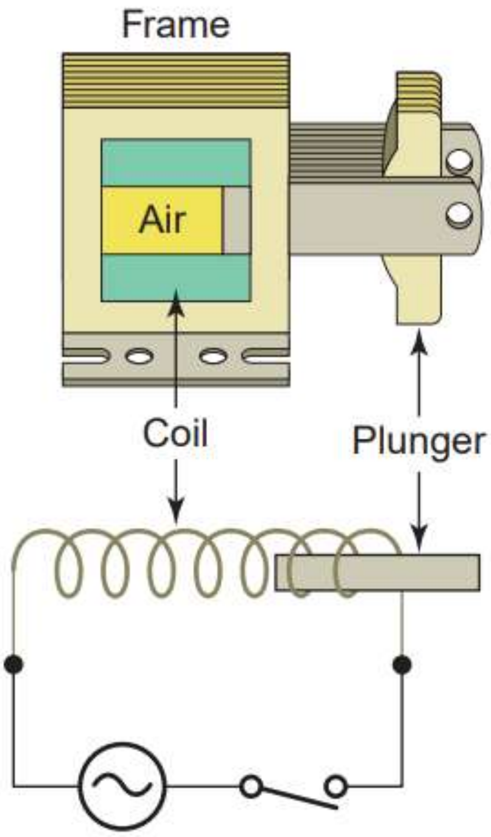
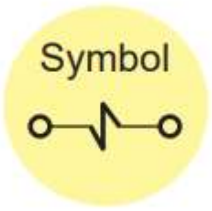
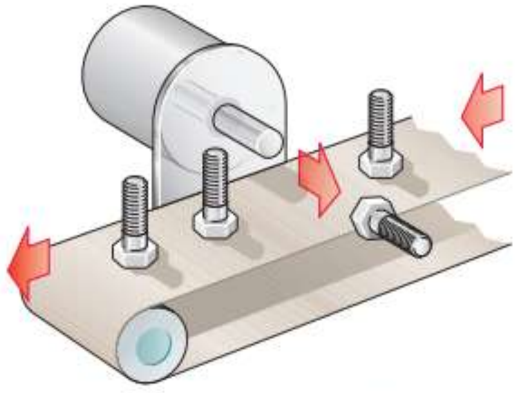
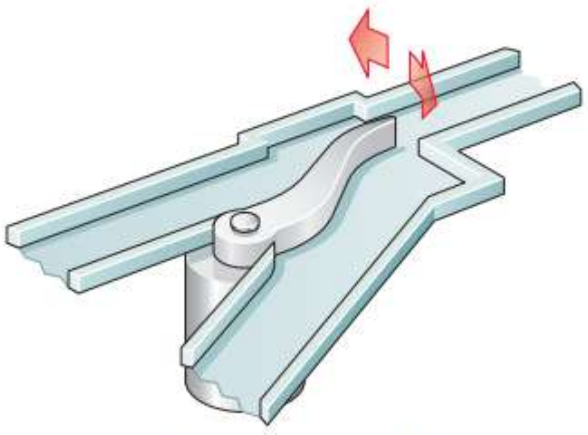
SOLENOIDI





