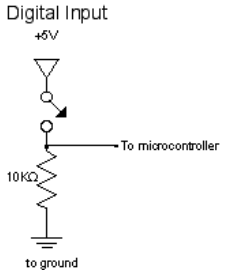
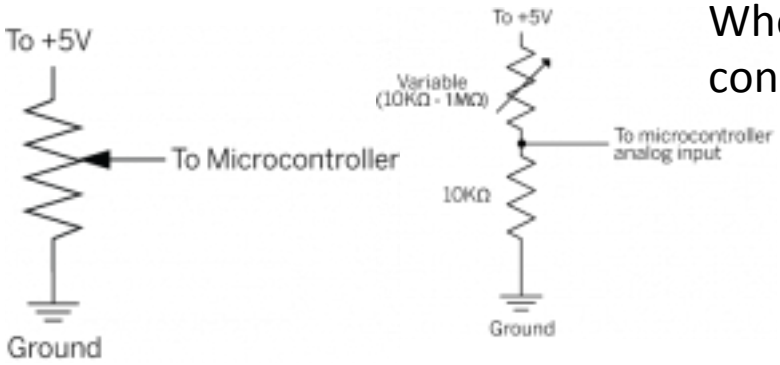
Review

Analog vs. Digital



Digital inputs have two states: off and on.
If voltage is flowing, the circuit is on.
If it's not flowing, the circuit is off.

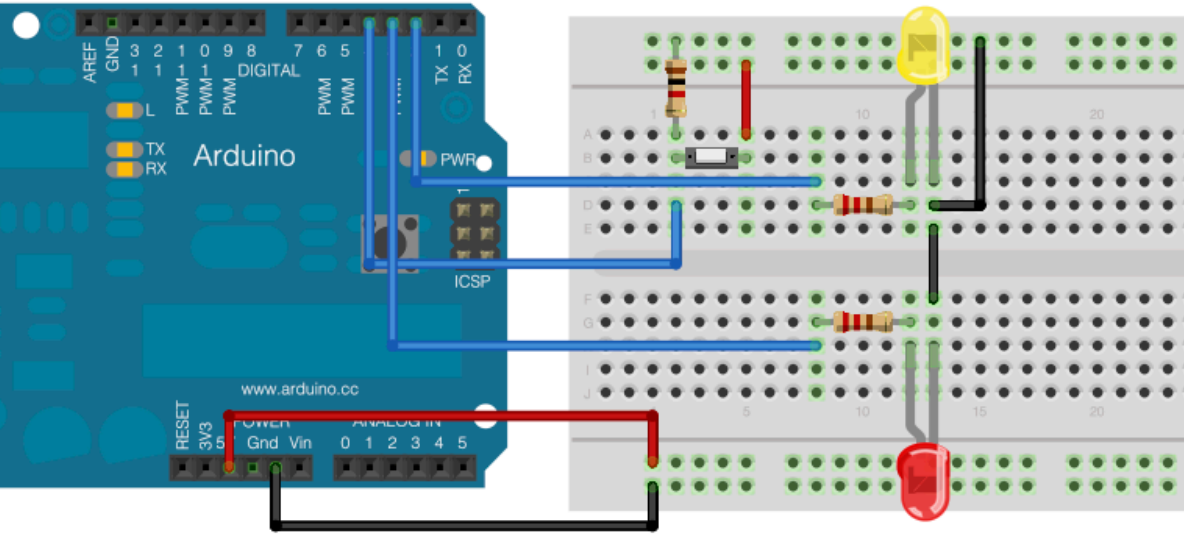


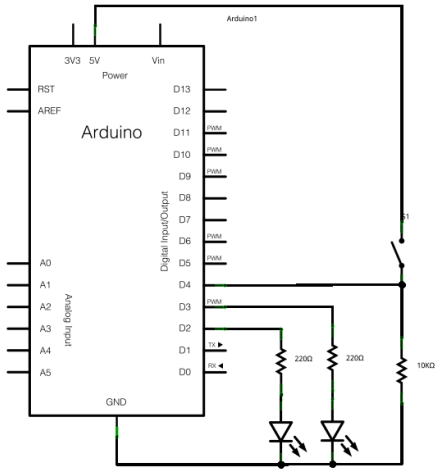
When we want to measure variably changing conditions like this, we need analog inputs.

