Building the Light Meter Tutorial

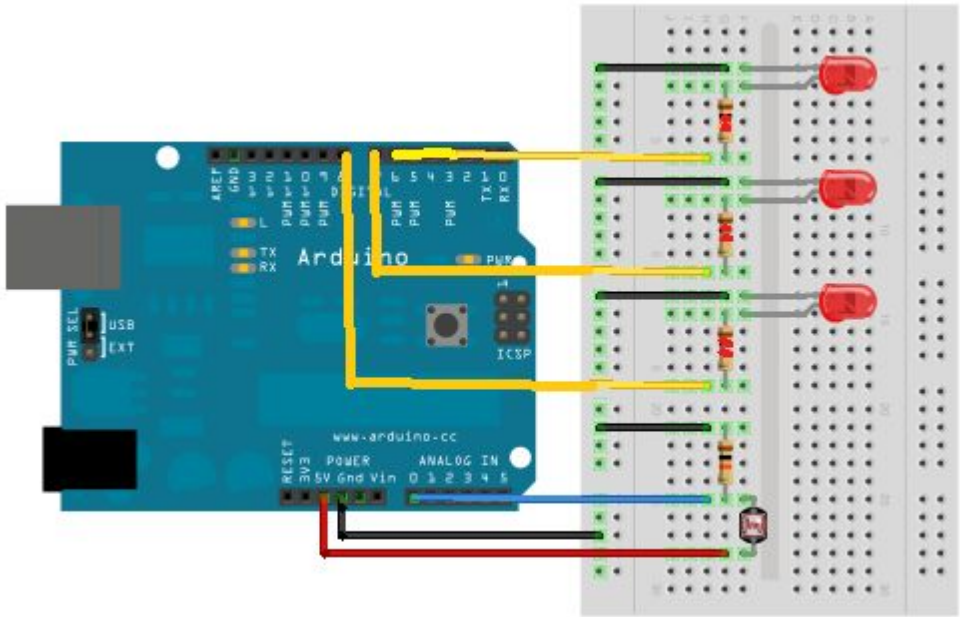
You should have a kit containing all the materials you need to build a prototype light meter and program it.

Activity - Open up your kits now and identify all the components, making sure you have everything.

List of materials per group:

- 1 x Arduino Uno or Mega
- 1 x USB cable to interface computer with Arduino
- 1 x computer with Arduino IDE (Arduino programming software)
- 1 x LDR (Light Dependent Resistor or photoresistor)
- 1 small breadboard
- 1 x 10k resistor
- 3 x LEDs (one of each color - Red, Green, Yellow)
- 3 x 220R resistors
- 10 male to male breadboard jumper wires
- 4 male to female breadboard jumper wires

The final light meter circuit you will end up with will look something like this:



If you think that this looks complicated, don't worry, we will have a working sample out the front of the class for you to look at any time you need to.

You might be aware that we will be using an **Arduino** to **build a circuit** and then **program it** to **detect** light intensity and **turn on** LED's to indicate the varying intensities of light detected.

But what is an Arduino…?

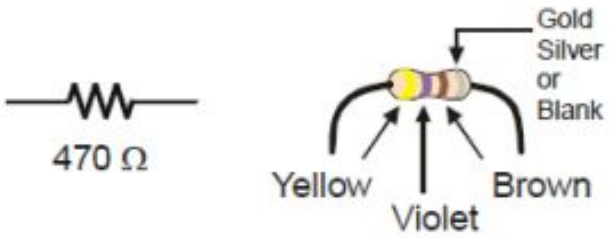
An Arduino is an open-source **microcontroller-based** kit for building digital circuits and devices that can sense and control objects in the real world.

What's a Microcontroller?

It's a programmable device that is designed into your cars, watches, machinery,mobile phone, calculator, clock radio, planes, etc. The microcontroller has been programmed to sense when you press a button, make electronic beeping noises, and control the device's digital display. They are also programmed to read sensors, make decisions, and orchestrate devices that control moving parts.

Building the LED indicator lights circuit

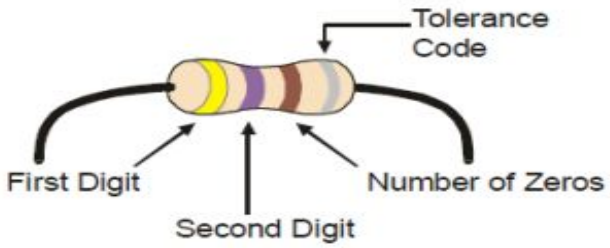
The **resistor** - a passive component that resists the flow of electric current. Each resistor has a value that indicates how much it resists current flow. This resistance value is called the ohm, and the sign for the ohm is the Greek letter omega: Ω or R.