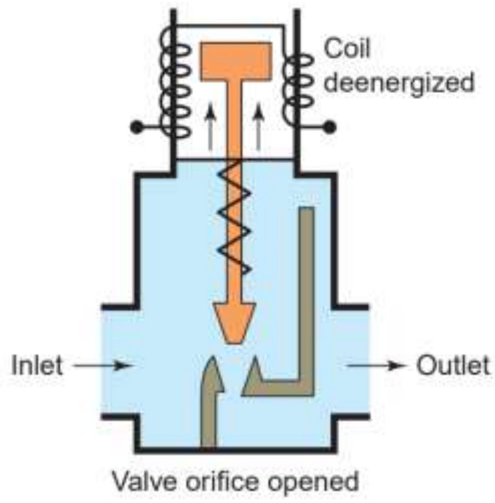


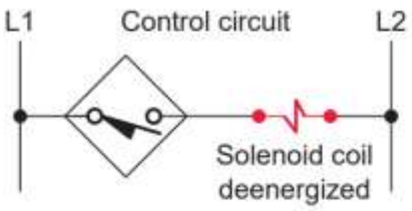
Linear solenoid

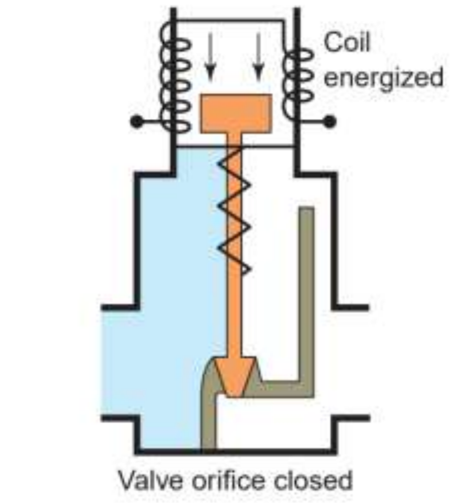


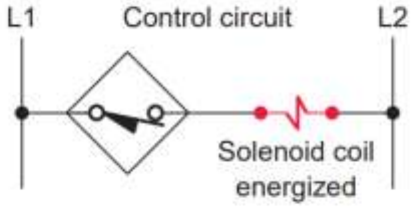
Rotary solenoid

VALVOLA A SOLENOIDE

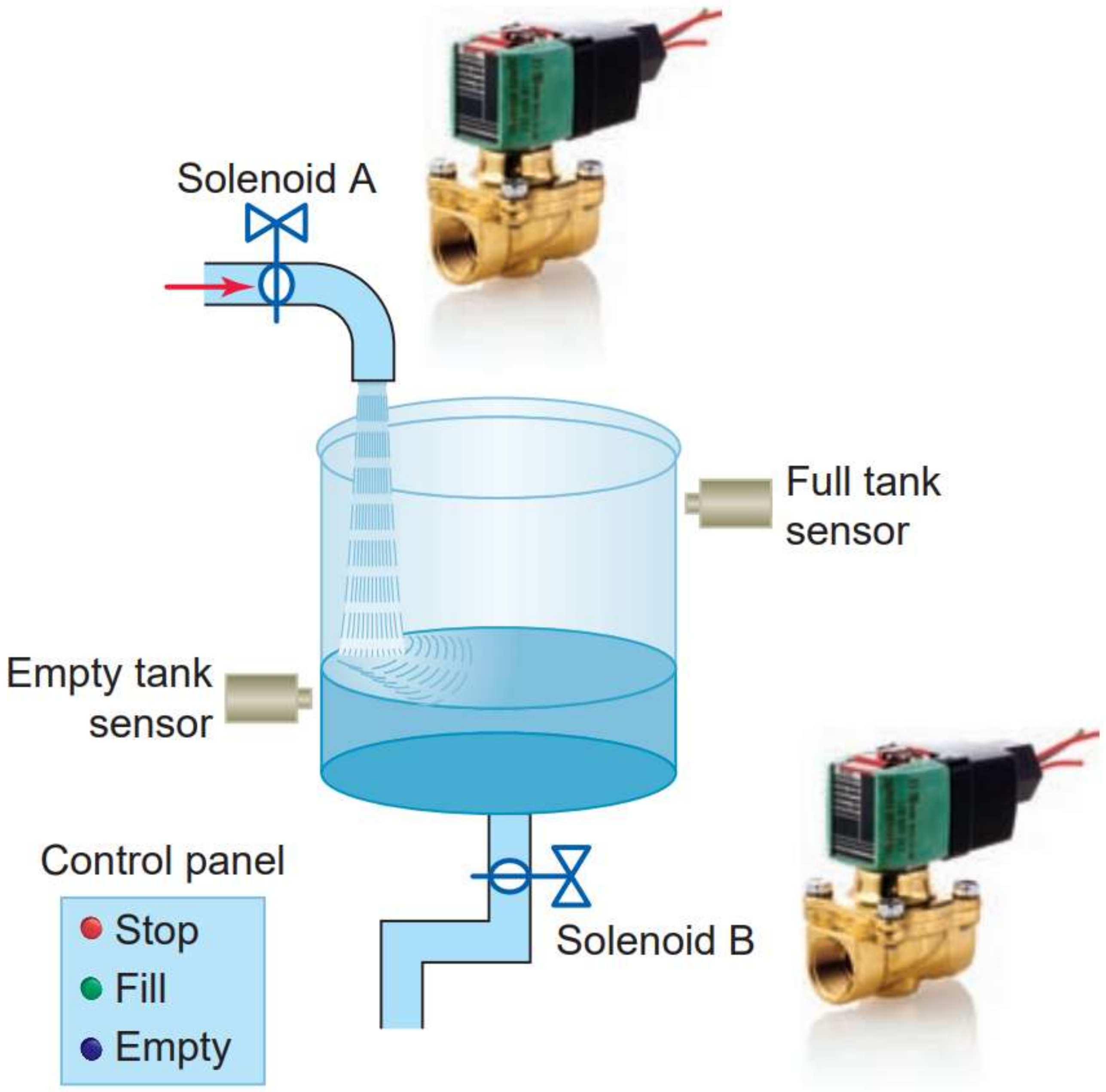