Examples: thermistors, photocells, force sensing resistors, flex sensors, …

A digital circuit

Here's a circuit for a program that reads the digital input on pin 4.
Then it turns on the LED on pin 2 if the input is high (i.e. the switch is on),
or turns on the LED on pin 3 is the input is low (the switch is off):



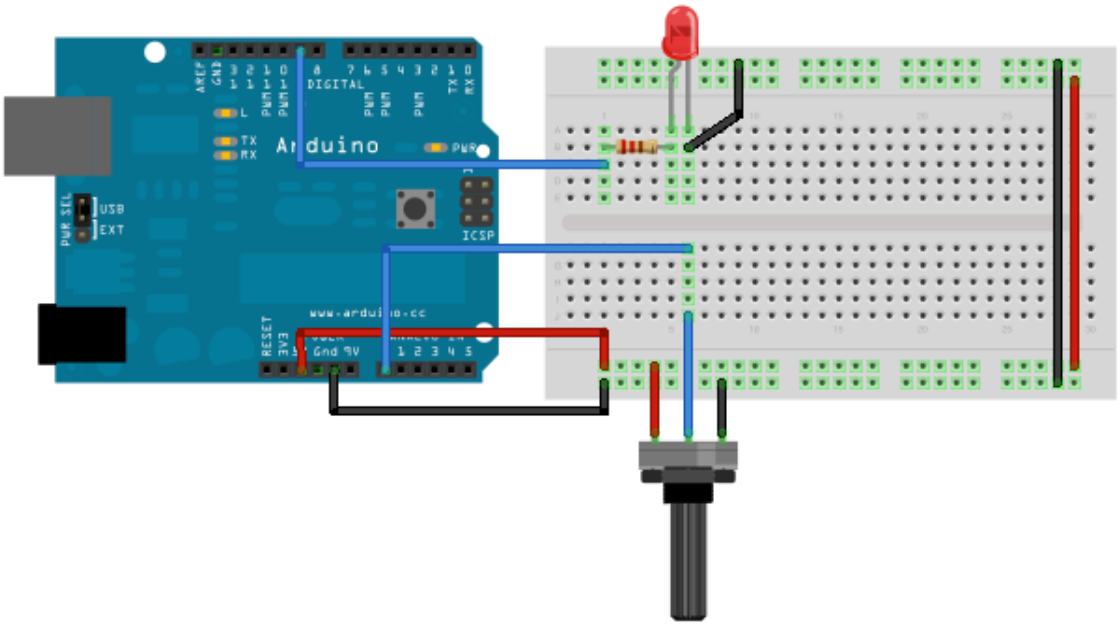


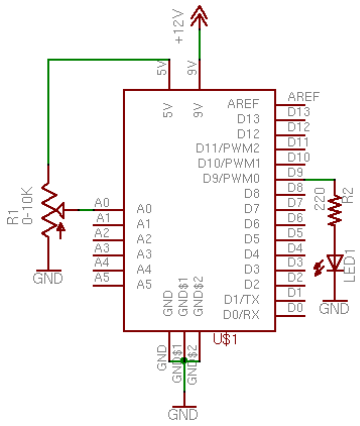
The switch code

```
// declare variables:
int switchPin = 4;      //  digital input pin for a switch
int yellowLedPin = 2;   //  digital output pin for an LED
int redLedPin = 3;      //  digital output pin for an LED
int switchState = 0;    // the state of the switch

void setup() {
  pinMode(switchPin, INPUT);       // set the switch pin to be an input
  pinMode(yellowLedPin, OUTPUT);   // set the yellow LED pin to be an output
  pinMode(redLedPin, OUTPUT);      // set the red LED pin to be an output
}

void loop() {
  // read the switch input:
  switchState = digitalRead(switchPin);

  if (switchState == 1) {
    // if the switch is closed:
    digitalWrite(yellowLedPin, HIGH);    // turn on the yellow LED
    digitalWrite(redLedPin, LOW);       // turn off the red LED
  }
  else {
    // if the switch is open:
    digitalWrite(yellowLedPin, LOW);   // turn off the yellow LED
    digitalWrite(redLedPin, HIGH);     // turn on the red LED
  }
}
```

An analog circuit

When you run this code, the LED should dim up and down as you turn the pot, and the value of the pot should show up in the debugger pane.





