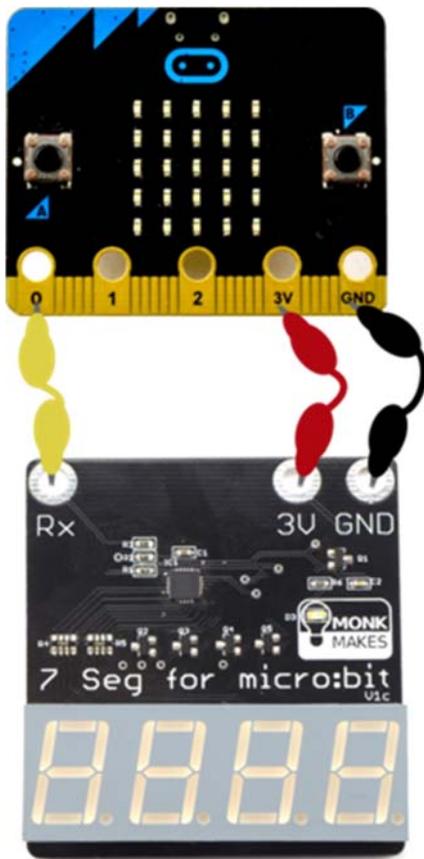# . 7-segment for micro:bit

MNK00066

The 7-segment for micro:bit is a four digit 7-segment display for micro:bit. You can use it to display numbers, but it can also display letters and other characters!

Features

- Easy to connect (just needs one micro:bit pin plus power)
- Powered directly from micro:bit pins
- Send messages to the display using the micro:bit's Serial blocks
- Useful for displaying readings from sensors, making clocks etc
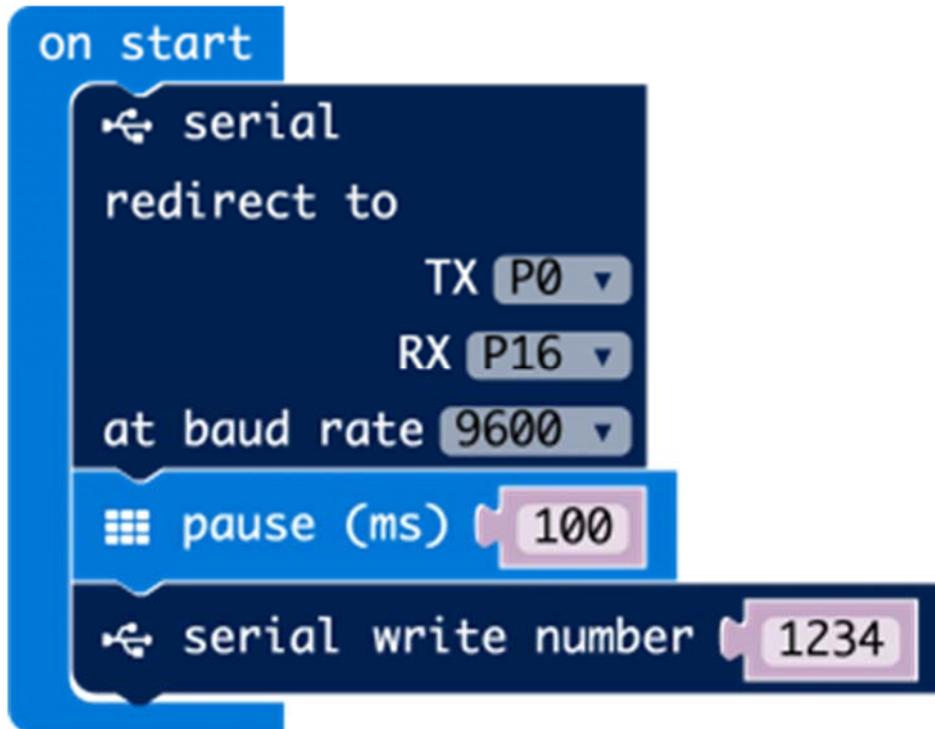
Getting Started

Connecting to your micro:bit

Connect the power pins GND and 3V between the micro:bit and the 7-Segment for micro:bit. Connect the Rx (receive) pin of the 7-Segment for micro:bit to any of the micro:bit's pins.

Block Code Examples

Display '1234'

This first example displays the number 1234.

The **serial redirect to** block allows the micro:bit to send (TX) serial messages on pin 0. It sets the RX pin to pin 16, but this is not used. When using the 7-segment display, the micro:bit only transmits messages, it does not receive them.

Note that you need to set **at baud rate** to 9600 as this is the speed of communication that the display is expecting.