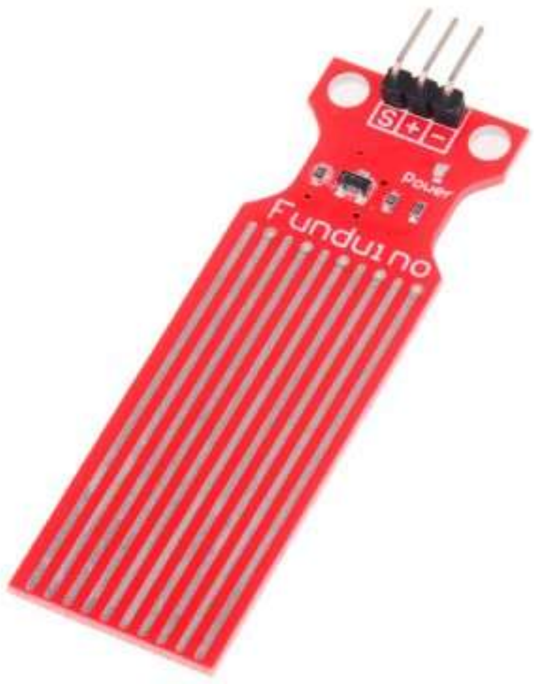




SISTEMA DI CONTROLLO LIVELLO ACQUA SERBATOIO



Water sensor



E’ un sensore analogico che puoi collegare ad arduino utilizzando uno dei 6 pin analogici (A0 – A5) .

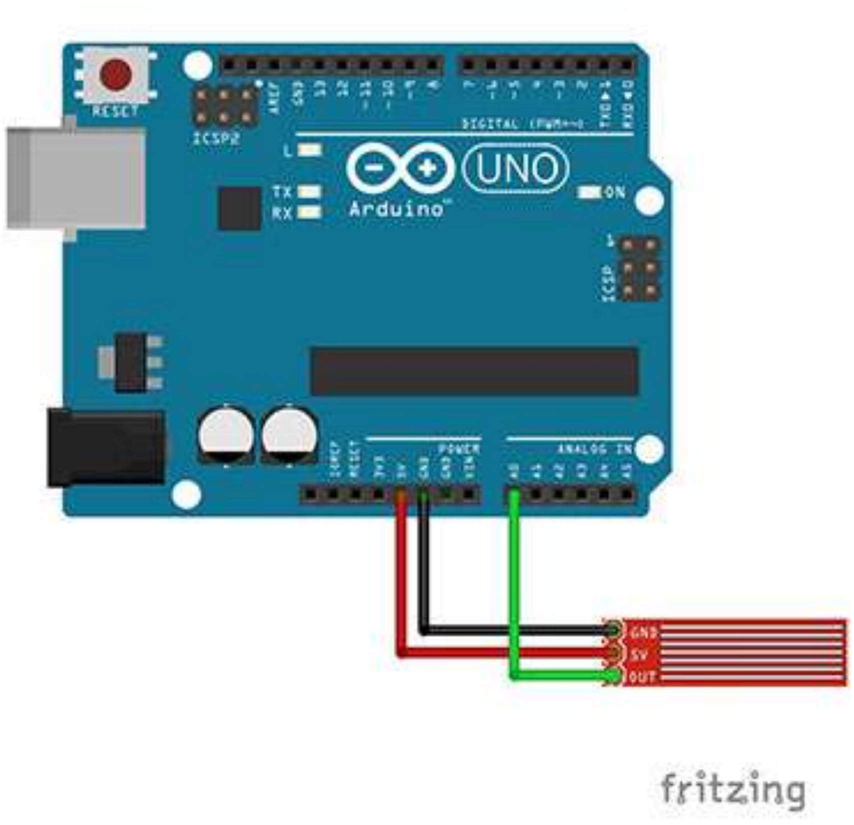
Specifiche del sensore Funduino water sensor

