Feb.4, 2019

Breadboard

- Dots in middle two parts (five rows each, rows labelled A to J)
    - vertically connected
    - between separator is disconnected
- The first two rows with the sign + and - (same as the bottom row)
    - each row horizontally connected
    - But + and - row is NOT connect with each other

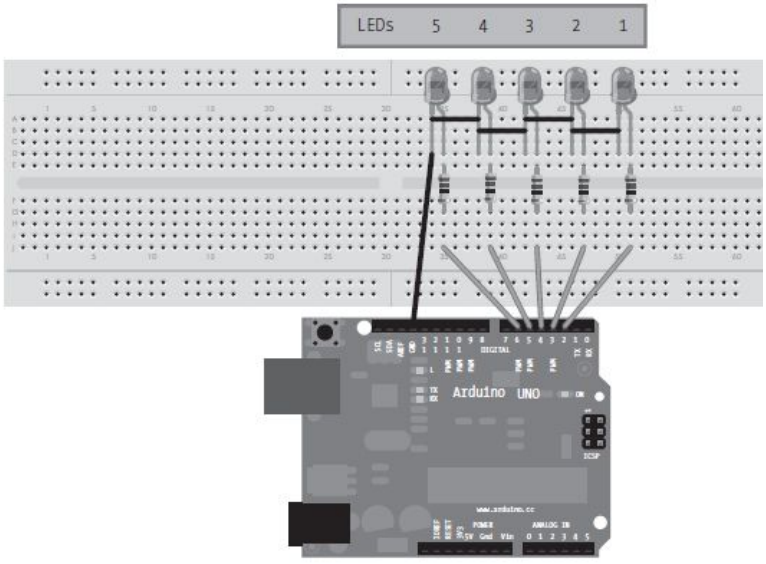
Ohm's Law

V= I x R
A resistor R, Vs flow into R, and Vf flow out from R.
Vs= 5V , Vf= 1.7V
10mA = 0.01A
R = (Vs-Vf)/I = (5-1.7)/0.01 = 330 Ω
Arduino recommended higher operation resistor 560Ω.
We will use 1000Ω=1kΩ resistor.

LED light

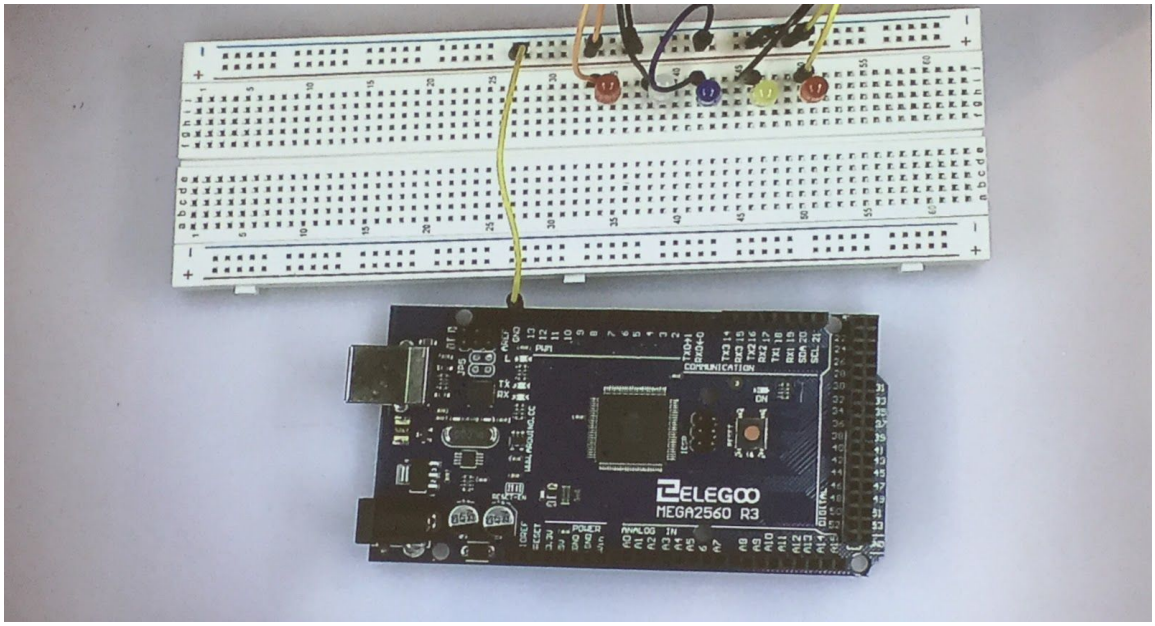
LED light has the two side called anode(+) and cathode(-).
Anode side is longer leg.
Cathode side is shorter leg.