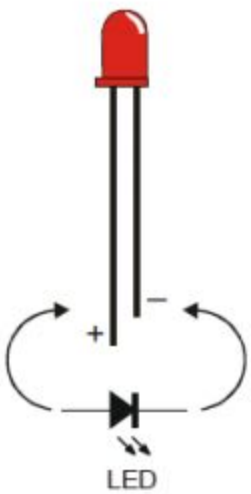
Activity - Use the table below and picture above to figure out the color code for the 220 Ω resistors you will need for the indicator lights.

Resistor Color Code Values

Digit	0	1	2	3	4	5	6	7	8	9
Color	black	brown	red	orange	yellow	green	blue	violet	gray	white

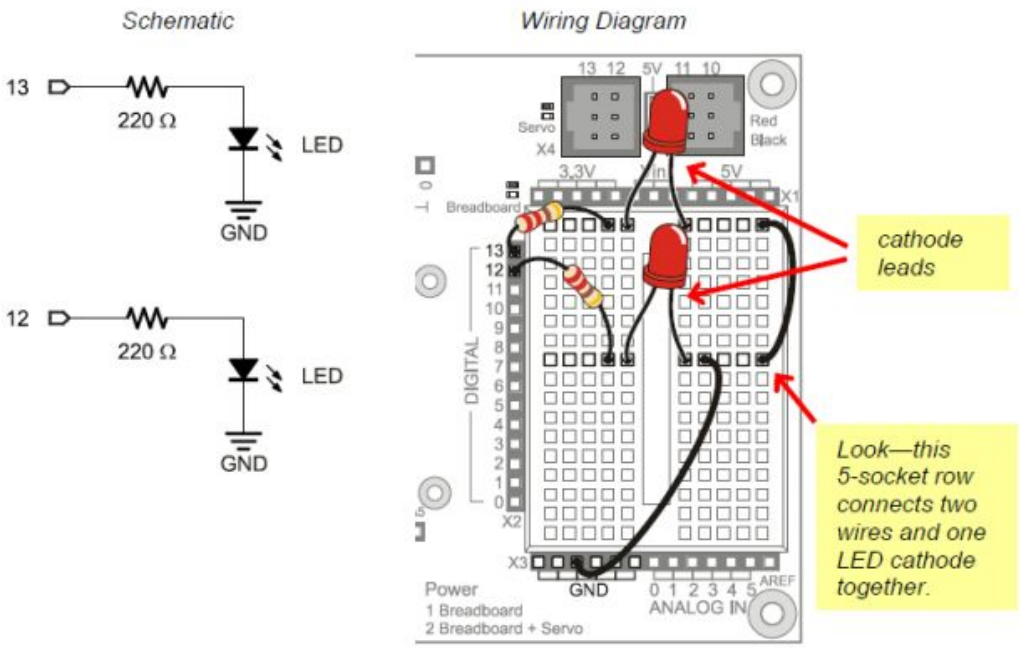


A **diode** is a one-way current valve, and a light-emitting diode (LED) lights up when a current passes through it. Since an LED is a one-way current valve, you have to make sure to connect it the right way for it to work.



Usually, the longer lead is connected to the LED's anode, and the shorter lead is connected to its cathode. But sometimes the leads have been clipped to the same length, so it's best to always look for the flat spot on the case. If you plug an LED in backwards, it will not hurt it, but it won't emit light until you plug it in the right way.

The (usually) white board with lots of square sockets in it is called a solderless **breadboard**. This breadboard has 17 rows of sockets. In each row, there are two five-socket groups separated by a trench in the middle. All the sockets in a 5-socket group are connected together underneath with a conductive metal clip. So, two wires plugged into the same 5-socket group make electrical contact.



Activity - Build the circuit shown above

Make sure your LEDs are connected correctly, ie. cathodes and anodes in the correct polarity.