- Color: Rosso
- Power Supply: 3.3V or 5V
- Working Current: < 20mA
- Output Voltage: 0~2.3V (con il sensore completamente immerso in acqua)
- Interface Definitions: 1: pin segnale, 2: pin GND, 3: pin positivo (+5v)
- Service Life: un anno approssimativamente
- Sensor Type: Analogico
- Weight: 8.0g
- Packaging Dimensions: 120.0 x 78.0x 10.0mm

Queste specifiche ti fanno rapidamente capire che tale sensore è stato studiato per essere utilizzato con microcontrollori come arduino accettando come alimentazione dai 3,3v ai 5v ed assorbendo meno di 20mA.

Dalle specifiche il funduino water sensor raggiunge in output i 2,3v se completamente immerso in acqua, questo significa che in termini numerici leggerai al massimo 471 come valore.

I collegamenti del water sensor



Lo sketch usato per il test

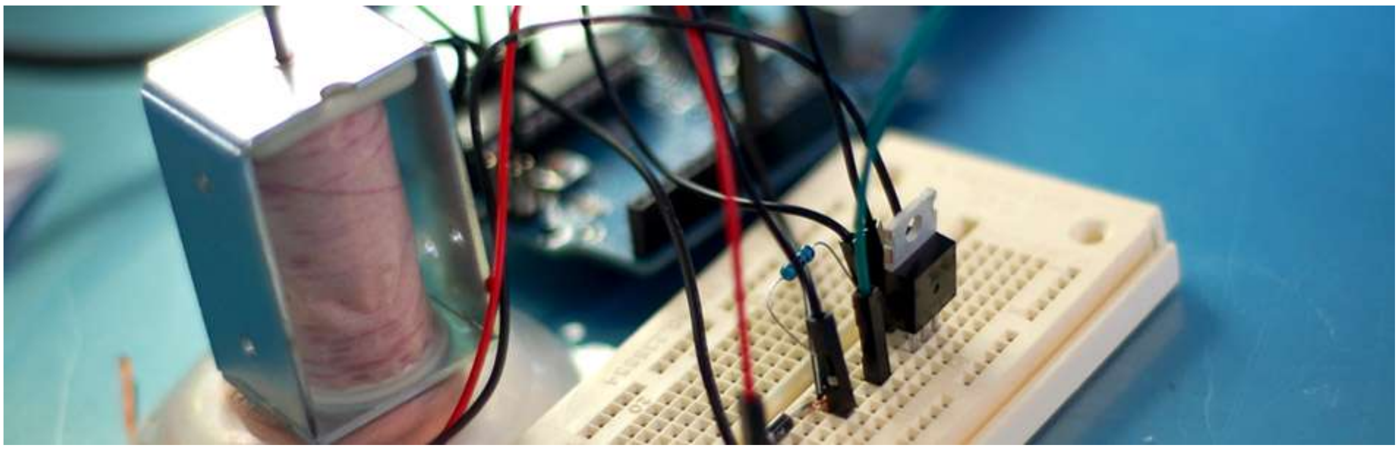
```
/*
  AnalogReadSerial
  Reads an analog input on pin 0, prints the result to the serial monitor.
  Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and gr

 This example code is in the public domain.
 */

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(100);        // delay in between reads for stability
}
```

High-Power Control: Arduino + TIP120 Transistor



Up until now, we have talked about working with a lot of low-power devices.

