

# Liberate your school's micro:bits

In the spring of 2016, Scottish schools were given one BBC micro:bit programmable device for every S1 pupil.

The intention was to encourage young people to learn to code. There was an article about the giveaway in SSERC Bulletin 255 [1]. Since then, some schools have given out the micro:bits to pupils, but when we ask teachers what happened to them in their schools, a common reply is, "They're all in a box in the computing department."

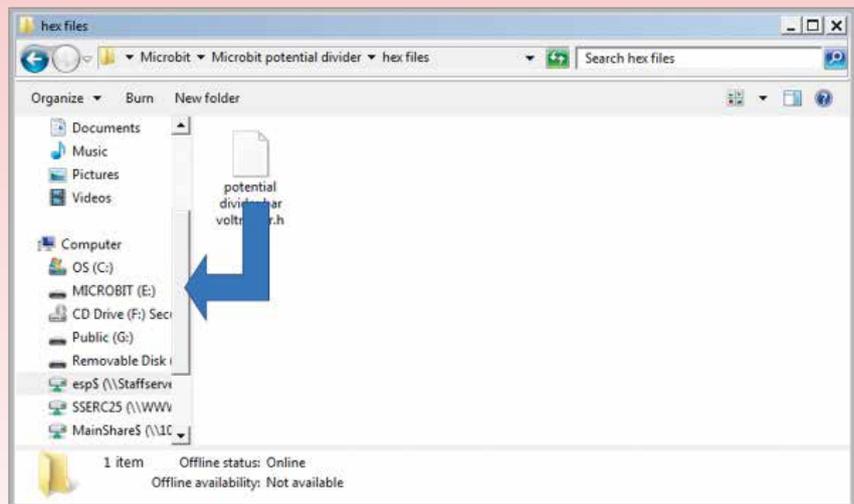


Figure 1 - Dragging a program on to the micro:bit.

If this is the case in your school, we're about to suggest that you liberate some of them as they are very easy to use as tools in science teaching. This might not be what was originally intended when micro:bits were given out, but we hope that what we're proposing will be a route towards some pupils coding for themselves.

If you visit the micro:bit area of our website [2], you will see a number of applications that we have devised. There are micro:bit versions of the Road Safety Reaction Timer [3] and the LED optics source [4]. We hope to cover the light gate system in a future Bulletin but the main focus of this article will be the teaching of potential divider circuits.

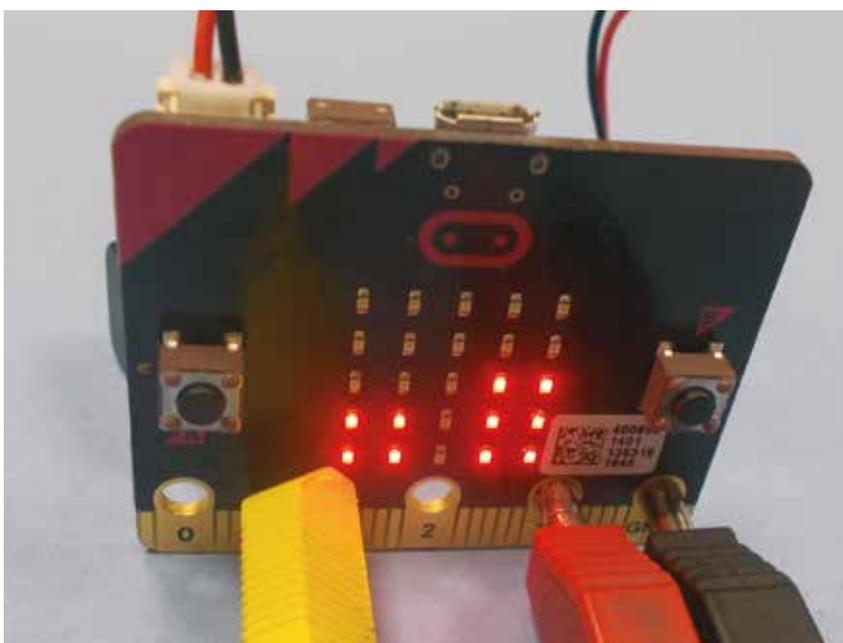


Figure 2 - Bar voltmeter display.

To get a micro:bit to run a pre-written program you need to do three things:

- Identify the program file you want to use. This might be one you have downloaded from our website. It will have the .hex file extension.
- Connect your micro:bit to a computer on which there is a copy of the .hex file you are going to use. Your micro:bit should show up as a drive on your computer - MICROBIT (E:) or (D:) or similar.
- Drag the .hex file on to the micro:bit, as shown in Figure 1.

All being well, a yellow LED on your micro:bit should flash. When it stops flashing, your micro:bit is programmed. It can only hold one program at a time.

### Python files

Files with the .py extension are also program files, written in a language called Python, and are available from our website for those who want to study or tweak our programs. They can be adapted using an application called an editor, then “flashed” on to a micro:bit. Flashing is done by the editor. It converts the Python program into a .hex file and puts it on the micro:bit. This is the correct way to program the micro:bit using a .py file. If you drag and drop them, they will not run.

