COMP6043 Physical Computing Lab 6

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.

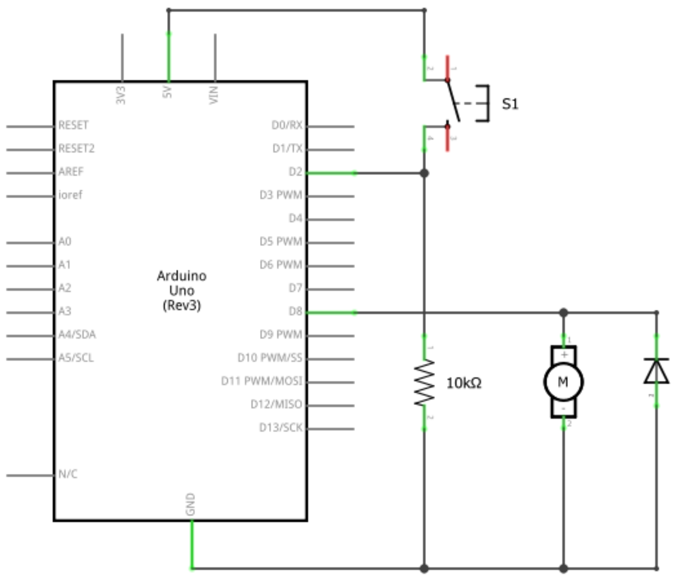
Once completed, upload your report to Canvas.

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 "Motorized Pinwheel" and "Zoetrope" projects in the Arduino Projects Book (p.94 – 113).

Task 1 [20%]

Attempt to drive the motor directly from the Arduino by wiring up your circuit as follows:



Note: the cathode of the diode is indicated by a stripe. The purpose of the diode is to eliminate flyback, i.e. the sudden voltage spike seen across an inductive load when its supply current is suddenly reduced or interrupted. Although the diode is not actually needed here (since the Arduino will not be able to deliver enough current to drive the motor), it is good practice to include it anyway.

Open the Arduino IDE and enter the following code:

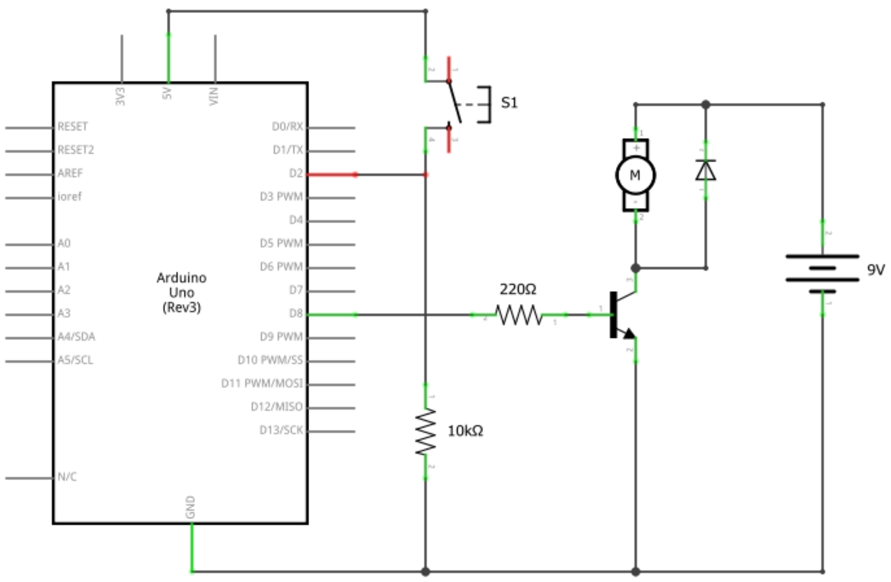
```
// named constants for the switch and motor pins
const int switchPin = 2; // the number of the switch pin
const int motorPin =  8; // the number of the motor pin

int switchState = 0;  // variable for reading the switch's status

void setup() {
  // initialize the motor pin as an output:
  pinMode(motorPin, OUTPUT);
  // initialize the switch pin as an input:
  pinMode(switchPin, INPUT);
}

void loop(){
  // read the state of the switch value:
  switchState = digitalRead(switchPin);

  // check if the switch is pressed.
  if (switchState == HIGH) {
    // turn motor on:
    digitalWrite(motorPin, HIGH);
  }
  else {
    // turn motor off:
    digitalWrite(motorPin, LOW);
  }
}
```

Upload your code to the Arduino. Attempt to turn on the motor and observe the result.