Sensors, LEDs, ICs, and the like are all capable of being powered directly from your Arduino, but as many awesome 5 and 3.3v components as there are, eventually you will find yourself holding a 12v solenoid, motor, or light and wondering "How the heck am I supposed to control this from my Arduino?"

Well today we are going to talk about doing just that from a magical device know as a transistor, specifically the TIP120 Darlington Transistor.

The reason I'm covering this particular transistor is because it is readily available, and you can usually pick one up from Radio Shack, Adafruit or other local parts store in a jam, but you can use any NPN darlington transistor like the BD651 exactly the same way.

How this works

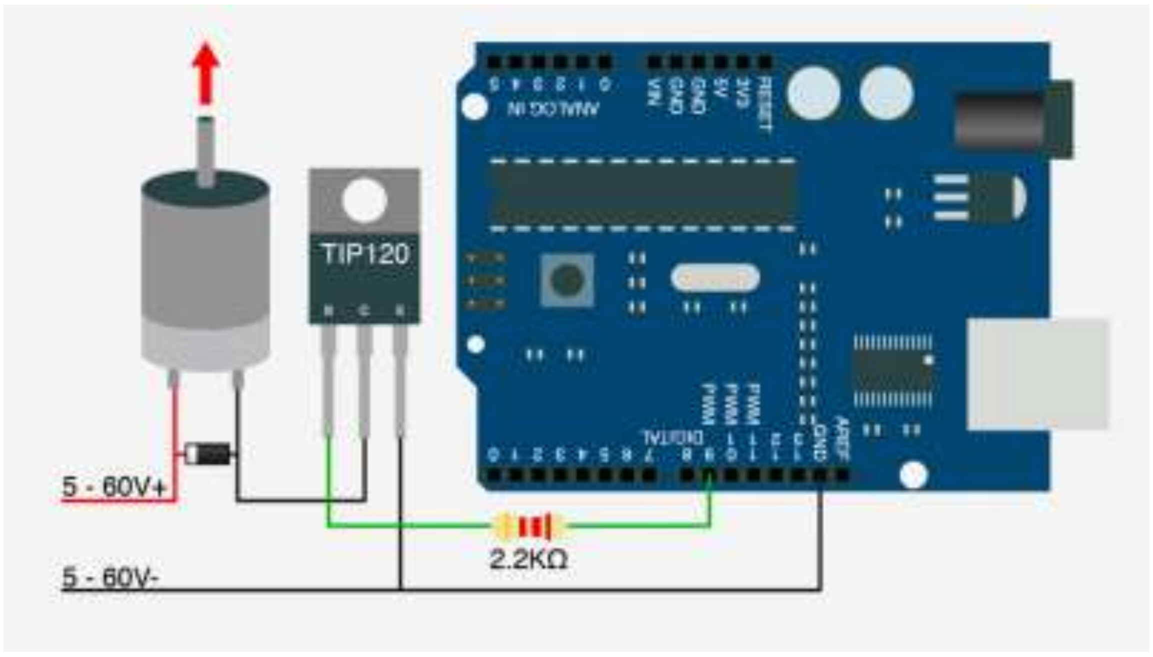
WARNING: I am about to simplify the crud out of this, so beware... it is here in an attempt to explain, in simple terms, what is going on.

If you don't know transistors at all, they are 3 lead components that have 2 simple functions, to switch or amplify (in this example it is setup as a switch).