Analog input code

```
int potPin = 0;    // Analog input pin that the potentiometer is attached to
int potValue = 0;   // value read from the pot
int led = 6;    // PWM pin that the LED is on.  n.b. PWM 0 is on digital pin 9

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  // declare the led pin as an output:
  pinMode(led, OUTPUT);
}

void loop() {
  potValue = analogRead(potPin); // read the pot value
  analogWrite(led, potValue/4);  // PWM the LED with the pot value (divided by 4 to fit in a byte)
  Serial.println(potValue);      // print the pot value back to the debugger pane
  delay(10);                     // wait 10 milliseconds before the next loop
}
```

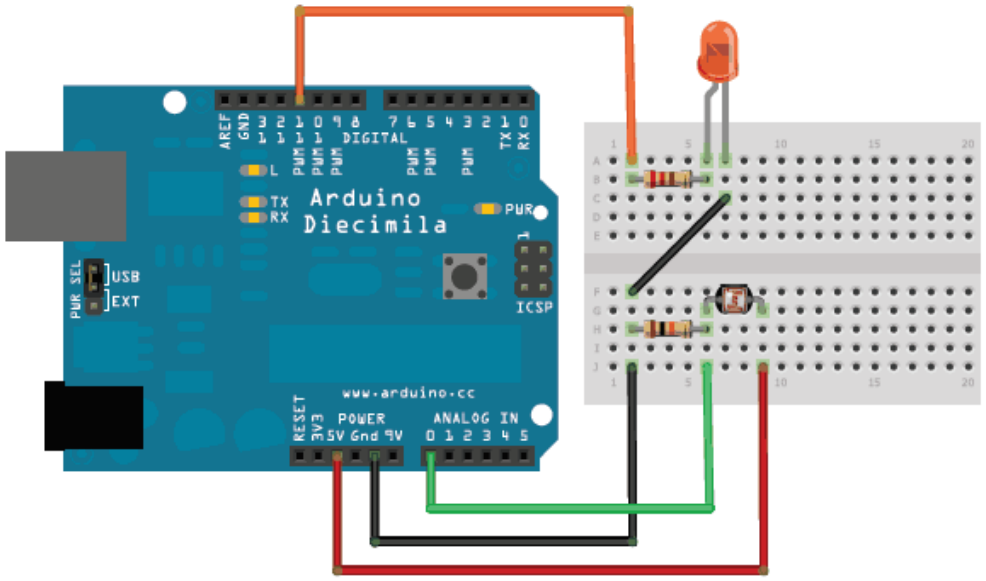
Another sensor, and the map() function

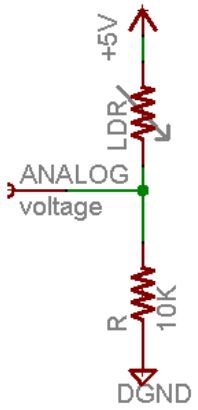
- Let's replace the potentiometer with a photocell (or any variable resistor)
- Add a fixed resistor? (voltage divider)
- Look at the serial monitor: find out the RANGE of the analog sensor - note the maximum and minimum values
  - Before, a potentiometer was giving us analog input values of 0 – 1023, the full range of the analogRead() function;
  - Now, we must modify the code using the map() function
  - Map the input (min – max sensor reading) to the output (0-255 – because 0 – 255 is the range of Arduino's analogWrite() function

Map the incoming values:

```
potValue = analogRead(potPin); // read the pot value
Serial.println(potValue);      // print the pot value back to the debugger pane
int brightness = map(potValue, 600, 950, 0, 255);
analogWrite(led, brightness);
```

