[Chart, line chart

Description automatically generated](https://morethanlearning21.wordpress.com/2021/08/04/microbit-and-science/)**Datalogging with the micro:bit  
(V1 and V2)**

**Stuart Ball**

**Using makecode blocks**

**Key Stage 2 – Key Stage 3**

**Logo

Description automatically generated**

**The Micro:bit and Science- Data logging**

A picture containing text

Description automatically generated

The micro:bit has been written about a lot . If you still don’t know what it is look no further than [www.microbit.org](http://microbit.org/).

I have been lucky enough to be involved with this educational microcontroller from it’s inception to the delivery to over a million students and beyond.  
For me it’s the perfect device to teach computing. This is my goto when asked how to teach computing across the curriculum. Over the past few years I have had the pleasure and opportunity to work with teachers and students around the world and introduce them to the world of physical computing. using the micro:bit

I am always looking at doing a lot, in a simple way with the micro:bit, aimed at the teacher who has absolute no previous knowledge of computing, who by learning some simple processes, learns to deliver a lot of learning to their students. So this booklet is based on a blog series on [www.computingatswchool.org.uk](http://www.computingatswchool.org.uk) and hopefully provide useful guide to using the micro:bit as a datalogger in the classroom

**Contents**

* The [micro:bit & Science](https://www.computingatschool.org.uk/news-and-blogs/2022/january/microbit-and-science)
* [Logging the data - basics](https://www.computingatschool.org.uk/news-and-blogs/2022/january/logging-the-data-basics)
* [Using code to make a simple micro:bit datalogger](https://www.computingatschool.org.uk/news-and-blogs/2022/february/using-code-to-make-a-simple-microbit-datalogger)
* [Datalogging with the micro:bit using variables and Algolrithm](https://www.computingatschool.org.uk/news-and-blogs/2022/april/variables-and-algorithms-on-the-microbit-for-datalogging)s
* [Visualising logged data with the micro:bit](https://www.computingatschool.org.uk/news-and-blogs/2022/april/visualizing-logged-data-with-a-microbit)
* [Datalogging with micro:bit V2 - part 1](https://www.computingatschool.org.uk/news-and-blogs/2022/may/datalogging-with-microbit-v2-part-1)
* [Datalogging with micro:bit V2 - part 2](https://www.computingatschool.org.uk/news-and-blogs/2022/may/datalogging-with-microbit-v2-part-2)
* [Datalogging with micro:bit V2 - part 3](https://www.computingatschool.org.uk/news-and-blogs/2022/may/datalogging-with-microbit-v2-part-3)
* [Using the micro:bit's radio functionality for datalogging](https://www.computingatschool.org.uk/news-and-blogs/news-and-blogs/2022/may/using-the-microbits-radio-functionality-for-datalogging)
* Examples of makecode programs used
* Hex files can be download at <https://www.computingatschool.org.uk/resource-library/2022/october/datalogging-with-the-microbit-hex-files>

# The micro:bit and Science

## 

[Chart, line chart

Description automatically generated](https://morethanlearning21.wordpress.com/2021/08/04/microbit-and-science/)

Creating your own data logger with micro:bit is really easy . It only takes a few lines of code to create a simple temperature measurer. But the micro:bit has wealth of features that allow learners to create code and use their own datalogging devices. Why should they do this, I believe that it gives them a greater understanding of not only the value of programming but understanding the scientific concepts that are also being studied. The section on these resources start with simple ideas that can be built upon and develop an increasing the complexity and learning curve as we go. But, here I am starting at the very basic level. For teachers who have only their micro:bits, no additional equipment and don’t consider themselves confident in coding and computing.

The micro:bit has a number of built in sensors that can measure the following:-  
– Temperature  
– Light  
– Acceleration  
– Magnetism  
If you are using the micro:bit V2, it has a sound sensor (microphone).

Temperature and light are the simplest and most useful to start with I find.

Let’s start with the MakeCode website ( I am assuming that you are familiar with the use of the MakeCode Editor – [http://makecode.microbit.org](http://makecode.microbit.org/) . If not check out tutorials on the MakeCode home page.

To create a simple temperature reading device , that a digital thermometer

Graphical user interface, text, application, chat or text message

Description automatically generated

Simply, place a Show number block inside the Forever block and drag from the Input commands the Temperature/Light block.

