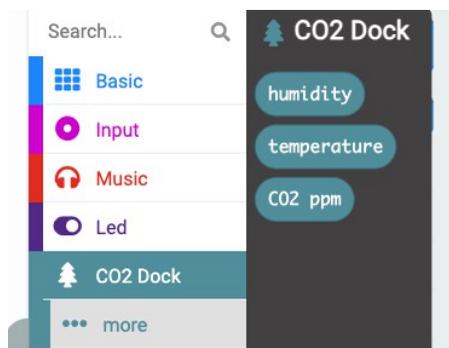
\* Click on the **+ Extension** and in the Search area paste the following web address: <https://github.com/monkmakes/makecode-extension-scd41>

This should bring up a single search result.

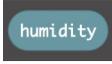
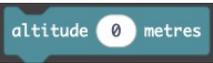


\* Click on the MonkMakes CO2 Dock extension and it will be installed.

\* Click on **← Go Back** and you will find that some new blocks have been added to your list of blocks under the category CO2 Dock.



## Blocks Description

Block	Description
	This block returns the CO2 concentration detected by the sensor in ppm (parts per million). Fresh air is usually about 400ppm. This sensor can measure concentrations as high as 5000ppm.
	This block returns the temperature in degrees Celsius.
	This block returns the relative humidity as a percentage.
<b>Advanced Blocks</b>	
	This temporarily suspends the CO2 Dock from making readings. This is rarely needed, but could be used to minimise power consumption in battery-powered projects.
	After using pause CO2 Dock, this block restarts the reading process.
	The NDIR (nondispersive infrared) sensor used in the CO2 Dock will give inaccurate results if you live at any significant altitude (more than a few hundred metres). For more accurate results use this block in your <b>on start</b> block. The default altitude is 0m.
	This block will remove any altitude or manual calibration settings, to return the sensor to it's original condition. Note 1.

Note 1. Use of this block very gradually erodes the sensor's EEPROM (2000 writes), so this block is limited to one call between resets.

## TROUBLESHOOTING

**Problem:** The amber power LED on the CO2 Dock for micro:bit is not lit.

**Solution:** Make sure that your micro:bit itself is receiving power. If your project is battery powered, try fresh batteries.

**Problem:** When I first run my program, the CO2 readings seem wrong, sometimes 0 or a very high number.

**Solution:** This is normal. The sensor takes some time to settle. Disregard any readings for the first few minutes after the sensor starts up.

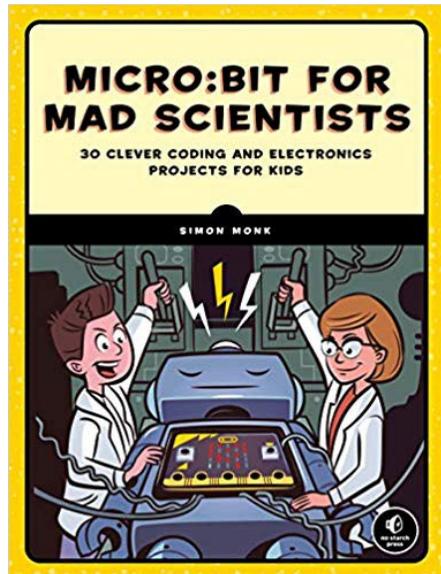
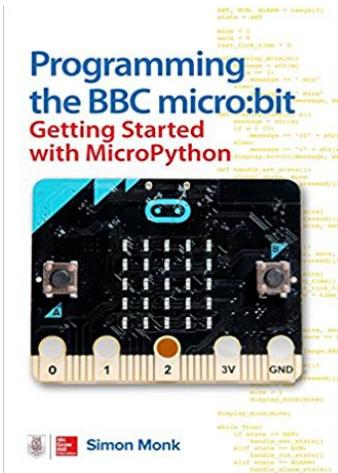
# LEARNING

## micro:bit Programming

If you want to learn more about programming the micro:bit in MicroPython, then you should consider buying Simon Monk's book 'Programming micro:bit: Getting Started with MicroPython', which is available from all major book sellers.

For some interesting project ideas, you might also like micro:bit for the Mad Scientist from NoStarch Press.

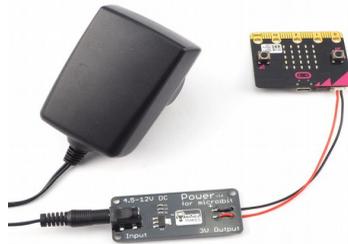
You can find out more about books by Simon Monk (the designer of this kit) at: <https://simonmonk.org> or follow him on X where he is @simonmonk2



# MONKMAKES

For more information on this kit, the product's home page is here:  
[https://monkmakes.com/co2\\_mini](https://monkmakes.com/co2_mini)

As well as this kit, MonkMakes makes all sorts of kits and gadgets to help with your maker projects. Find out more, as well as where to buy here:  
<https://monkmakes.com> you can also follow MonkMakes on X @monkmakes.



From left to right: Solar Experimenters Kit for micro:bit, Power for micro:bit (AC adapter not included), Electronics Kit 2 for micro:bit and 7 Segment for micro:bit.