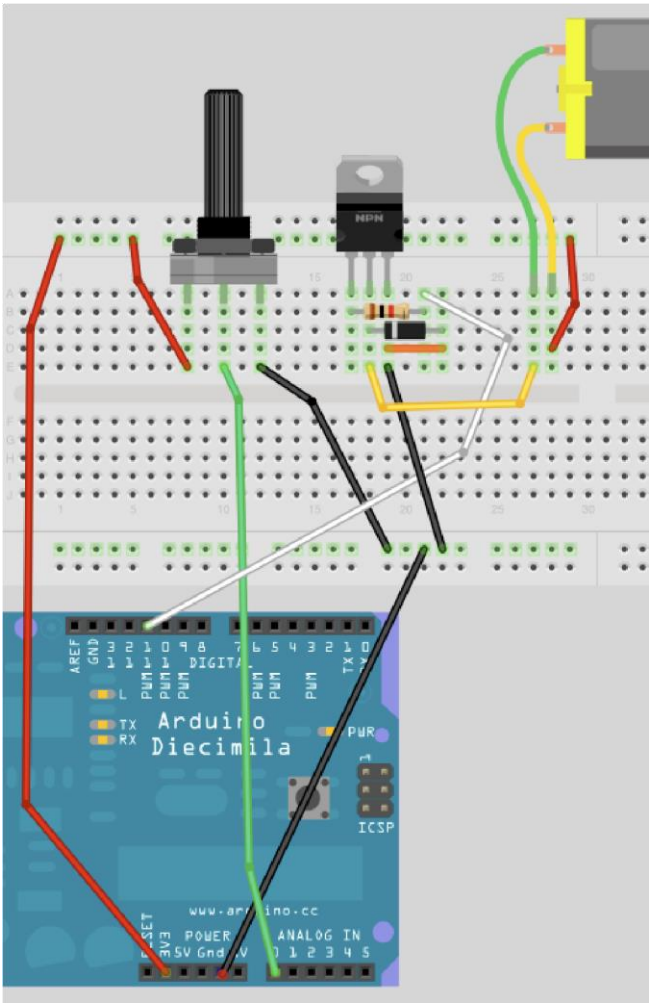
Project 11 - Drive a DC Motor

We are now going to step away from LED 's for a little while and take a look at a different piece of hardware, the DC Motor. In your kit you have been supplied with a standard 1.5 to 4.5 volt DC Motor. In this project you must power the Arduino using the 9v DC Power Supply and NOT the USB cable. Before uploading the sketch, turn the potentiometer all the way anti-clockwise. After the sketch is uploaded DISCONNECT the USB cable before powering the motor so it is powered by the 9v supply only. Also make sure your diode is the right way around with the white stripe going to +5V.

What you will need

DC Motor	
4K7 Potentiometer	
TIP-120 Transistor	
1N4001 Diode	
1KΩ Resistor	
9v Power Supply	

Connect it up



Enter the code

```
int potPin = 0;           // Analog in 0 connected to the potentiometer
int transistorPin = 11;

 // connected to the base of the transistor
int potValue = 0;         // value returned from the potentiometer

void setup() {
  // set  the transistor pin as output:
  pinMode(transistorPin, OUTPUT);
}

void loop() {
  // read the potentiometer, convert it to 0 - 255:
  potValue = analogRead(potPin) / 4;
  // use that to control the transistor:
  analogWrite(transistorPin, potValue);
}
```

Project 11 - Code Overview

Before you power up the circuit, double check that everything has been connected correctly and in particular that the diode is the right way around. Failure to do this can result in permanent damage to your Arduino. Once you are happy that you have wired it all up correctly then go ahead and upload your code.

```
int potPin = 0;
int transistorPin = 11;
int potValue = 0;
void setup() {
pinMode(transistorPin, OUTPUT);
}
void loop() {
potValue = analogRead(potPin) / 4;
analogWrite(transistorPin, potValue);
}
```

This code is very simple indeed. We declare 3 integers that will hold the values of the pin we attach our potentiometer to, the pin we connect the transistor to and the value read from the potentiometer.

