

Glow Your Own

Session 4

Building and coding your circuit

You can build, test and code your circuits virtually on Tinkercad: www.tinkercad.com

If you would like to join the Glow Your Own class on Tinkercad, please log on to <https://www.tinkercad.com/joinclass/TQ4WP2KL8Z91> and enter your nickname. If you need a reminder of your nickname, or a new nickname, please email visitral@stfc.ac.uk

For a reminder of how to access Tinkercad and get started, see the 'How To' sheet for Session 1. There are instructions for how to upload your code in the 'Duino' How To sheet.

Remember, if you're using building physical circuits with your real Arduino:

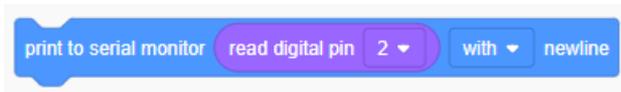
- Always remember to unplug your Arduino when you're changing components.
- Make sure all of the small components are kept out of reach of young children.
- Make sure that all components are tidied up at the end of the session, and none are left on the floor or table.

Using the serial monitor

You might have noticed by now, but coding often doesn't work the first time you write something! In this case, it's helpful to be able to **debug** (fix) our code by getting the Arduino to let us know what values it's using. We can do that using the **serial monitor**.

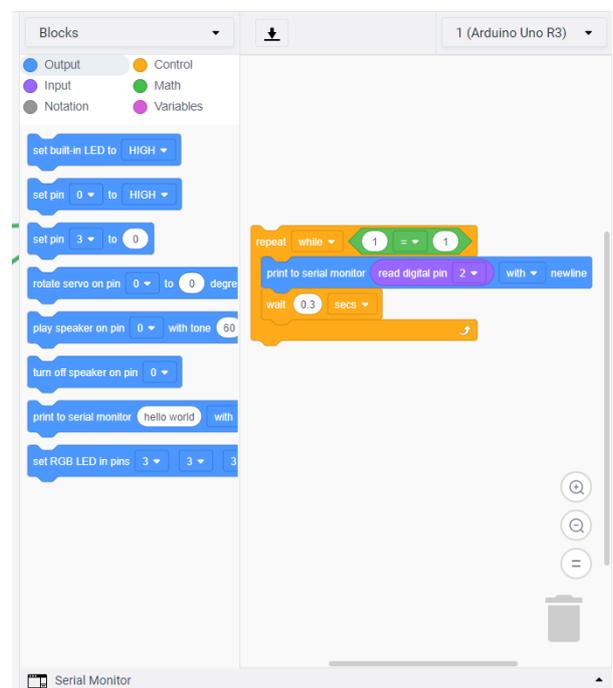
The serial monitor in Tinkercad

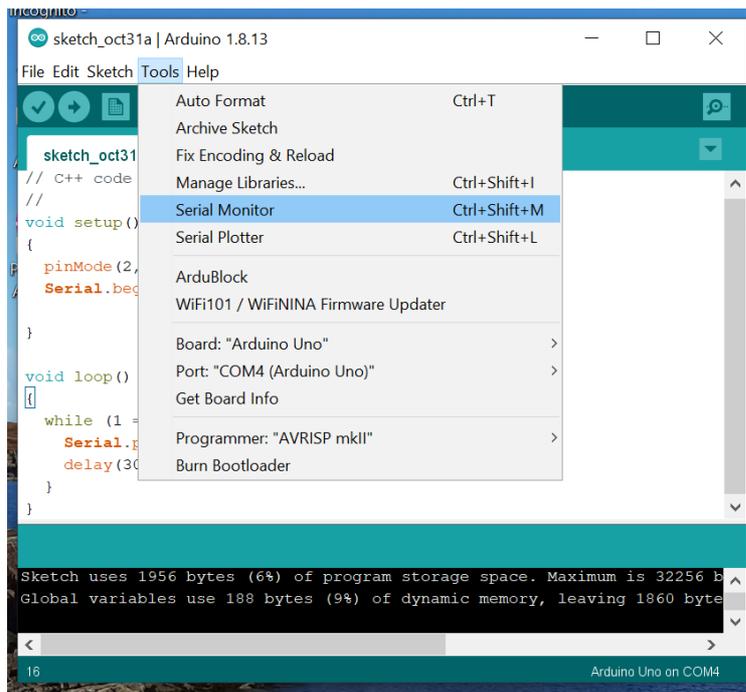
In Tinkercad, you can ask the Arduino to print a value to the serial monitor using the **"print to serial monitor"** block



