Programming the Physical World
Stop Light Simulator

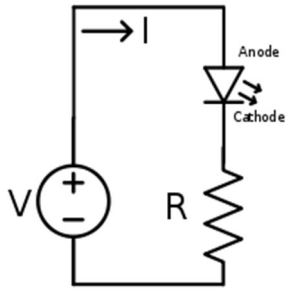
In this lab we are going to write a program that interfaces with computer hardware to make something physical occur.

For this activity we will require the following:

1. Arduino Uno
2. Four male/male jumper wires
3. Solderless breadboard
4. Red LED, Yellow LED, Green LED
5. 330 Ω resistor
6. USB cable (A – B)

The hardware we will be using will be the Arduino which has 13 Digital IO (Input Output) pins and 6 Analog input pins which can be easily programmed to send output signals to or receive input signals.

For this activity, we will be using pins 8, 9 and 10 as well as one of the Ground pins (labeled GRD). Here is a diagram of an LED setup. To get the LED to light we need approximately 2.1 volts across it. The purpose of the resistor is to limit the current through the LED. Pin 8, 9, and 10 are capable of supplying 5 volts. If we do not use a current limiting resistor, the LED will light brightly for a while and then go dark. For a nominal brightness we want about 10 mA to flow through the LED. This means a 330 Ω resistor should do. Actually we should be OK with a resistor or 180 Ω to 550 Ω.



We will be using a solderless breadboard to hook up the components. Pin 6 (Ground) should hook to one end of the resistor. Pin 12 (GPIO 18) should hook to the Anode of the LED (the longer lead).

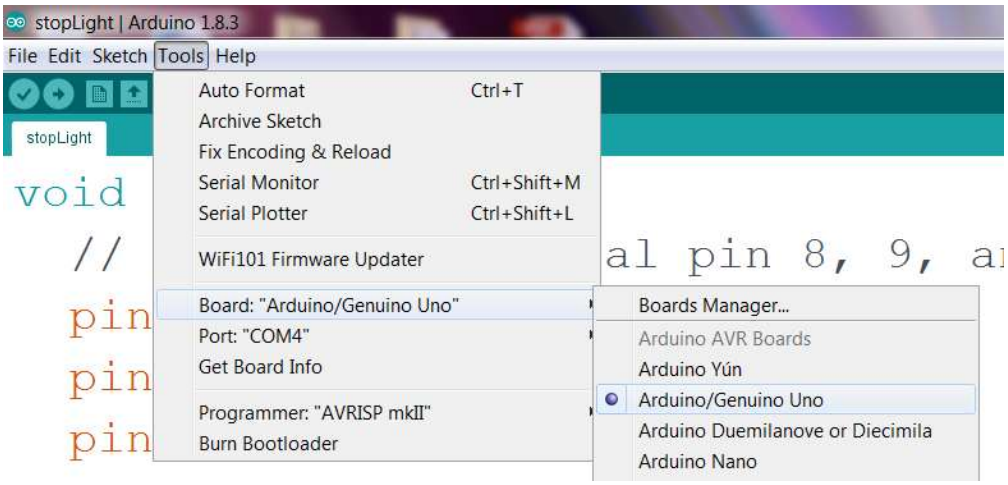
Hardware Assembly Procedure

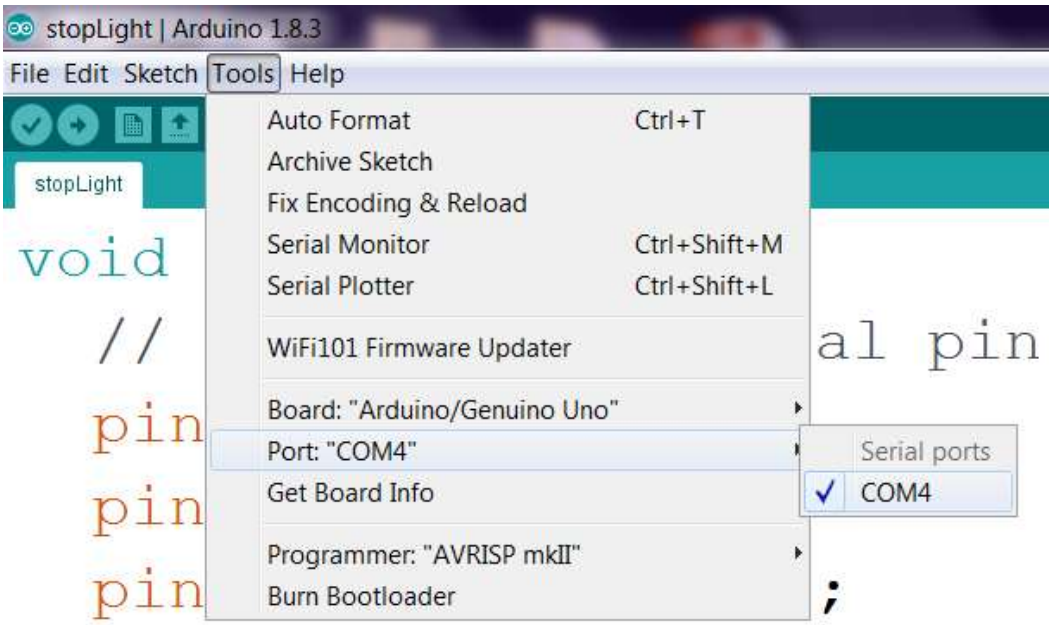
1. Add 330 Ω resistor to the breadboard
2. Add Red LED to breadboard so Cathode of LED (shorter lead) is connected to resistor
3. Add Yellow LED to breadboard so Cathode of LED (shorter lead) is connected to resistor
4. Add Green LED to breadboard so Cathode of LED (shorter lead) is connected to resistor
5. Connect GRD on the Arduino header to unconnected lead of resistor
6. Connect Pin 8 on Arduino header to Anode of Red LED (longer lead)
7. Connect Pin 9 on Arduino header to Anode of Yellow LED (longer lead)
8. Connect Pin 10 on Arduino header to Anode of Green LED (longer lead)
9. Hook up your Arduino to your computer.