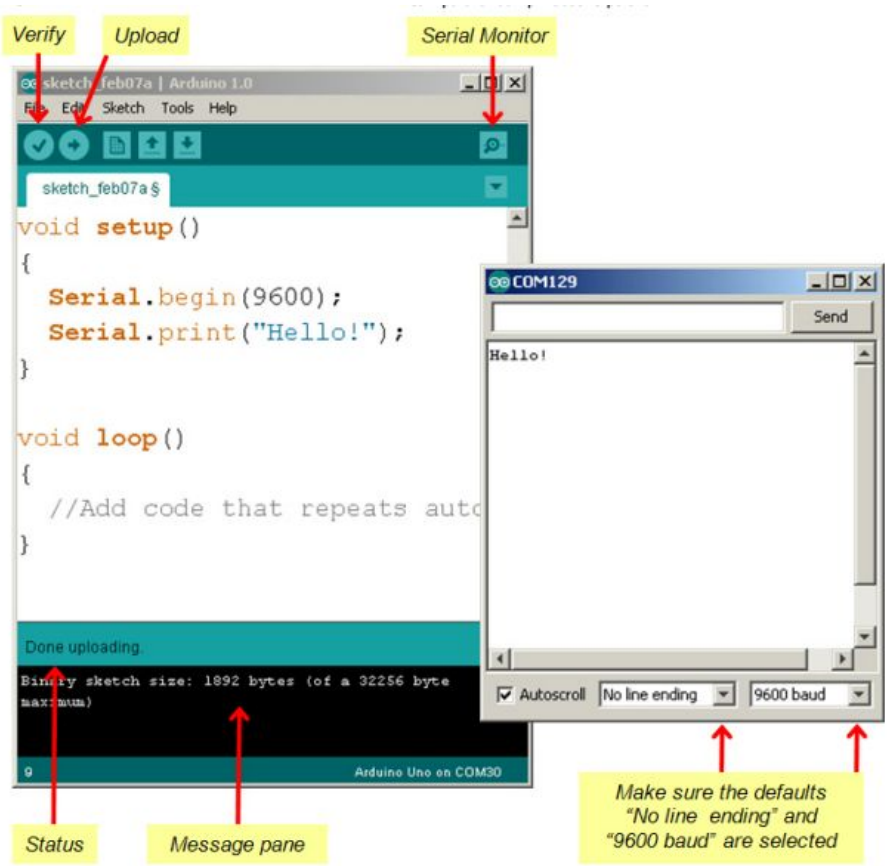
Programming the LED indicator lights circuit

You computers should have the Arduino programming software already installed. You will use the Arduino Development Environment software to write programs (called sketches). Arduino sketches are written by adding C and C++ programming language statements to a template.

Activity - Open your Arduino software and carefully type in the code:

```
void setup()
{
        pinMode(13, OUTPUT);
}

void loop()
{
        digitalWrite(13, HIGH);
        delay(500);
        digitalWrite(13, LOW);
        delay(500);
}
```



Go into the **Tools** menu and make sure that the correct Arduino board & serial port is selected.

Click the Verify button to make sure your code doesn't have any typing errors.

Look for the "Binary sketch size" text in the message pane. If it's there, your code compiled and is ready to upload to the Arduino.

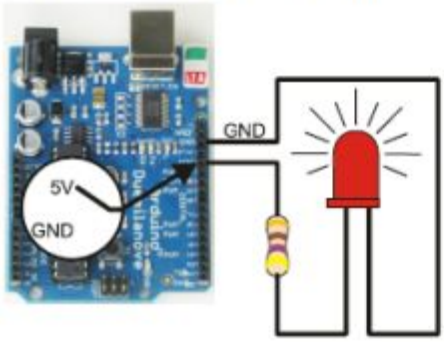
If there's a list of errors instead, it's trying to tell you it can't compile your code. So, find the typing mistake and fix it.

Click the Upload button. The status line under your code will display "Compiling sketch...," "Uploading...," and then "Done uploading."

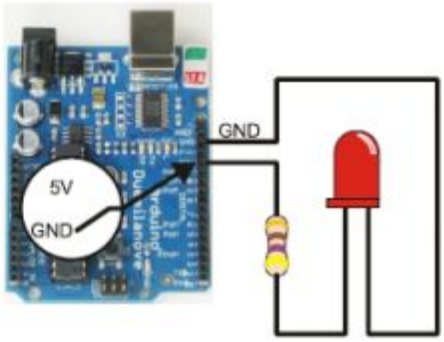
Your program should now be running on the Arduino, the LED connected to pin 13 should be blinking.

Use File → Save to save your sketch. Give it the name **Blinky**

```
digitalWrite(13, HIGH);
```




```
digitalWrite(13, LOW);
```



Activity - Modify the circuit & sketch you did above so that pins 6, 7 and 8 are used to make the 3 LEDs blink, **all at the same time**.

Activity - Modify the circuit & sketch you did above to make 3 LEDs blink, **one at a time**.

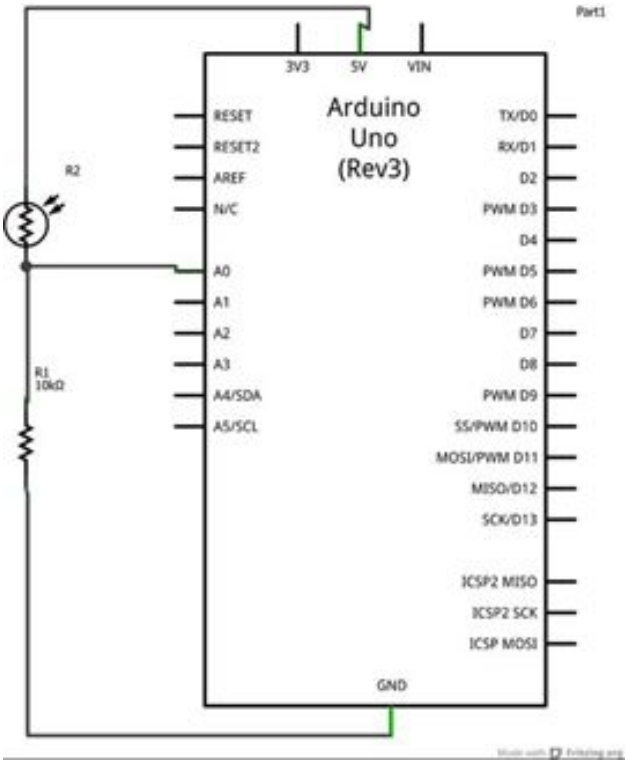
Building and programming the LDR light detector circuit

