Glow Your Own

Session 6

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.

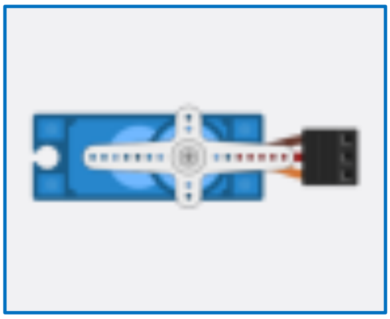
Remember, if you're using building physical circuits with your real Arduino:

- 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.

Circuits to build and code

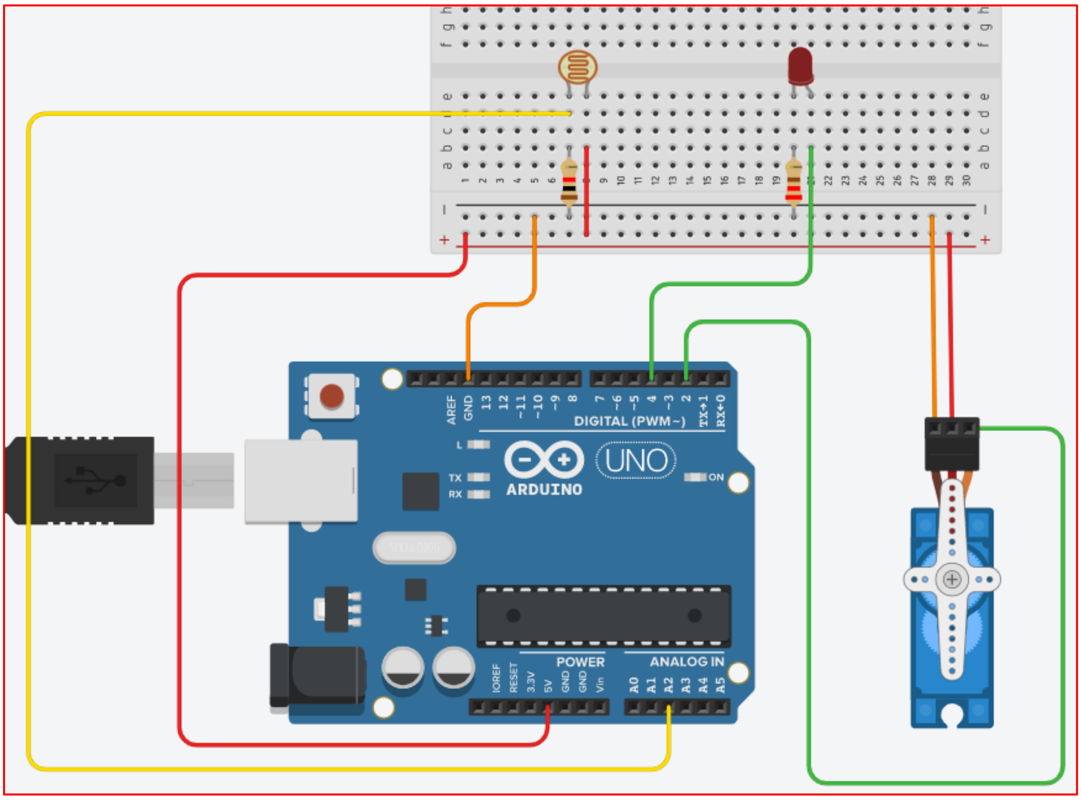
Using a motor

Motors are really useful – we use them all the time in our experiments to control our instruments and our robots. We can also use them in our lanterns, to move components around. We will be using a type of motor called a **servo motor** – which rotates to an angle set by the Arduino. Using Tinkercad, we can write code and use an Arduino to control how we want the motor to move. Most servo motors only rotate between 0 and 180 degrees – half a circle. Servo motors look like this:

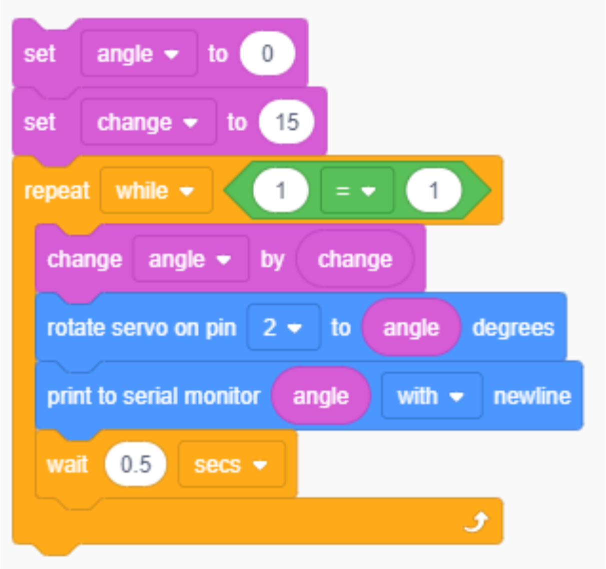


Firstly, we'll build the circuit that connects the servo motor to the Arduino. We can keep the circuit we've used previously – controlling an LED using a photoresistor or a button, and just ann the servo motor. There are three 'legs' to the servo motor – and they're helpfully coloured in Tinkercad so it's easier to know how to connect each pin. The brown leg is connected to the Ground, the red leg is connected to power (5V) and the orange leg is connected to the Arduino pin that sends the signal (which tells the servo motor how to rotate). In Tinkercad, hovering over each part also tells you how to connect the servo.

In the circuit on the next page, the servo motor is connected to pin 2 – we'll use that pin to send our commands to the servo motor, to tell it how to move. Notice that the colours of the wires match – with brown wires going to Ground and red wires going to power. This makes it easier to spot if there are any mistakes!

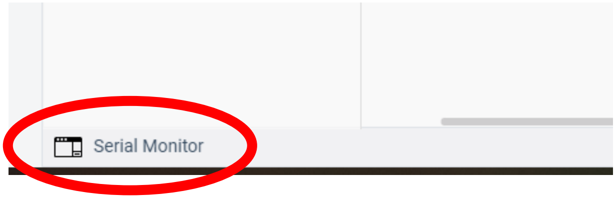


To code the servo motor, the easiest way is to use variables (values that the Arduino remembers), to tell the servo motor how to move. To start with, we'll code the servo motor to move through half a circle using the following code:



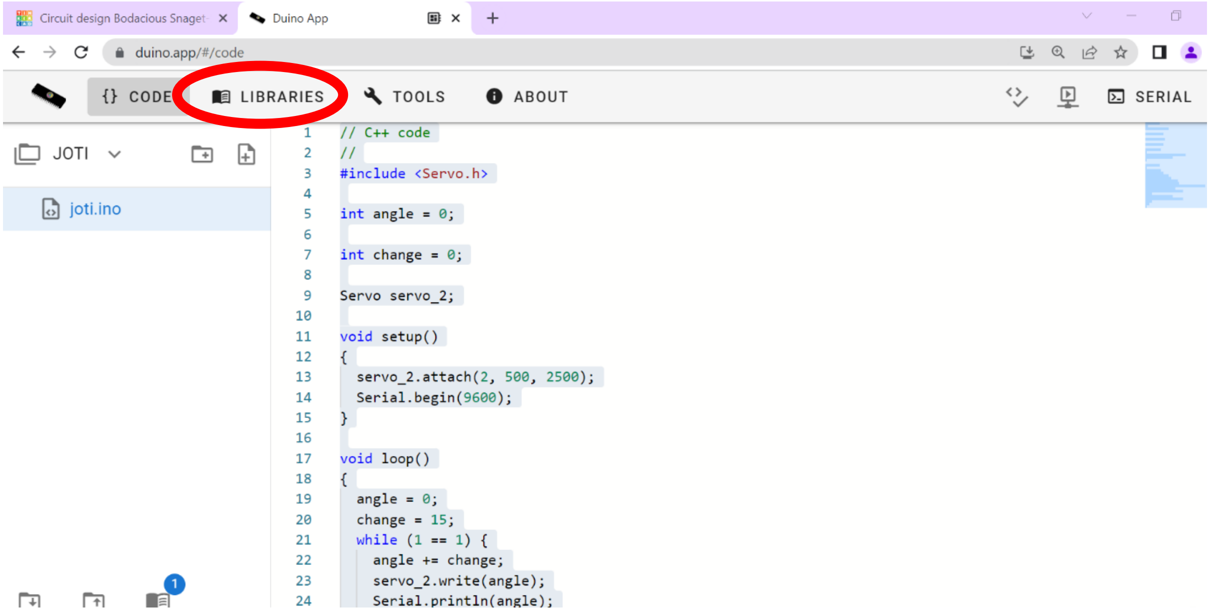
```
set angle ▾ to 0
set change ▾ to 15
repeat while ▾ 1 = ▾ 1
    change angle ▾ by change
    rotate servo on pin 2 ▾ to angle degrees
    print to serial monitor angle with ▾ newline
    wait 0.5 secs ▾
```

Remember, the block **“print to serial monitor”** means we know the angle of the servo motor, as it will print that value to the screen. If you’re using Tinkercad, you can see the Serial Monitor at the bottom of the screen:



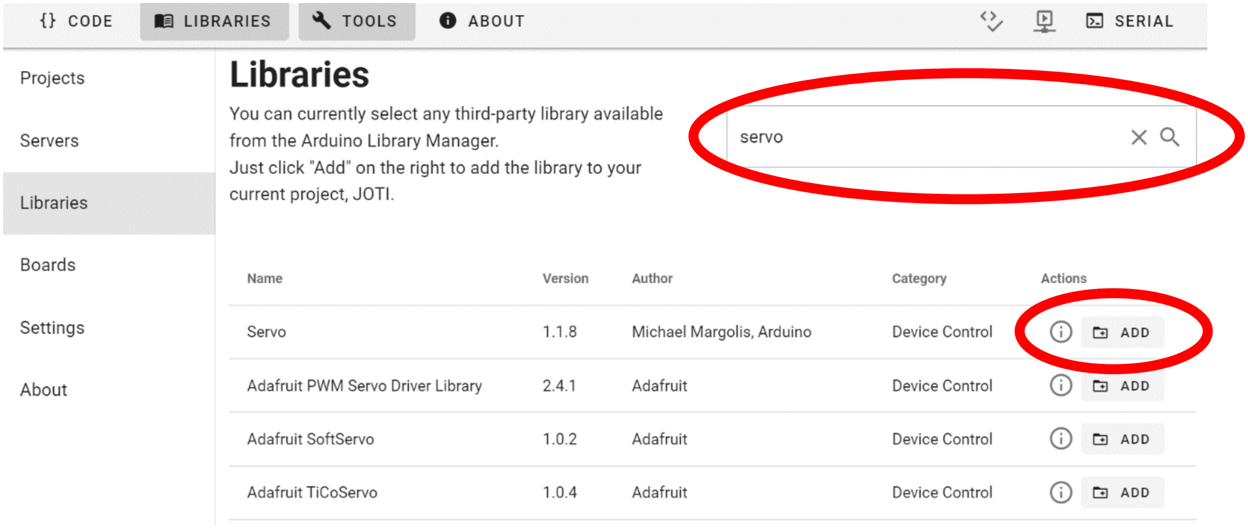
If you’re using a physical Arduino and the IDE, you can see the Serial Monitor by clicking on the “**Tools**” menu and choosing “**Serial Monitor**”. In the Duino App, you can see the Serial Monitor by clicking on the **“Monitor”** tab.

Talking to servo motors is pretty complicated, but luckily people have already written a whole set of instructions that allow Arduinos to talk to them: when we’re using a real Arduino, we need to make sure these instructions are included. They’re written in something called a Library: you can add it in the Duino App by clicking on LIBRARIES at the top:



```
// C++ code
//
#include <Servo.h>

int angle = 0;

int change = 0;

Servo servo_2;

void setup()
{
  servo_2.attach(2, 500, 2500);
  Serial.begin(9600);
}

void loop()
{
  angle = 0;
  change = 15;
  while (1 == 1) {
    angle += change;
    servo_2.write(angle);
    Serial.println(angle);
```

In the Search bar, type “servo”:



Libraries

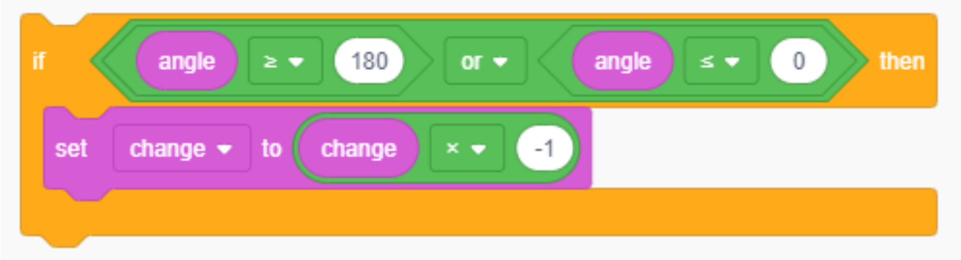
You can currently select any third-party library available from the Arduino Library Manager.
Just click "Add" on the right to add the library to your current project, JOTI.

Name	Version	Author	Category	Actions
Servo	1.1.8	Michael Margolis, Arduino	Device Control	ADD
Adafruit PWM Servo Driver Library	2.4.1	Adafruit	Device Control	ADD
Adafruit SoftServo	1.0.2	Adafruit	Device Control	ADD
Adafruit TiCoServo	1.0.4	Adafruit	Device Control	ADD

Then click “ADD” next to the top search result – called “Servo”. You can then compile and upload your code to your Arduino as normal.

Does your motor move?

You’ll notice that the servo motor only moves through half a circle. You can make it go backwards and forwards by using an **if statement**, as in the code below:



```
if angle ≥ 180 or angle ≤ 0 then
    set change to change × -1
```

Put the **if statement** in your **repeat loop**. Does the motor move backwards and forwards now? How could you use this in your lanterns?