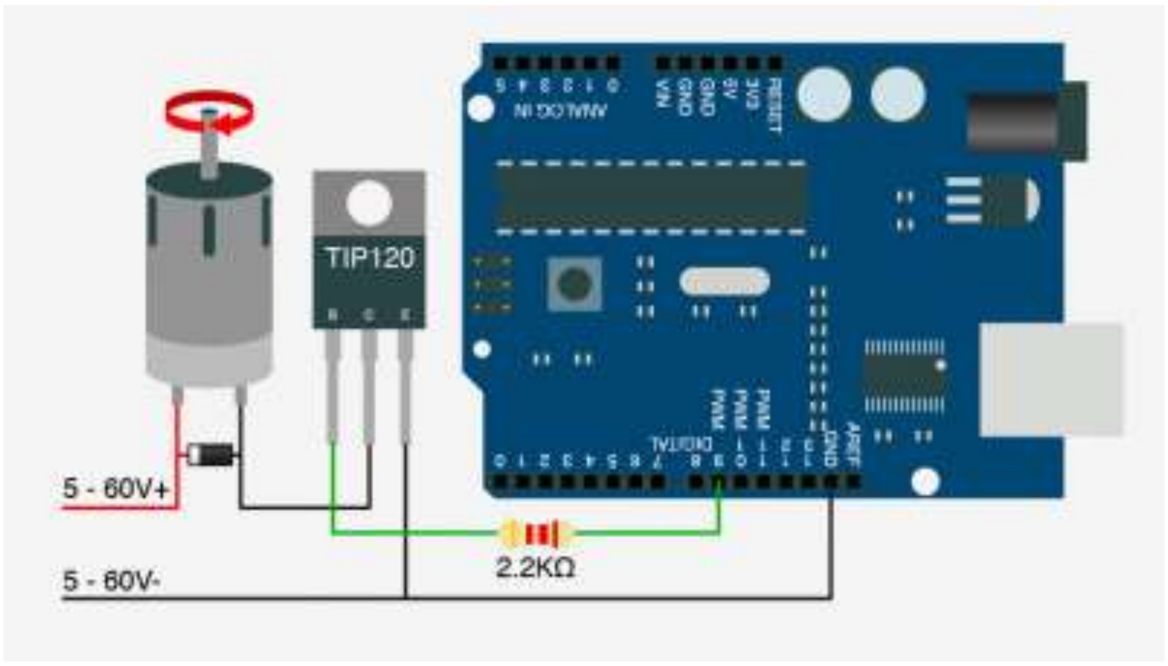
You basically have an In called the Collector, an Out called the Emitter, and a Control called the Base. When you send a HIGH signal to the base (control pin), the transistor switches and allows current to flow from the collector (in) to the emitter (out).

So we connect it so that our motor, solenoid or light is connected to V+ but not ground (V-). Ground is connected to the transistor's collector.