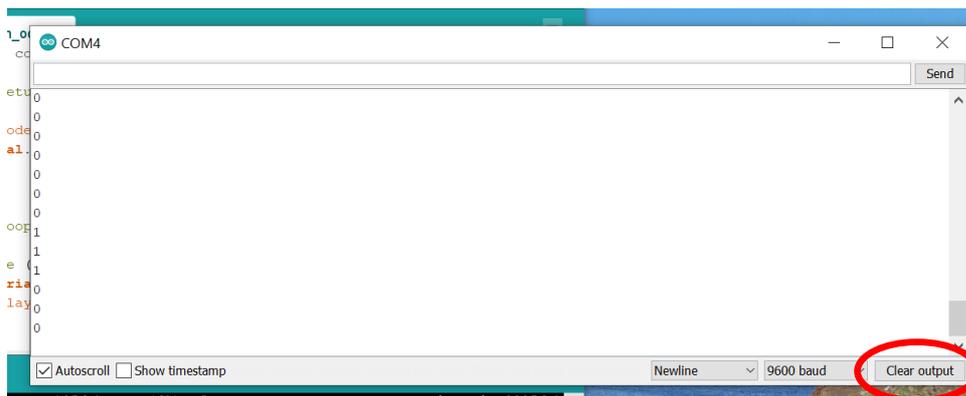
Here, we're asking the Arduino to print the value it is reading from digital pin 2. You can see the serial monitor in Tinkercad at the bottom of the Code section:

When you click on the Serial Monitor button, you will see values appear when you simulate your code.





This opens up a new window, with the serial monitor – you can clear the serial monitor using the “Clear output” button.