In the setup() function we set the pinmode of the transistor pin to put.

In the main loop potValue is set to the value read in from analog pin 0 (the potPin) and then divided by 4.

We need to divide the value read in by 4 as the analog value will range from 0 for 0 volts to 1023 for 5 volts. The value we need to write out to the transistor pin can only range from 0 to 255 so we divide the value of analog pin 0 (max 1023) by 4 to give the maximum value of 255 for setting the digital pin 11 (Using analogWrite so we are using PWM).

The code then writes out to the transistor pin the value of the pot. In other words, when you rotate the potentiometer, different values ranging from 0 to 1023 are read in and these are converted to the range 0 to 255 and then that value is written out (via PWM) to digital pin 11 which changes the speed of the DC motor. Turn the pot all the way to the left and the motor goes off, turn it to the right and it speeds up until it reaches maximum speed when the pot is turned clockwise all the way.

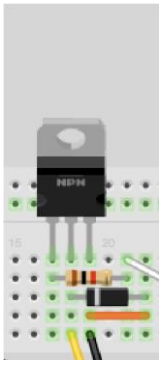
Now let us find out how our new electronic components introduced in this project work.

Project 11 - Hardware Overview

The hardware for Project 11 is as below:

DC Motor	
4K7 Potentiometer	
TIP-120 Transistor	
1N4001 Diode	
1KΩ Resistor	
9v Power Supply	

The circuit is essentially split into two sections. Section 1 is our potentiometer, which is connected to +3.3v and Ground with the centre pin going into Analog Pin 0. As the potentiometer is rotated, the resistance changes to allow voltages from 0 to 3.3v to come out of the centre pin, where the value is read using Analog Pin 0.

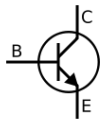


The second section is what controls the power to the motor. The digital pins on the Arduino give out a maximum of 40mA (milliamps). The 3V DC Motor requires around 500mA to operate at full speed and this is obviously too much for the Arduino. If we were to try to drive the motor directly from a pin on the Arduino serious and permanent damage could occur.

Therefore, we need to find a way to supply it with a higher current. We therefore take power directly from the 3.3v pin on the board, which takes its power in turn from the DC regulator on the board which takes power from our 9v Power Supply. A DC regulator is the larger black object next to the DC in jack and all it does is take the input voltage and reduce it down to 5v. It is good for 800mA of power, which is more than we need for our small DC motor.

However, this project controls the speed of the motor so we need a way to control that voltage to speed up or slow down the motor. This is where the TIP-120 transistor comes in.

A transistor is essentially a digital switch. It can also be used as a power amplifier. In our circuit we use it as a switch. The electronic symbol for a transistor look like this:



The transistor has 3 legs, one is the Base, one is the Collector and the other the Emitter. These are marked as C, B and E on the diagram.

In our circuit we have 3.3 volts going into the Collector via the motor. The Base is connected via a 1KR resistor to Digital Pin 11. The Emitter is connected to Ground. We send pulses via PWM out to Pin 11, and this voltage is reduced using a 1KR resistor. Whenever we apply a voltage to the base, via Pin 11, this makes the transistor turn on allowing current to flow through it from Collector to Emitter and therefore powering the motor that is connected in series with this circuit.
A motor is an electromagnet and it has a magnetic field whilst power is supplied to it. When the power is removed, the magnetic field collapses and this collapsing field can produce a reverse voltage to go back up its wiring. This could seriously damage your Arduino and that is why the diode has been placed the wrong way around on the circuit. The white stripe on the diode normally goes to ground. Power will flow from the positive side to the negative side. As we have it the wrong way around no power will flow down it at all. As we have it across the C and E legs of the transistor this will not impede current flow at all. However, if the motor were to produce a “back EMF” and send current back down the wire, the diode will act as a valve and prevent it from doing so. The diode in our circuit is therefore put in place to protect your Arduino.