The **pause** is necessary after redirecting serial communication to allow time for communication to start.

Using the **serial write number** block writes a value to the display.

DISPLAY 'Byte'

As well as displaying numbers, you can also display text, although with just seven segments per digit many of the letters and punctuation are very approximate. For example we could display the word 'Byte' using the code below.
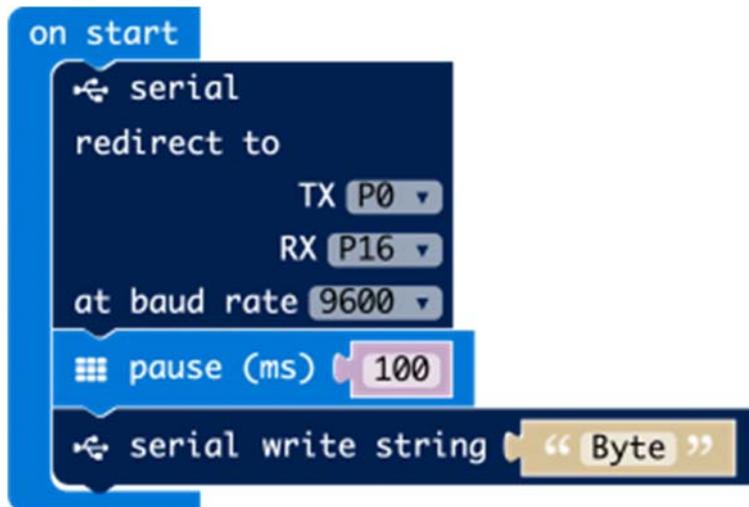
Note that in this case, we are using the **serial write string** block as we want to display text rather than a number.

Clearing the Display

If you just send text or numbers to the display using **serial write string** or **serial write number**, then each time you send another character it pushes all the currently displayed characters one position to the left and then puts the new character in the right-most position. If you send the special character "/", the display will be cleared.

Buffered Mode

If you want your display to update repeatedly, then just writing new values as described above will result in the display flickering. To avoid this you need to use the display in buffered mode. In this mode, you compose what you want to be displayed and then when you are ready tell the display to change whatever is currently being displayed to what you now want to be displayed.

The special character "," (comma) is used to switch the display into buffered mode and also to update the display. The next example uses the display in buffered mode.

Countdown Timer

This timer starts at a number (change n to the number up to 9999 that you want to count down from). When the countdown reaches 0 the message "Done" will be displayed.
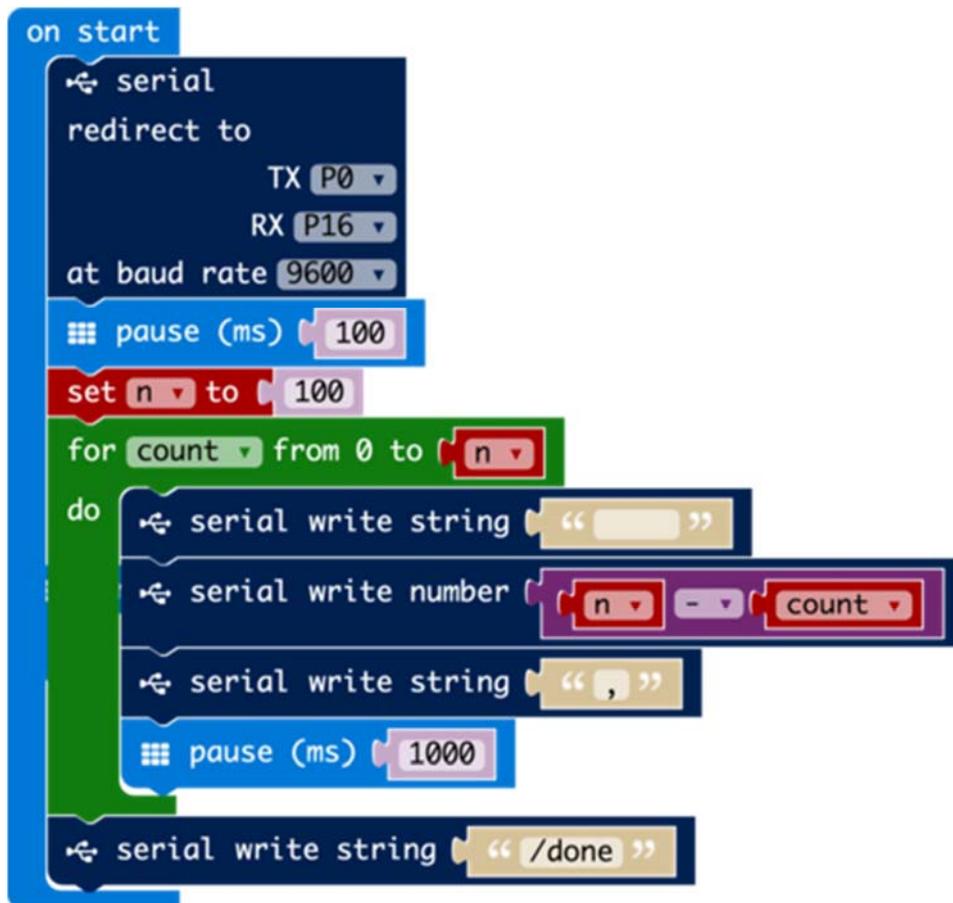