The micro:bit will now continually scroll the current temperature and light level. Students have now created their own digital measuring device.

Try this with the light or with the micro:bit V2, sound block

# Logging the Data – basics

## Now that students have their micro:bit logging device we need to set them a task to investigate. But, before that, students need to understand the measurements they are collecting. This section looks at how the micro:bit simulator can be used to understand the measurements the micro:bit are taking. This knowledge can be used in future projects when building control devices such as alarms or automatic light.

When using the light sensor the micro:bit simulator changes to show the default light measurement which is 128 and a yellow half circle. Click on the circle and drag up , the amount that is coloured yellow will change. Drag until there is no yellow left. The display show 0. This represents no light. Drag downwards and fill the circle with full yellow. The display will show 255, Full light.

* Graphical user interface

  Description automatically generated
* Graphical user interface

  Description automatically generated with medium confidence
* A picture containing graphical user interface

  Description automatically generated

micro:bit simulator showing light readings

So now we have a minimum value , 0, a maximum value 255 and a mid point value 128. You may want to ask students what is the mid point of 255? 255 divided by 2 equals 127.5. So why is the micro:bit showing 128 as the mid point? It is because in computing , 0 is a value, a number. The micro:bit is showing 0-255, that means it is dealing with 256 values, mid point of which is 128.

The temperature sensor works in a much more straight forward way. Again the simulator will show a virtual thermometer. Which can also be changed by dragging up and down. It will show the maximum value of 50C and a minimum -5C.

* A picture containing graphical user interface

  Description automatically generated
* A picture containing qr code

  Description automatically generated
* A picture containing graphical user interface

  Description automatically generated

micro:bit simulator showing temperature readings

You may want to ask students if they think the temperature reading is an accurate measurement and how could they check this? The micro:bit will generate its own small amount of heat. This will affect the accuracy of the readings. Students could check how much of an affect, by using a thermometer next to the micro:bit and they could compare the difference. They could then subtract this from the micro:bit reading in their recordings.

Now students have a simple device to record light or temperature. They can undertake different science investigations. Such as the warmest or coldest places around the school, or are different colour surfaces different temperatures or my favourite,, which is the best place for the Headteacher to their book, where the light is not too bright or too dark

This section shows how to take simple measurements with the micro:bit and is a great way to using computing in science lessons. The next post will develop this, students can use code to improve the accuracy, as in the case of the thermometer and the functionality of their micro:bit measuring device.

**Using code to make a simple Micro:bit Datalogger**

The previous sections explored how to use the built-in sensors on the micro:bit to make and display simple measurements such a light and temperature.

Graphical user interface, text, application, chat or text message

Description automatically generated

This post explores how students can use code to improve their data logger. They will be able to control when to record and display the sensor reading. This makes it easier to record sensor readings in a notebook, table, or spreadsheet.

Using the code commands students will be able to press a button to record and store a measurement, another button to display the measurement, and another to reset the measurement ready to take another.

This is an opportunity (if it has not already been explored), to discuss with students how the micro:bit with only two input buttons can have three inputs. Button A and Button B and pressing them both as A and B forms a third input. Also discuss with the practicality of using the Shake command as an input in this project. Is it a good idea?  
In Makecode create the following programme.

Graphical user interface, application

Description automatically generated

For this students will need to create a variable. This a ‘place’ where values can be stored, manipulated and recalled. Create a variable called light ( use a name that is recognisable) . In the Makecode command list, click Variables, then Make a Variable. , name the variable, this will create three new blocks.

Graphical user interface

Description automatically generated

When the micro:bit is powered up the variable is set to zero.

When button A is pressed, the sensor is read and assigned to the variable’s value. This will not change, even if the sensor reading changes unless button A is pressed again. Add a show icon block to indicate the data has been captured.

Pressing button B displays the captured sensor reading. Students can now easily record this information on a spreadsheet or table.

Pressing A+B sets the variable value back to zero ready for the next reading.

The next post shows how students can automate processes using variables and algorithms in their code.

**Datalogging with the micro:bit using Variables & Algorithms**

The last section looked at how a variable can be used to store and display sensor data on a microbit. This section looks at how you can extend that concept, by using a simply algorithm to manipulate the data collected.

An algorithm is a sequence of instructions that are followed to complete a task. In this case the micro bit will automatically convert a temperature reading from Centigrade to Fahrenheit using a simple formula.