Demonstrate your project to your lecturer once completed.

Task 2 [30%]

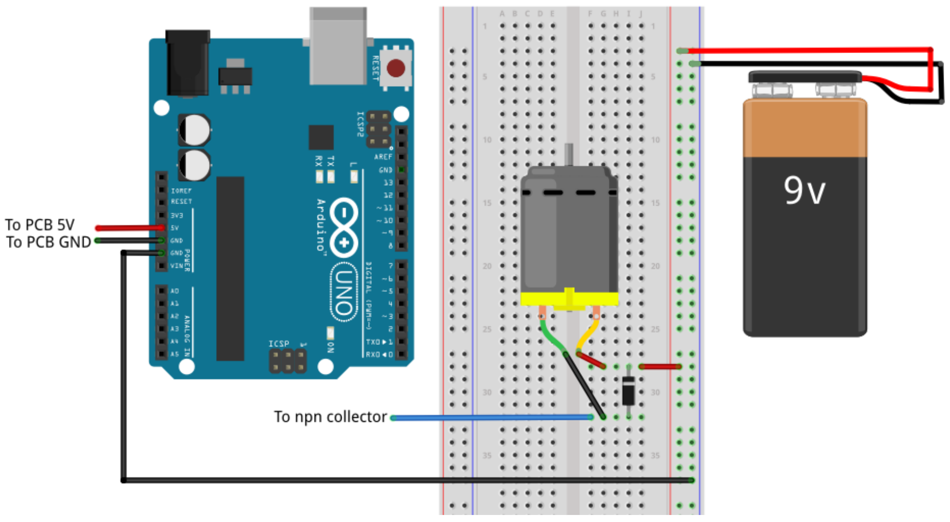
The Arduino cannot deliver enough current to drive the DC motor so we will instead switch the motor on/off via a transistor. Wire up your circuit as follows (remove power from the Arduino while you are wiring up your circuit):



The 220 Ω resistor connected to the transistor base is already in place on the PCB. Similarly, the transistor emitter is already connected to GND on the PCB.

NOTE: if the diode is not connected correctly, you will blow the transistor when the motor switches off. Call your lecturer to check your circuit before you connect the 9 V battery and plug in the Arduino.

The following is a "breadboard-view" of how to connect the 9 V battery, the DC motor and the diode:



Once your lecturer has checked your circuit, plug the Arduino back in. Since you are using the same code from task 1, you do not have to re-upload your code. You should find that you are now able to turn the motor on via S1.

Demonstrate your project to your lecturer once completed.

Task 3 [10%]

Given that the base-emitter forward voltage, V_{BE}, is 0.7 V, and the common-emitter current gain, β_F, is 100, where

$$\beta_F = \frac{I_C}{I_B},$$

calculate the current flowing through the motor when it is switched on.

Include all calculations in your report.

What component supplies this current? What component controls this current?

Task 4 [20%]

Modify your project so that you can adjust the speed of the motor via a potentiometer, i.e. the motor speed should be proportional to the output voltage from the potentiometer.

Copy and paste your code into your report.

Demonstrate your project to your lecturer once completed.

Task 5 [10%]

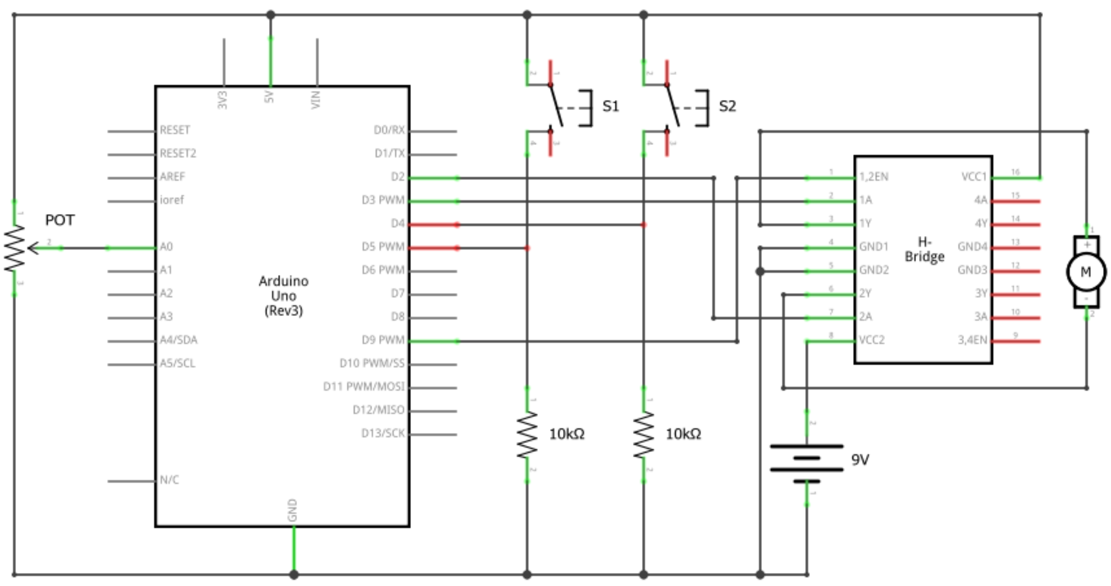
Express the motor speed as a percentage of its full speed and display the result on the serial monitor, e.g. if the motor is running at half-speed, you should read 50% on the serial monitor.