We will use an LDR (Light dependent Resistor, also known as a photoresistor) to change the voltage across a circuit. The intensity of light shining onto an LDR changes it's resistance value.

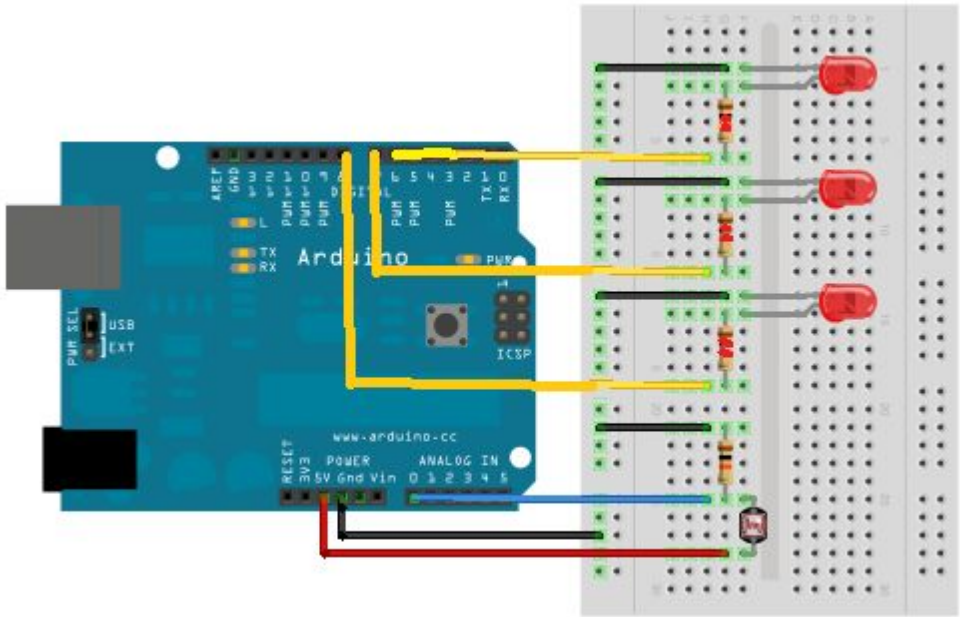
We use a circuit to convert the varying resistance to a voltage that the Arduino can measure. We can then read this changing voltage using the Arduino and print those measurements to the Serial port monitor (on the computer screen) and light up LED lights to indicate the levels of light being detected. The more light that passes through your filtered water and shines onto the LDR light detector, the lower the resistance of the LDR, and hence the greater the voltage level that the Arduino will read.

We will be writing a program in the Arduino programming language that will be downloaded onto the Arduino board to be executed in an endless loop. The program will tell the Arduino microcontroller to read inputs (in this case a varying analog voltage level proportional to light intensity) and to indicate output by lighting up a series of LED lights.

Activity - Add to the LED blink circuit that you made earlier, the LDR part of the light meter circuit shown below.



The completed circuit should look something like:



Activity - Carefully type in the Arduino sketch below. Run it, open the serial monitor and observe what happens when you alter the level of light shining on the LDR.

```
int sensorReading;              // Declaring an integer variable

void setup()

{
  Serial.begin(9600);            // starts the serial port monitor
  pinMode(6,OUTPUT);             // 1st LED
  pinMode(7,OUTPUT);            // 2nd LED
  pinMode(8,OUTPUT);            // 3rd LED

}

void loop()

{
  sensorReading=analogRead(0);      // read voltage level & store in variable

  if (sensorReading>700)                // is the value read over 700 ?
  {
        digitalWrite(6,HIGH);           // yes, then turn on pin 6 (LED 1)
  }

  else digitalWrite(6,LOW);             // no, turn off pin 6 (LED 1)
  Serial.println(sensorReading);        // output voltage reading onto the

  delay(1000);                  // 1 second delay (wait 1000 milliseconds before proceeding)
}
```

Activity - Modify the program so that the LEDs come on at different light thresholds.

You now have made a simple and effective light meter…!!