First create the code to capture a temperature reading from the microbit's temperature sensor. This is always a value in Centigrade. Store this reading as a value in a variable. Use button A to log this reading Use button B input to display the reading.

Add an show icon block to the on button A event block to show that the reading has been captured.

Graphical user interface, application

Description automatically generated

Next when both A and B buttons are pressed the microbit will convert the temperature value into a Fahrenheit value.  
To convert any centigrade value in fahrenheit we use some simple maths. Simply multiply the value in centigrade by 1.8 and add 32. We are going to use some code to automatically do this.

Create a new variable, call this Fahrenheit. Add a Set Fahrenheit block to 0 to the on Start event block. Next add an On button pressed block and change this to A+B. Now add a set Fahrenheit to block to the On Button A+B pressed block. Now go to the Math command list and drag a multiplication command block into the space after the to . Got back to the Variable command list, find the temperature variable and drag this to replace the first zero in the multiplication block. Change the second zero in the block to 1.8. Drag another Set Fahrenheit to block, this time drag an addition block from the Math commands. Now drag the fahrenheit to replace the first zero. Change the second zero to 32.

Graphical user interface, application

Description automatically generated

Now when both buttons A+B are pressed the microbit will multiply the value stored in the variable temp by 1.8 and store this in the new variable fahrenheit. Next this value has 32 added to it and stored in the fahrenheit variable. The value is then displayed as the captured temperature as measured on the farhenheit scale. Pressing the B button will display the captured value in centigrade, as this value has not been changed and is still stored in the temp variable. It will only changed when a new reading is taken.

This activity is a great learning opportunity for students

**Visualizing logged data with a micro:bit**

So far these posts have looked at how to measure external factors such as light, temperature and sound (with micro:bit V2). The LEDs have been able to give a numerical value that students can record, then enter into a graphing application such as Excel or Sheets. This process is good practice for students in collecting and recording data and great investigative science, especially for younger students.

The real power of data logging is when the device not only takes measurements, but stores and displays them in real time for instant analysis. Amazingly the micro:bit can do this with two lines of code. The micro:bit V2 has some specialised data logging that offer additional functionality , I will discuss that in a future post. This method will work on both versions.

Firstly you need to connect the micro:bit so that you can directly flash data to it. Details how to do this can be found here – <https://youtu.be/qSjMDG84bMY>

Next …

Graphical user interface, text, application, chat or text message

Description automatically generated

The plot bar graph block can be found under the LED commands, drag into it the input from a sensor you want to measure. In this case light, enter 255. This represents the maximum value for light, and 50 for temperature. The command then works out the number of LEDs to illuminate to illustrate the proportion of the level value. So a value of 0, dark, shows 1 LED, a maximum value 255, full light is illustrated by all 25 leds being illuminated. Values in between are represented proportionally.

But this command hides a fantastic secret. A clue is given by the ‘Show Console button’. Click this and the simulator will change. By altering the value of the light, by click and dragging over the yellow circle. You will see not only the number of LEDs illuminating changing, but a line trace, plotted in real time. The values are also listed below.

Graphical user interface, application, Teams

Description automatically generated

Download this code to a micro:bit via USB. Keep the micro:bit connected to the computer. An extra button will appear, labelled Show Console …. Device. Click this. The screen will now show the data from the sensor on the micro:bit. Try covering and uncovering the LEDs on the micro:bit and see the changes in the line being plotted.

Graphical user interface, application

Description automatically generated

Using the pause button will pause the stream of data so that the line graph can be analysed. The other feature is the download button. This will allow you to download all the data over period measured as a CSV file.

Icon

Description automatically generated

This can then be imported in to your preferred spreadsheet application.

**Datalogging with micro:bit V2 - part 1**

In October 2020, four years after the release of the micro:bit V1, the micro:bit V2 was released. It has a wealth of updated features including a small speaker and microphone. Full details can be founded here - <https://microbit.org/new-microbit/>

Diagram, schematic

Description automatically generated

The new features that support datalogging, not only include a new in built sensor, a microphone, for recording sound levels, but changes to other components the ability for the micro:bit to record and store data. To support this new commands have been created in the Makecode Editor.