LED_Wave_Project

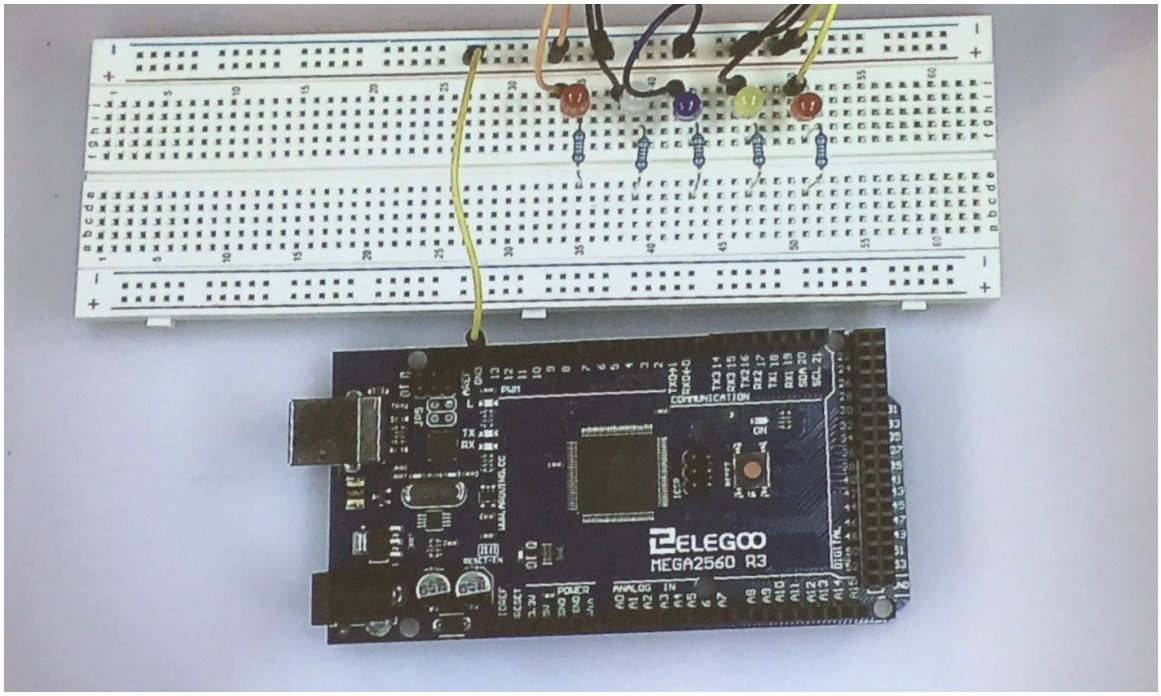


First connect the light with shorter side on the left and longer side on the right onto the breadboard. Then, use the wires connect light to the ground (Arduino and USB cable). Next

steps, according to the image, we can connect the 5 LEDs using wire by connecting negative legs.
A better way to do: ground wire connect to the top row with labelled +. Since the +row on the top is horizontally connected on the breadboard, we could connect each LEDs by wires on the + row. Make sure the wire connect each LED and the + row is on the same vertical column with the LED light. (since the LED connect at middle, row labeled fghij, it is vertically connected).



Then, connect the resistors onto the breadboard. For each resistors, we connect them one side on the same column with the LED light, the other side on the row below the separator, in order to make it easier to connect onto the Arduino-USB cable.



