

Glow Your Own Coding Club – June Workshop Guide

During this workshop, we will work on our Arduino coded art projects, by making our own physical circuits, with the 'hack' textile button to activate a sequence of LEDs.

Here is a step-by-step guide of the session:

Items needed:

- Arduino + USB Cable (and UNO R3 software download)
- Breadboard
- LEDs (approx. 2-3)
- Jumper/Dupont wires (4x black wires for ground and 6x wires in a colour of your choice)
- Resistors (1x 1k Ω and 3x 220 Ω)
- 2 x Crocodile clips
- Conductive tape
- 2 x 5cm Long cables (with either end open)* *cut two 5 cm strips of cable then get a pair of scissors and carefully shave off the plastic at either end of these two 5cm strips so that about 1cm of wire is visible on either end.
- 1 x Glove, or a bit of paper
- 1 x A4 paper
- Coloured pens

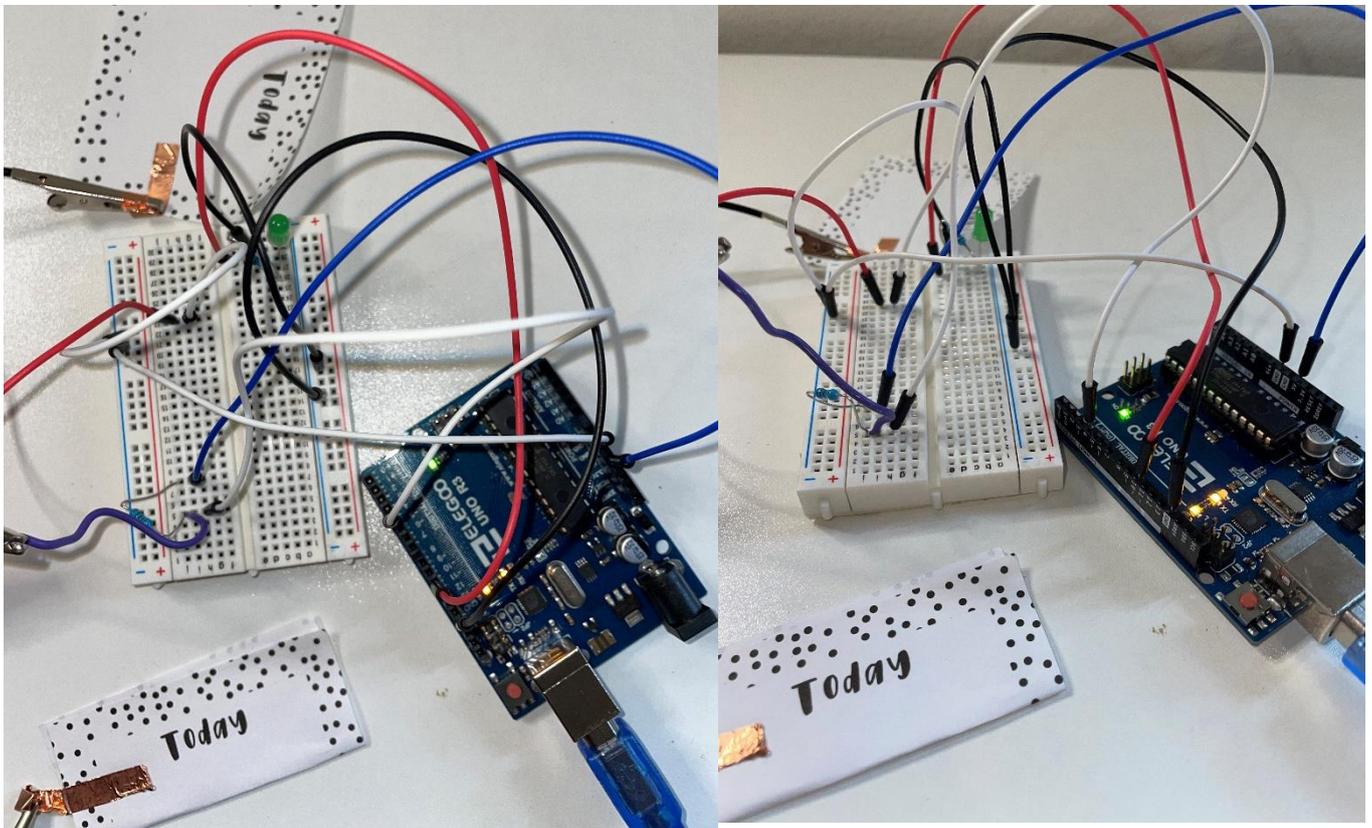
If you can before the session:

Clear the breadboard so that only two LEDs are set up as they were set up in May's session. To recap or re-make this set-up, follow steps 1-10 below.

- 1) Collect your equipment together.
- 2) Get an LED (any colour) and place the longer end in A28 and the shorter end in A29
- 3) Get a 220 Ω resistor and it place in B28 and C28
- 4) Put a coloured wire (of your choice) in E28 and pin 10 on Arduino
- 5) Put 1x ground wire (black) in C29 and then in the blue MINUS side of the breadboard, close to column J and row 29.
- 6) Put 1x ground wire (black) into the bottom of the blue MINUS side of the breadboard, close to column J, and the other end of the wire into a GND pin on your Arduino.
- 7) Place one more LED into your breadboard, putting the long end into A17 and the short end into A19
- 8) Put a 220 Ω resistor into B17 and C17
- 9) Put a coloured wire (of your choice) into E17 and in pin 11 on your Arduino
- 10) Put 1x ground wire (black) into C19 and the blue MINUS strip of the breadboard closest to column J, and near row 19.

Now for the 'Hack' touch button

- 11) Place a 1K Ohm resistor into J8 and J5
- 12) Take one jumper wire or cut wire (as described above) and place one end into i5 and connect the other end to one end of the crocodile clip.
- 13) At the other end of this crocodile clip, clip in a small bit of conductive tape
- 14) Take a coloured jumper wire and place one end into i8 and the other end into 5V pin on the Arduino board
- 15) Get one other coloured jumper wire and place one end into H5 and the other end into Pin 4 on the DIGITAL side of the Arduino board.
- 16) Get one jumper wire (or cut wire, as described above) and place one end into i23 and connect the other end to one end of a different crocodile clip.
- 17) At the other end of this crocodile clip, clip in a small bit of conductive tape.
- 18) Take a black jumper wire and place one end into H23 and the other end into the blue MINUS strip closest to J.



- 11) Copy and paste the below code into your UNO R3 Software and export it to your Arduino (as per previous GYO pdf guides).
[Make sure the lines of code are displayed as below, by either copying and pasting one line at a time or copying the whole code and adjusting the line spacing using your Enter key whilst in the UNO R3 Software]:

```
// constants won't change. They're used here to set pin numbers:
const int buttonPin = 4; // the number of the pushbutton pin
const int ledPin1 = 10; // the number of the LED pin
const int ledPin2 = 11; // the number of the LED pin

// variables will change:
int buttonState = 0; // variable for reading the pushbutton status

void setup() {
  Serial.begin(9600);
  // initialize the LED pin as an output:
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the buttonState is LOW:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin1, LOW);
    delay(300);
    digitalWrite(ledPin2, LOW);
    delay(300);
  } else {
    // turn LED off:
    digitalWrite(ledPin1, HIGH);
    delay(300);
    digitalWrite(ledPin2, HIGH);
    delay(300);
  }
  Serial.println(buttonState);
}
```

Common troubleshooting

- Check that all components are pushed into pins fully (resistors can be a little tricky to push in and they bend easily, so we recommend checking these first!)

When uploading code to Arduino:

- Check that the software is recognising the correct Port (*go to 'Tools' in the top toolbar, then select 'Port'. You need to select the COM Port that has your Arduino listed*).
- Check that the software is registering your Arduino as the correct model (*go to 'Tools' in the top toolbar, then select 'Board'. Make sure the model you select is the same as is written on your Arduino*).