Copy and paste your code into your report.

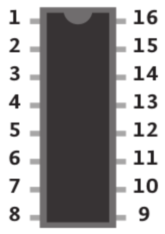
Task 6 [10%]

If you reverse the polarity of the voltage across the motor, it will rotate in the opposite direction. Rather than do this manually, we can use a H-bridge to reverse the polarity via a switch.

Wire up your circuit as follows:



Note: you can identify the orientation (and, therefore, the pin numbers) of the H-bridge as follows:



Enter the following code in the Arduino IDE:

```
const int controlPin1 = 2;
const int controlPin2 = 3;
const int enablePin = 9;
const int directionSwitchPin = 4;
const int onOffSwitchStateSwitchPin = 5;
const int potPin = A0;

// create some variables to hold values from your inputs
int onOffSwitchState = 0;
int previousOnOffSwitchState = 0;
int directionSwitchState = 0;
int previousDirectionSwitchState = 0;

int motorEnabled = 0; // Turns the motor on/off
int motorSpeed = 0; // speed of the motor
int motorDirection = 1; // current direction of the motor
```

```
void setup(){
  // intialize the inputs and outputs
  pinMode(directionSwitchPin, INPUT);
  pinMode(onOffSwitchStateSwitchPin, INPUT);
  pinMode(controlPin1, OUTPUT);
  pinMode(controlPin2, OUTPUT);
  pinMode(enablePin, OUTPUT);

  // pull the enable pin LOW to start
  digitalWrite(enablePin, LOW);
}

void loop(){
  // read the value of the on/off switch
  onOffSwitchState = digitalRead(onOffSwitchStateSwitchPin);
  delay(1);

  // read the value of the direction switch
  directionSwitchState = digitalRead(directionSwitchPin);

  // read the value of the pot and divide by 4 to get
  // a value that can be used for PWM
  motorSpeed = analogRead(potPin)/4;

  // if the on/off button changed state since the last loop()
  if(onOffSwitchState != previousOnOffSwitchState){
    // change the value of motorEnabled if pressed
    if(onOffSwitchState == HIGH){
      motorEnabled = !motorEnabled;
    }
  }

  // if the direction button changed state since the last loop()
  if (directionSwitchState != previousDirectionSwitchState) {
    // change the value of motorDirection if pressed
    if (directionSwitchState == HIGH) {
      motorDirection = !motorDirection;
    }
  }

  // change the direction the motor spins by talking
  // to the control pins on the H-Bridge
  if (motorDirection == 1) {
    digitalWrite(controlPin1, HIGH);
    digitalWrite(controlPin2, LOW);
  }
  else {
    digitalWrite(controlPin1, LOW);
    digitalWrite(controlPin2, HIGH);
  }

  // if the motor is supposed to be on
  if (motorEnabled == 1) {
    // PWM the enable pin to vary the speed
    analogWrite(enablePin, motorSpeed);
  }
  else { // if the motor is not supposed to be on
    //turn the motor off
    analogWrite(enablePin, 0);
  }
  // save the current On/Offswitch state as the previous
  previousDirectionSwitchState = directionSwitchState;
  // save the current switch state as the previous
  previousOnOffSwitchState = onOffSwitchState;
}
```

Demonstrate your project to your lecturer once completed.