Then connect the wire with the Arduino-USB cable on its PWM capable pin3-7.
Then we finished connect the hardware, connect it with the laptop. Deal with software, sketch.

Sketch
LED_wave_v1:

```
void setup(){
        pinMode(3, OUTPUT);
        pinMode(4, OUTPUT);
        pinMode(5, OUTPUT);
        pinMode(6, OUTPUT);
        pinMode(7, OUTPUT);
}
void loop(){
        digitalWrite(3, HIGH);
        delay(500);
        digitalWrite(3, LOW);
        digitalWrite(4, HIGH);
        delay(500);
        digitalWrite(4, LOW);
        digitalWrite(5, HIGH);
        delay(500);
        digitalWrite(5, LOW);
        digitalWrite(6, HIGH);
        delay(500);
        digitalWrite(6, LOW);
        digitalWrite(7, HIGH);
        delay(500);
        digitalWrite(7, LOW);
}
```

Note for v1:
Check the code first before upload to Arduino in order to avoid fire the Arduino cable.
Check the processor setup is Mega 2560, and Port is empty before connect the Arduino to laptop.

Sketch improvement with global variable.
Version 2:
LED_wave_v1:

```
int d=500 //delay 500ms (define global variable d)
void setup(){
        pinMode(3, OUTPUT);
        pinMode(4, OUTPUT);
        pinMode(5, OUTPUT);
```

```
        pinMode(6, OUTPUT);
        pinMode(7, OUTPUT);
}
void loop(){
        digitalWrite(3, HIGH);
        delay(d); //change the delay (500) to delay(d)
        digitalWrite(3, LOW);
        digitalWrite(4, HIGH);
        delay(d);
        digitalWrite(4, LOW);
        digitalWrite(5, HIGH);
        delay(d);
        digitalWrite(5, LOW);
        digitalWrite(6, HIGH);
        delay(d);
        digitalWrite(6, LOW);
        digitalWrite(7, HIGH);
        delay(d);
        digitalWrite(7, LOW);
}
```

Note :
since the Arduino sketch are using language similar as C language, so this version is doing the exactly same thing as the version 1.
When change d (delay time) to a smaller number, like 100, it reduce the time of the period between two light wave. When a change to 1, people cannot tell the light is changing.

Sketch Improvement with For Loop.

Version 3:
LED_wave_v2:

```
int d=500
void setup(){
        pinMode(3, OUTPUT);
        pinMode(4, OUTPUT);
        pinMode(5, OUTPUT);
        pinMode(6, OUTPUT);
        pinMode(7, OUTPUT);
}
void loop(){
        for(int a = 3; a < 8; ++a){
                digitalWrite(a, HIGH);
                delay(d);
```

```
            digitalWrite(a, LOW);
        }
}
```

Note:
Save sketch before compile and upload to Arduino.
Make sure you current and voltage are correct in order to avoid fire the hardware.

Varying LED Brightness with Pulse-width Modulation

When we do digitalWrite to HIGH or LOW, it turn to be LED turn on and off, we now consider to control the LED brightness by analog value.
By using another for loop, we can introduce a new variable to control the brightness of the LEDs.
The LED brightness can be any number between 0 - 255.

Sketch showing one LED light
LED_wave_v2:

```
int d=500
void setup(){
        pinMode(3, OUTPUT);
        //pinMode(4, OUTPUT);
        //pinMode(5, OUTPUT);
        //pinMode(6, OUTPUT);
        //pinMode(7, OUTPUT);
}
void loop(){
        for(int a = 0; a < 256; ++a){
            analogWrite(3, a);
            delay(d);
        }
        for(int a = 255; a >= 0; --a){
            analogWrite(3, a);
            delay(d);
        }
        delay(200);
}
```

Note:
The LED on digital pin 3 exhibits cycle increases and decreases. The LED turn on, increasing in brightness until fully lit, and then reverse. This also can be done on five LEDs as well.