Circuits to build and code

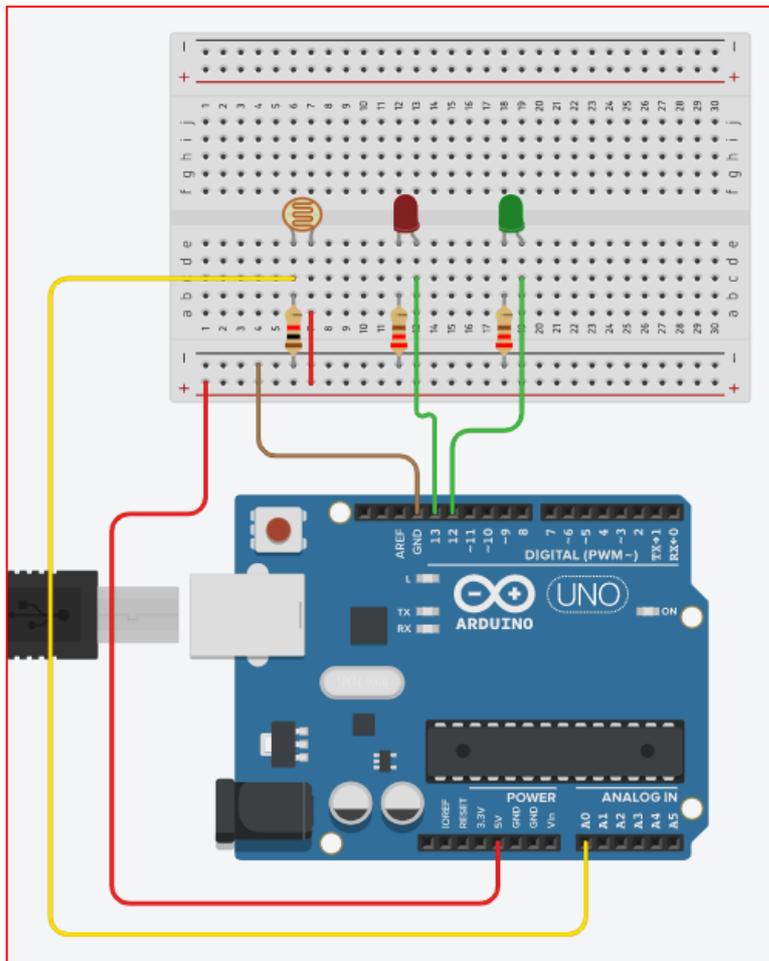
Turning an LED on when it gets dark

You can control your LED by measuring the light around your circuit using a **photoresistor** – so that the LED turns on when it gets dark.

The photoresistor changes its value depending on the light around it. In Tinkercad and in your Arduino kits, the photoresistor looks like this:



To start with, let's build a circuit that makes our LEDs turn on when it gets dark. There are two circuits: one output circuit (with the LED), and one input circuit (with the photoresistor). As you can see, the photoresistor needs another resistor to work properly – use a 1 k Ω resistor for this. Use 220 Ω resistors for the LEDs, as before.



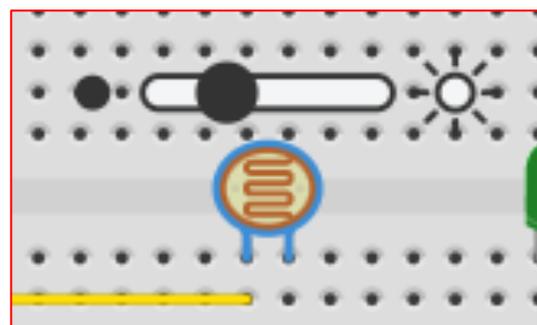
To code our new circuit, we need to use an “if then else” block: as with the button, this block is used when we want to ask the computer a question. We ask what value the photoresistor is saying (which tells us how bright it is). If it's dark, we instruct the Arduino to turn the LEDs on, if it's very

dark we instruct the Arduino to turn just one LED on, and if it's bright we instruct the Arduino to turn the LEDs off, using the code below.

```
forever
  print to serial monitor read analog pin A0 with newline
  if read analog pin A0 > 550 then
    comment it is quite bright ...
    print to serial monitor It is quite bright! with newline
    set pin 12 to LOW
    set pin 13 to LOW
  else
    if read analog pin A0 > 350 and read analog pin A0 < 551 then
      comment it is average light..
      print to serial monitor It is average light with newline
      set pin 12 to HIGH
      set pin 13 to LOW
    else
      comment it is dark...
      print to serial monitor It is dark! with newline
      set pin 12 to HIGH
      set pin 13 to HIGH
  wait 0.3 secs
```

The block “**print to serial monitor**” means we can tell what value the photoresistor is telling the Arduino, as it will print that value to the screen. You’ll also notice we’ve put in some **comments**. Comments are really useful in your code – the Arduino ignores them, but they tell us (and other people!) what you are trying to do in your code.

When you’re simulating your circuit in Tinkercad, click on the photoresistor and using the slider you can change the light level the photoresistor sees.



Challenge with photoresistors

The values that we've used in the code above, to decide whether it's dark or light, can be found by looking at the serial monitor, and experimenting to see what values the photoresistor returns when it's very dark or very bright. These are then 'hard coded' into the Arduino (we type the numbers in: 451 and 229 above). We can code the Arduino to work out those numbers for itself by calibrating the photoresistor: recording the readings when it's bright and dark.

Can you work out how to calibrate your photoresistor? You could use a button to measure the maximum and minimum light reading over a period of 5 seconds, and use these values in the code above. You'll need to use variables to do this! Good luck!