Once a program is installed on a micro:bit, it remains there until another is put on. The micro:bit can be powered via the lead that connects it to the computer, but most of the time you are likely to use it with a battery pack.

Go to the micro:bit area of the SSERC website [2]. Download the Potential Divider material. Extract it from the .zip file and drag the file *pd bar voltmeter.hex* to your micro:bit as described above. If you have any difficulty locating or extracting the file, please get in touch.

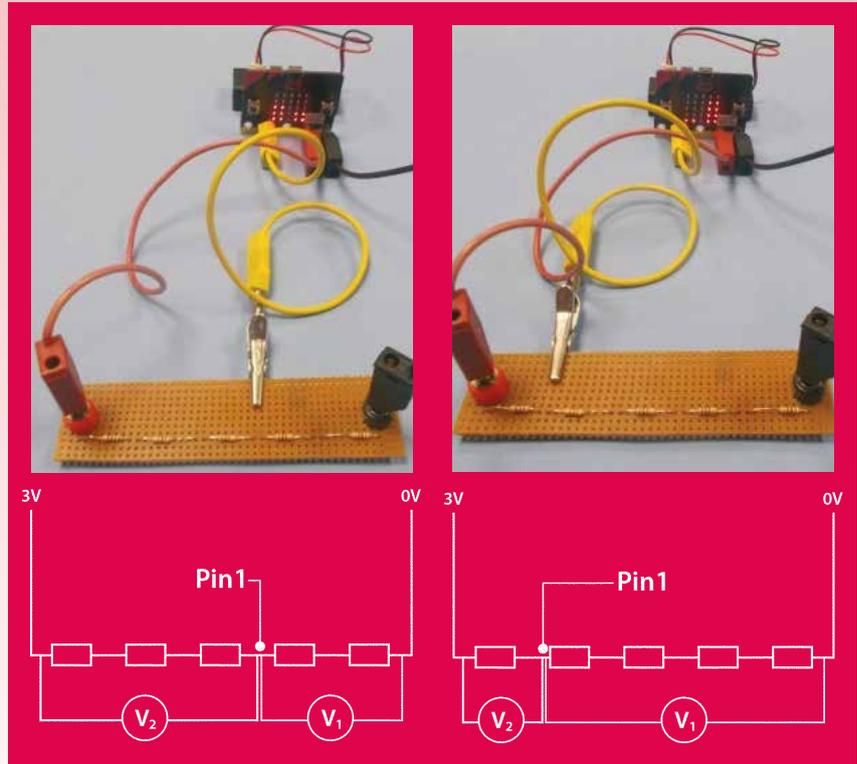


Figure 3 - Resistor chain potential divider.

Forgetting for a moment about the 4 mm leads in the picture, Figure 2 shows the sort of display you should see on your micro:bit.

The two columns of LEDs represent voltages across components in a potential divider. The connections, marked 0,1,2,3 V and GND, are called pins.

Figure 3 shows a simple potential divider circuit made up of a chain of 5 identical resistors. Whilst ours is soldered to strip board, we have also successfully made them using block connectors.

The ends of the potential divider are connected to GND and 3 V on the micro:bit. This means that there is a potential difference of 3 V across the resistor chain. Pin 1 on the micro:bit is connected to a lead that can be clipped to different points between the resistors. The micro:bit is programmed to read the voltage between Pin 1 and GND (0 V) and to display a bar proportional to this voltage on the right of its display. The bar on the left shows the potential difference between Pin 1 and the 3 V pin. The heights of the bars represent V1 (right bars) and V2 (left bars) in Figure 3. ▶

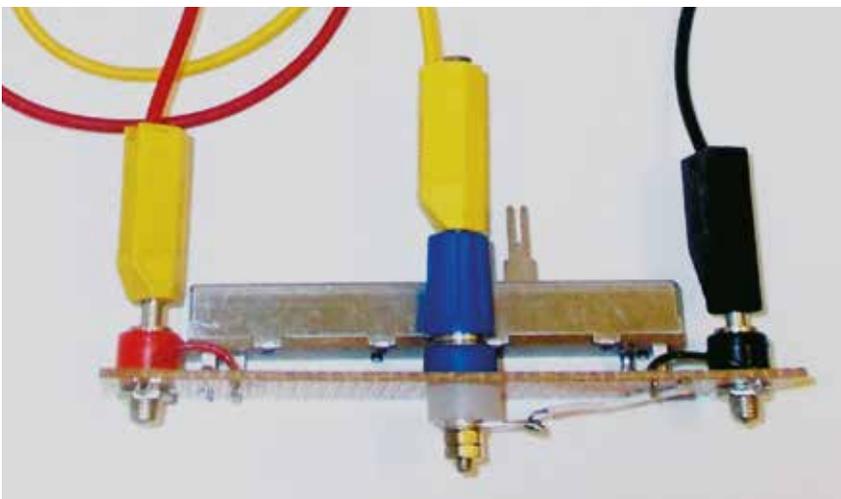


Figure 4 - Resistor chain replaced by potentiometer.