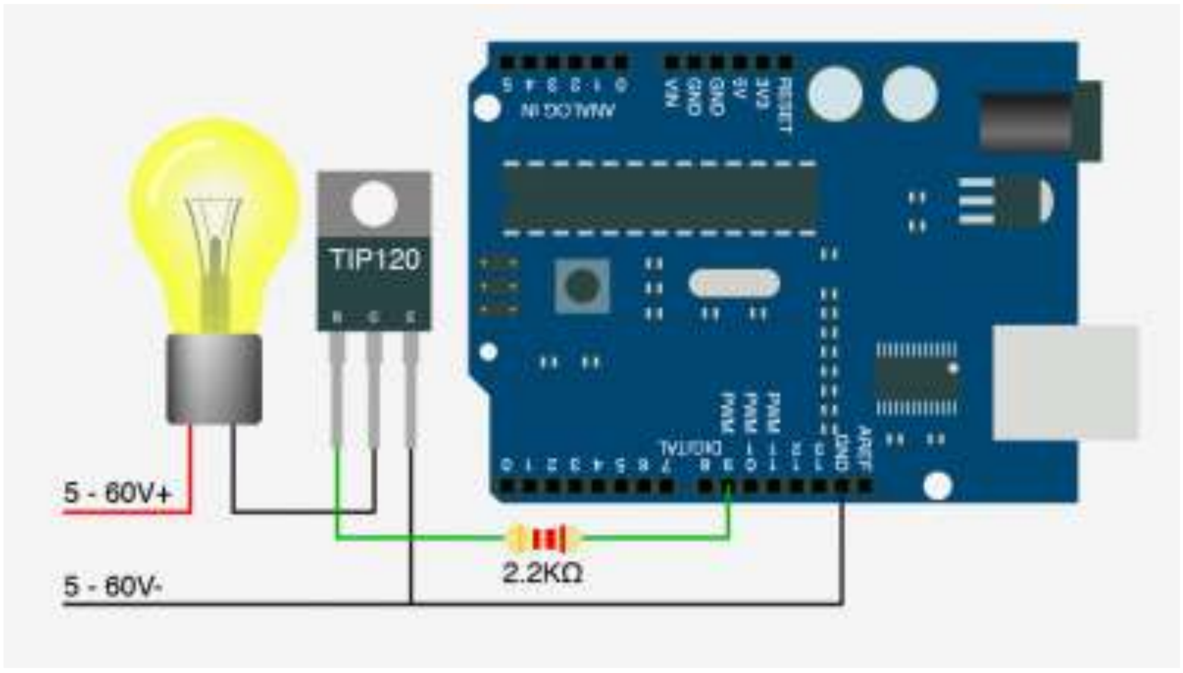
When our arduino sends a HIGH signal to the transistor's base, it switches the transistor (connecting the collector and emitter) and completes the circuit for the motor, solenoid, or light.



TIP120
5 - 60V+
5 - 60V-
2.2KΩ
ANALOG IN
DIGITAL
PWM
GND
AREF
RESET



TIP120
5 - 60V+
5 - 60V-
2.2KΩ
ANALOG IN
DIGITAL
PWM
GND
AREF
RESET



TIP120
5 - 60V+
5 - 60V-
2.2KΩ
ANALOG IN
DIGITAL
PWM
GND
AREF
RESET

Hooking it up / What's the diode used for?

This circuit is pretty simple. This type of transistor is switched by current and not voltage, so we need to make sure to supply the correct current to the base to switch it, so a resistor is connected from the Arduino to the base to limit the current to the proper amount.

You can see that in 2 of the 3 illustrations, there is a diode parallel to the device we are powering. Any time you are powering a device with a coil, such as a relay, solenoid, or motor, you need this guy, and don't leave home without it.

What happens is when you stop powering the coil, a reverse voltage, up to several hundred volts, spikes back. This only lasts a few microseconds, but it is enough to kill our transistor.

So this diode (only allows current to pass one way) is normally facing the wrong direction and does nothing. But when that voltage spikes comes flowing the opposite direction, the diode allows it to flow back to the coil and not the transistor. We will need a diode fast enough to react to the kickback, and strong enough to take the load.

A rectifier diode like the 1N4001 or SB560 should do the job. If you are looking for extra protection you could use an optoisolator between the Arduino and the transistor. An optoisolator optically isolates both sides (high and low power) of the circuit so the high-voltage can not possibly come back to the microcontroller.

Just make sure that protection diode is facing the correct way (stripe facing the V+ of device). If it is facing the wrong direction, the device you are trying to power will not work as the diode will just allow the current to bypass it.

Limitations

Transistors like the TIP120 are really great for controlling high-power devices from your microcontroller, but they do have some limitations. This current configuration is only useful for switching DC current, so don't try this with an AC source, also transistors have both a voltage and an amperage/current limitation.

The TIP120 can handle switching up to 60V, and the amperage is limited to 5A, or up to 8A pulses of 300µs. I have managed to blow out one of these with a 5A load because of heat. Actually anything over a few amps, especially when the current is constant (like in a motor) and not short pulses, I would recommend using a heat-sink.
I usually just solder a bent pice of metal to the back, just something to help dissipate the heat. Just note, if you are using more than one of the TIP120s, you can not solder them to the same heat-sink as the back is connected to the base of the transistor, not the emitter. If you need to switch more than 5A or AC, I would look at using a relay instead.