Photocell circuit





Code for one photocell and one LED

```
int potValue = 0;   // value read from the pot
int led = 6;    // PWM pin that the LED is on.  n.b. PWM 0 is on digital pin 9

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  // declare the led pin as an output:
  pinMode(led, OUTPUT);
}

void loop() {

  potValue = analogRead(potPin); // read the pot value
  Serial.println(potValue);      // print the pot value back to the debugger pane
  int brightness = map(potValue, 600, 950, 0, 255);
  analogWrite(led, brightness);

  delay(10);                     // wait 10 milliseconds before the next loop
}
```

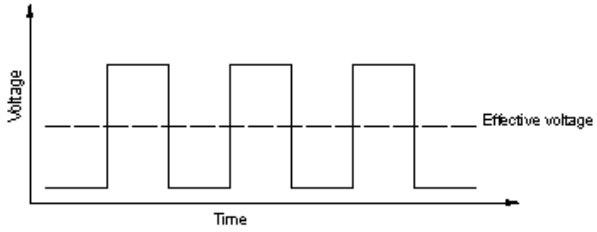
NOTE: you may have to modify the map() function, depending on your sensor

Excersize

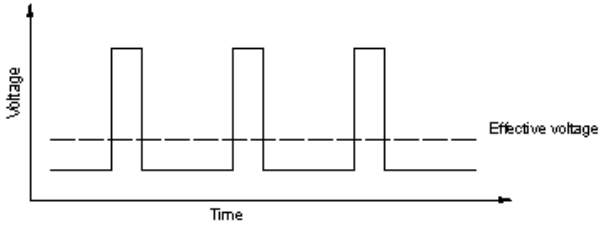
- Use the map() function to make the LED get less bright as the photocell gets darker, and brighter with MORE light…

Analog Output: Pulse Width Modulation (PWM)

- Most microcontrollers “fake” an analog voltage output by producing a series of voltage pulses at regular intervals, and varying the width of the pulses (PWM)
- DUTY CYCLE = ratio of the time the pin is HIGH to the time it is LOW

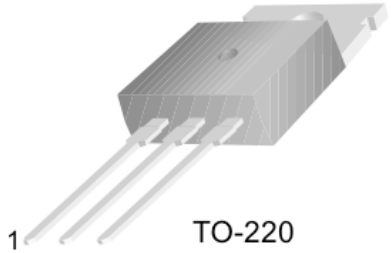


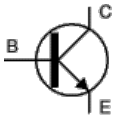
Here the duty cycle is 50%, making the voltage output about half



The duty cycle is less because it is OFF for longer

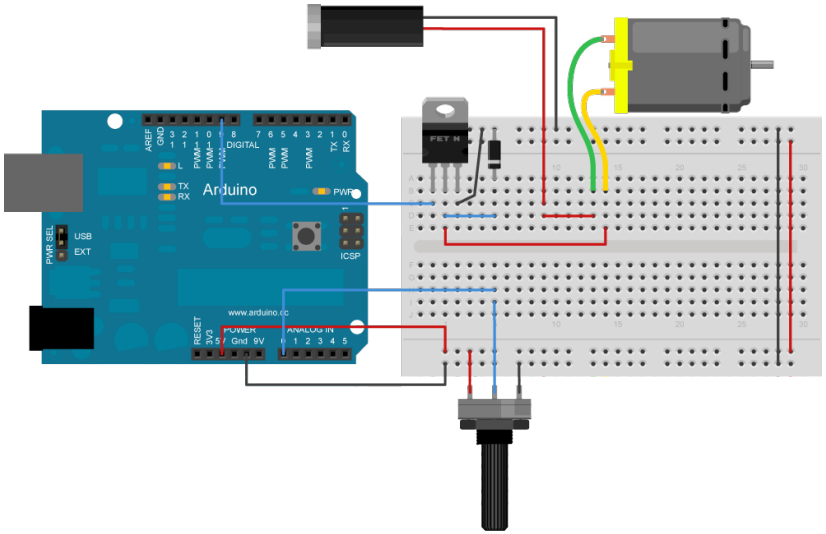
Powering a high-current load using a transistor





1.Base 2.Collector 3.Emitter

An electronic switch



Transistor circuit code

```
//try changing the speed of the motor or the intensity
//of the lamp using the potentiometer. Try this code:
 const int potPin = 0;           // Analog in 0 connected to the potentiometer
 const int transistorPin = 9;    // 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(9, potValue);
 }
```