It is thus fairly easy for pupils to investigate the relationship between resistance and “share of the volts”.

An optional next step could be to connect a sliding potentiometer to the micro:bit (Figure 4).

Figure 5 shows a potential divider consisting of a 1 kΩ resistor and an LDR. The connection to Pin 1 is made to the junction between the LDR and resistor. The other connections are to GND and 3 V. The LDR’s resistance increases when it is dark. If we cover the LDR, there is therefore an increase in the share of the volts across it which will be indicated on the micro:bit display. Although the resistance of the fixed resistor has not changed, its share of the volts must decrease if the LDR’s share has increased.

### Capacitors

If you replace the LDR with a capacitor, say 100 microFarads, and the resistor with one of resistance 220 kΩ, it is possible to get data to enable students to plot a charging graph by finding the time to reach a particular voltage. Voltage could be measured in “bars”, or you could convert. In our program, 1 bar = 0.65 V.

The micro:bit is also capable of “writing a voltage” to a pin. This means that there will be a potential difference across any component connected between the pin and GND. The program used here writes a voltage of 3 V to Pin 2, but only when the voltage across the right hand component in the potential divider equals or exceeds 4 bars.

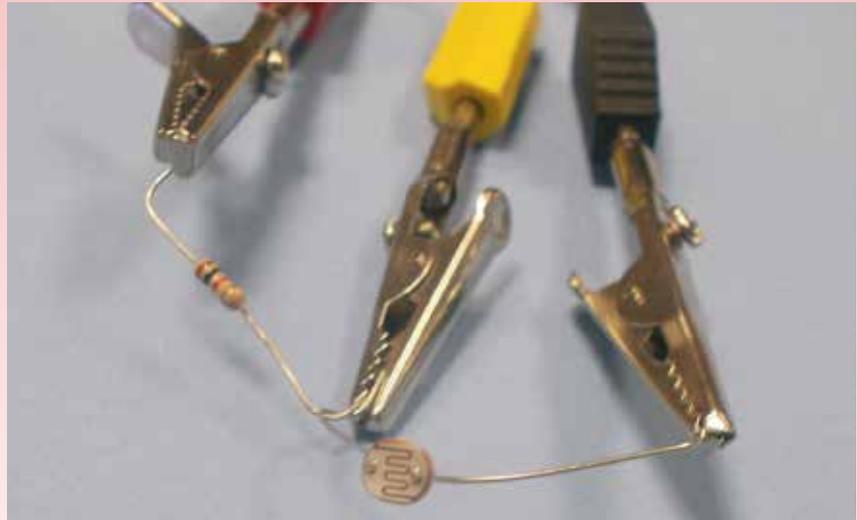


Figure 5 - LDR in potential divider.

Figure 6 shows an LED and protective resistor connected using a block connector. If this is used with a micro:bit running the potential divider program, the LED should come on when the LDR is in darkness. The program is set to make this happen when the voltage at Pin 1 exceeds a voltage determined by the coding. Essentially, it is mimicking a MOSFET circuit. However, the micro:bit is programmable. It is very easy to change the code so that it switches the output device on at a lower or indeed greater Pin 1 input voltage.

These set-ups are intended to help children to understand what happens to the voltages across components in a potential divider. They can then add an output device to make a circuit that actually does something. The next step is to adapt either the circuit or the program.

We are grateful to physics teacher Graham Crawford of Liberton High School in Edinburgh for testing out some of our ideas with his classes and feeding back their suggestions for improvement. A delegate at a recent course reported that he had adapted our light gates program to trigger the “Monkey and Hunter” experiment. We would love to hear from you too if you use the micro:bit.

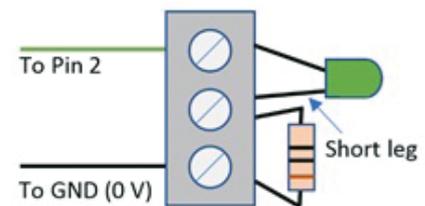


Figure 6 - Wiring an LED using a block connector.

### References

- [1] [http://www.sserc.org.uk/images/Bulletins/255/SSERC255p5\\_6.pdf](http://www.sserc.org.uk/images/Bulletins/255/SSERC255p5_6.pdf).
- [2] [www.tinyurl.com/microbit-sserc](http://www.tinyurl.com/microbit-sserc).
- [3] [http://www.sserc.org.uk/images/Bulletins/241/SSERC\\_bul241\\_p7.pdf](http://www.sserc.org.uk/images/Bulletins/241/SSERC_bul241_p7.pdf).
- [4] <http://www.sserc.org.uk/bulletins226/2007/222-autumn-2007/1227-led-source-for-optics242>.