Before we can start using these new features for datalogging, we need to make some simple changes to the makecode editor. These features are still being tested, so they are not available by default. This what I have been able to discover and get working. To use the sound sensor (microphone), the commands are featured by default and can be found in the Input command list, when the microphone is active, a small red LED will light up. Try replacing light or temperature in the previous examples with sound level. This will also work in the previous post's example where data collected was displayed in real time. Use the value of 255 for sound level

To utilise the V2's ability to record and store data we need to change the makecode editor to the beta experimental version. This is easily done by. Click on the 'COG' icon on the upper right of the browser window. From the menu that appears click on 'about'. Next click 'experiments'. From the experiments that appear click on 'beta editor'. You can just type in this URL -<https://makecode.microbit.org/beta> ,

Graphical user interface, text, application

Description automatically generated

but it's worth having a look at the other experimental features that might be added to Makecode in the future, the accessible blocks for example might be useful for some students.

Now that you are using the beta version of the editor, You will need to use this version every time when using the datalogging features on the V2. We also need to load into the commands specifically designed for data-logging and give the microbit the ability to flash data to it's memory. Click Advanced in the Editor command list, next click Extensions, choose datalogger (if you can't see it use the search function at the top of the screen). You will see a new set commands called Data Logger added to the command list. When you save a program made with these commands this extension will automatically be loaded. Just note that if you try and use this setup with a V1 you will get an error. It is a good idea to start a new project if you are going to code again with a micro:bit V1.

**Datalogging with micro:bit V2 - part 2**

[In the previous post](https://www.computingatschool.org.uk/2022/may/datalogging-with-microbit-v2-part-1)we looked at the new features of micro:bit V2, and how to set up the makecode environment to code it.

The code below sets up the microbit V2 to record the sound level and store the data.  
A screenshot of a computer

Description automatically generated with medium confidence

Here we setup an array with a single column. Every time the sensor takes a reading, the value will be placed on a different row in that column.

In an On Start event code block, drag and drop a Set columns block. Reduce the number of columns by clicking the – (minus sign). Replace the letter “a” with the words Sound Level (something that describes the value). The next three commands show a tick icon for 2 secs. This is just to show that the code is running.

The next event block sets up by taking a reading every second (1000ms). This value could be changed to any time interval you like but must be in ms.

The next command block - log data, takes a reading from the sound sensor and stores it in the array. But, unlike previous examples, where we only used variables, when a new value is recorded it would overwrite the previous value, in an array the new value is recorded on a new row in the column of the array that was set up.

So every second a new value will be added to the array until the microbit can no longer store any more data.  The microbit V2 can record a total of 11000 entries, divided amongst the number of columns.

The next block is optional, mirror to serial allows the data to be sent to a computer and the data viewed in real-time. - [Visualizing logged data with a microbit](https://www.computingatschool.org.uk/news-and-blogs/2022/april/visualizing-logged-data-with-a-microbit)

The show Leds is just used to indicate that the microbit is logging the data.

On log full - indicates when the log is full and can no longer store any data.

On Button pressed, calls the command delete Log, which deletes the data stored in the array. You need to do this as the array doesn't reset every time you use it, data is just added to the end of the last set of data collected. The pause is added to indicate to the user that they need to wait for this process to happen.

Download this code to your micro:bit and you are now ready to begin log changes in the sound levels.

**Datalogging with micro:bit V2 - part 3**

This section looks at how to access and visualise the logged data.

This command instructs the microbit to record a reading from the sound sensor and stores a value in an array every one second.

Graphical user interface, application

Description automatically generated with medium confidence

That data is saved directly to the microbit in a file called MY\_DATA.HTM.

To access this file Open your File Manger on your computer, find the microbit and double click on it , open like you would a flash drive. You will see the MY\_DATA.HTM listed.

Graphical user interface

Description automatically generated

Double click on this and it will open in your browser. You should see something like this.

Graphical user interface, text

Description automatically generated

NOTE: If you are not seeing this , then check that you have created and downloaded the code to the microbit using the BETA version of the editor.  
<https://makecode.microbit.org/beta>

This page allows you to download all the data as a CSV file . This is very useful as the data can be accessed later and used in the spreadsheet of your choice. Copy allows the data to be copied and pasted into a spreadsheet, really useful if you are collecting daily data and are adding it to a culumunitive data collection.

This data is not a live stream, so Update Data, adds the data that has been captured whilst looking at the file.

Clear log, does exactly that, and prepares it for the next collection.

Visual Preview, is where the magic happens, clicking this will produce an interactive visualisation of your data.

