COMP6043 Physical Computing Lab 4

Note: if a task asks you to demonstrate your work to your lecturer, you must demonstrate and get your work signed off. Otherwise, no marks will be awarded for your work.

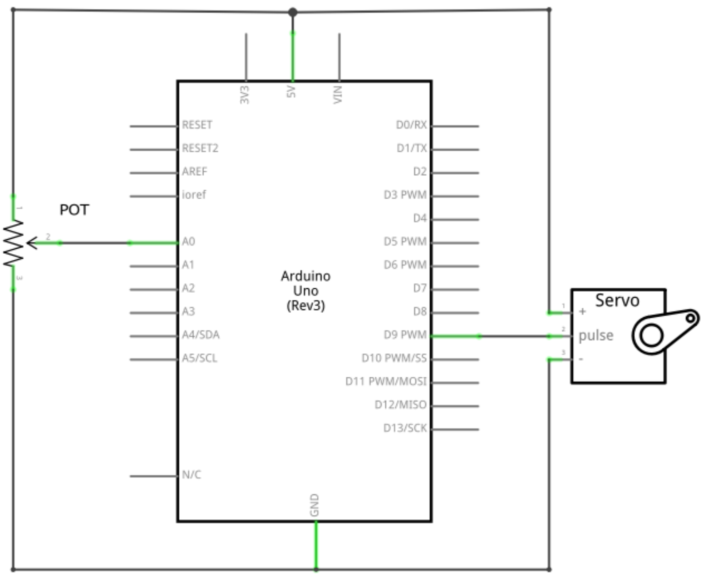
Once completed, upload your report to Canvas.

Pay particular attention when connecting the PCB to the Arduino power pins. It is very important that you connect with the correct polarity, i.e. 5 V on the Arduino to 5 V on the PCB, and GND on the Arduino to GND on the PCB. If you wire up with the incorrect polarity, you will burn out the on-board temperature sensor and damage the board.

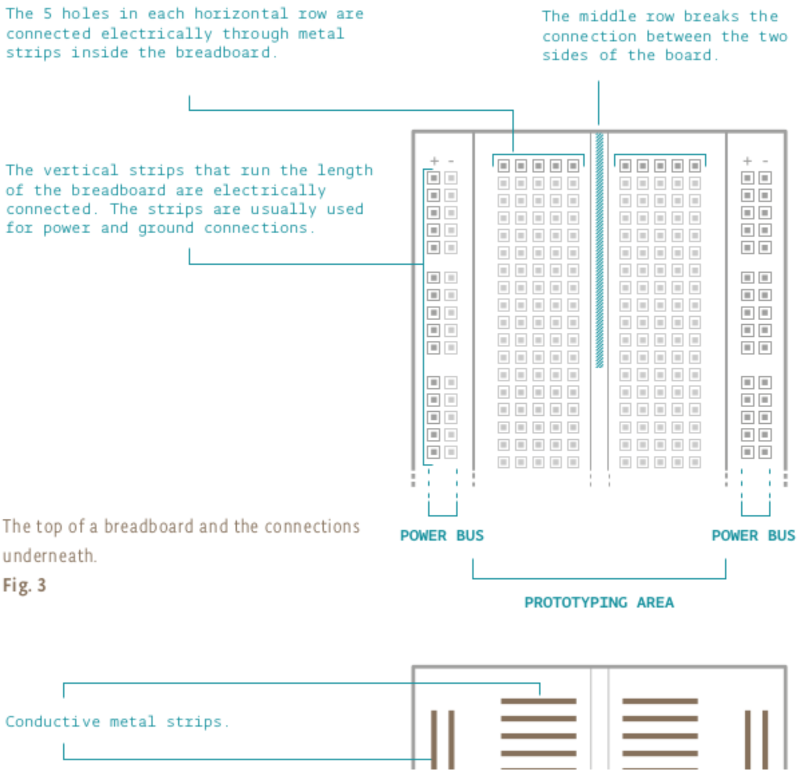
This lab is based on the “Mood Cue” project in the Arduino Projects Book (p.62).

Task 1 [60%]

Wire up your circuit as follows:



In order to wire up the servo motor, you will need to connect it to the breadboard. As shown below, the horizontal and vertical rows of the breadboard carry electricity through thin metal connectors under the plastic with holes. All points on the same conductive strip are electically common, allowing you to easily connect multiple components together.



The 5 holes in each horizontal row are connected electrically through metal strips inside the breadboard.

The middle row breaks the connection between the two sides of the board.

The vertical strips that run the length of the breadboard are electrically connected. The strips are usually used for power and ground connections.

Conductive metal strips.

The top of a breadboard and the connections underneath.
Fig. 3

Deliver power to the breadboard by connecting 5 V and GND pins on the PCB to the power rails (vertical strips) of the breadboard:

- 5 V → +
- GND → −

Connect your servo motor to the "prototyping area" and wire up as required.

Open the Arduino IDE and enter the following code:

```
// include the servo library
#include <Servo.h>

Servo myServo;  // create a servo object

int const potPin = A0; // analog pin used to connect the potentiometer
int potVal;  // variable to read the value from the analog pin
int angle;   // variable to hold the angle for the servo motor

void setup() {
  myServo.attach(9); // attaches the servo on pin 9 to the servo object
  Serial.begin(9600); // open a serial connection to your computer
}
```

```
void loop() {
  potVal = analogRead(potPin); // read the value of the potentiometer
  // print out the value to the serial monitor
  Serial.print("potVal: ");
  Serial.print(potVal);

  // scale the numbers from the pot
  angle = map(potVal, 0, 1023, 0, 179);

  // print out the angle for the servo motor
  Serial.print(", angle: ");
  Serial.println(angle);

  // set the servo position
  myServo.write(angle);

  // wait for the servo to get there
  delay(15);
}
```

Upload your code to the Arduino and verify that your project works correctly. When you turn the potentiometer dial, you should observe the servo motor move to a new position. Open the serial monitor and note the values displayed.

Demonstrate your project to your lecturer once completed.

Task 2 [10%]

Add an LED to your project, the brightness of which depends on the position of the servo arm. When the servo arm is at 0°, the LED should be off. It should be at full brightness when the servo arm is at 179°, 50% brightness when at 90°, etc.

Copy and paste your code into your report.

Demonstrate your project to your lecturer once completed.

Task 3 [10%]

Add another LED to your project. This LED should be on when the servo motor is within 10° of its extremeties (i.e. less than 10° or greater than 170°) and off otherwise.

Copy and paste your code into your report.

Demonstrate your project to your lecturer once completed.

Task 4 [20%]

Add a push-button switch to your project such that the servo motor will only change position if the button is pressed. If the button is not pressed, the servo motor should remain in its current position, regardless of a change in the position of the potentiometer dial.

Copy and paste your code into your report.

Demonstrate your project to your lecturer once completed.