The value of count actually increases from 0 to whatever **n** is, but the number displayed is **n – count**, so the number displayed decreases.

Each time around the loop four spaces are sent to the display to clear it of any existing number. This is an alternative to using "/" but unlike "/" does not change the display mode back to non-buffered. The number is then sent to the display followed by the "," (comma) command to tell the display to swap its current contents for the new value.

When the count down has finished, the message "done" is displayed. The "/" at the start of this message puts the display back into non-buffered mode so that a final comma is not needed to refresh the display.

Thermometer

This display is a great companion to the MonkMakes Sensor for micro:bit. In this example, we use this display to show the current temperature.
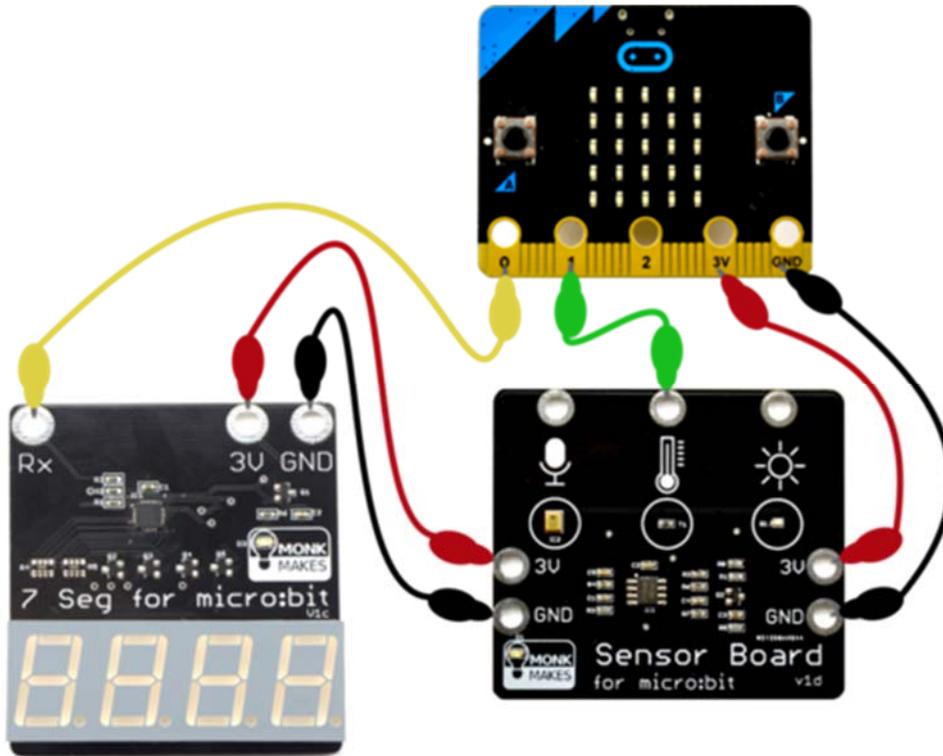
As you can see, the micro:bit supplies power to the Sensor board that then supplies power to the display.

Here is the code for the thermometer. Click on it to open it.

For more information on how the temperature measurement part of the code works, see the Sensor boards page.

To display the temperature the display is used in "buffered mode". The next value to display is first blanked and then the temperature sent to the display followed by "*C,". The * character displays something close to the degrees symbol and the comma on the end tells the display that now is the time to switch what is displayed to the new message.

Trouble Shooting

Problem: The display is not very bright.

Solution: The limited current available from the micro:bit means that the display has to be kept at a low brightness to be able to work. The display will be less bright when using batteries through the battery connector than when using a USB power supply.

Problem: The display is not refreshing.

Solution: The display takes time to operate and so if you try and refresh it too fast, it can hang. If it does this, introduce a longer delay between refreshes using the **pause** block. Momentarily un-clipping the positive power lead to the display will cause it to reset if the display has stopped updating.