Chart, line chart

Description automatically generated

Clicking on a data point will give you information about that point. So pupils will be able to identify the highest an the lowest values.

Clicking on the camera in the top right allows pupils and teachers to capture the plote graph that can be printed out or inserted in a document as a recrord of activity.

Now you are able to record and log sensor data on a microbit, try experimenting with recording more that one sensor reading , buy adding another column to the array command.

**Using the micro:bit's Radio functionality for Datalogging**

So far, we have used a single micro:bit to log, display and record data. The last section looked  at how the micro:bit V2 can be used to log data remotely. The same can be achieved with micro:bit V1 , by using two microbits , connected via radio and transmitting data between them.

Firstly you will need two micro:bits, one will act as the transmitter of the data, collecting the data via it's sensors. The second will be the receiver, receiving the data and sending it to a computer for  real time display and storage into a  CSV file.

**Setting up the transmitter**

The first thing to do is code a micro:bit to act as the transmitter, this micro:bit will collect the data and send it to the other micro:bit using radio signals.

Graphical user interface, application

Description automatically generated

In the on start command block add the radio set group command found in the Radio commands list. You need to set this to a number that is the same for both the reciever and transmitter. This is because If you haven’t set a radio group for the micro:bit, it will use one selected randomly. To be certain that your program will send or receive data using the same radio group, you will need to first choose and set a radio group for your program if you want it to work between different micro:bits. Also in a classroom situation, if students do not set a group number for their own microbits, data will sent to all microbits, chaos will ensue. Students can choose a group number from 0-255.

This is a great example to get students to think about networking and how computers communication with one another. You can find out a lot more with this great free resource from Nominet ,[Networking with the micro:bit.](https://www.nominet.uk/networking-with-the-microbit/)

The other commands here are just to show the code is functioning.

The next command event takes a light and temperature reading every minute and sends the value as a data packet via radio. You could use a forever loop here, but I found that I got errors in the data as was trying to read and send the data too quickly. With measurements like light and temperature, they will change slowly. So a measurement every minute (or longer) is adequate. Also, it will save on battery power. The radio send value command captures the value generated by the sensor, and attaches a label to it. which can be used to identify the value on the micro:bit receiving the data. Students may want to an led flash image here and a clear screen to indicate when data is sent.

You may want students to increase the transmission range. You can do this by adding the radio transmit power command, to the onstart block, the default value is 6 , and the maximum is 7. This could increase the range between the two micro:bits to 70m. This command can be found in Radio > more

**Setting up the receiver**

On the second micro:bit set up the code to receive the data.

A picture containing graphical user interface

Description automatically generated

Connect this micro:bit to your computer via USB cable. See this blog post for details

The on start command block is as before. Notice the radio set group command has the same number as the transmitting micro:bit.

The on radio received command waits for a radio signal. When data is received it sends the data via the USB cable to the computer.

To view and record the data, make sure the receiver micro:bit is attached to the computer. Makecode is open. Click on Show Console Device, which should appear under the micro:bit simulator. (Press the reset button on the back of the micro:bit if it isn't). You should now see the data being logged. Once you have finished logging the data. Download the data as a CSV file and graph in a spreadsheet application (make sure the computer does not go to sleep whilst this takes place otherwise you will lose your data)

**Code examples used in this resource**

The hex files of these files can be downloaded at <https://www.computingatschool.org.uk/resource-library/2022/october/datalogging-with-the-microbit-hex-files>

**Section 1 and 2**

Reading the light sensor

Graphical user interface

Description automatically generated with medium confidence

Reading the temperature sensor

Graphical user interface, text, application, chat or text message

Description automatically generated

Reading the sound sensor (v2 only)  
A picture containing graphical user interface

Description automatically generated

**Section 3**

Capturing the data

Graphical user interface, application

Description automatically generated

**Section 4**

Using an algorithm to convert data

Graphical user interface, application

Description automatically generated

**Section 5**

Send data over a serial connection

Graphical user interface, text, application, chat or text message

Description automatically generated

**Section 6**

Logging data with the micro:bit V2

A screenshot of a computer

Description automatically generated with medium confidence

**Section 8**

Transmitting data between micro:bits

Transmitter Code

Graphical user interface, application

Description automatically generated

**Section 8**

Receiver code

A picture containing graphical user interface

Description automatically generated