Note: If you have the LED hooked up backwards it will not light and you can correct this by reversing its connections.

Now we are ready to program the IO pins. In order to do this we will write a short Arduino script (called a sketch). Start Arduino as follows:

1. Start your Arduino development environment.
2. On your Tools menu, check that the Board is set to Uno and the Port is set to the value you noted in step 5.





First Experiment

Type in and run the following script to verify our LEDs are hooked up correctly:

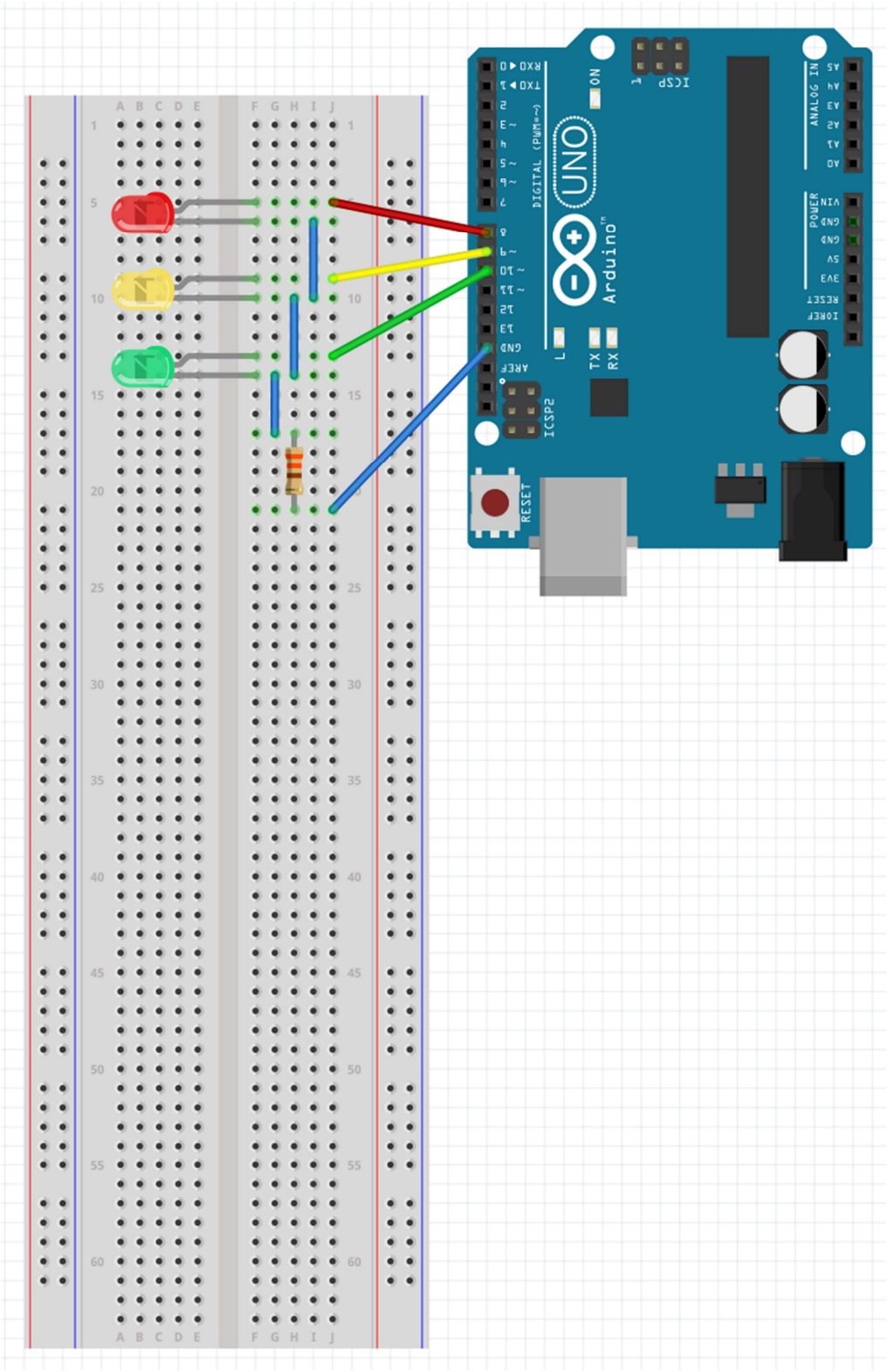
```
//Turn LEDs on

// the setup function runs once when you press reset or power the board
void setup() {
      // initialize digital pin 8, 9, and 10 as an output.
          pinMode(8, OUTPUT);
          pinMode(9, OUTPUT);
          pinMode(10, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
          digitalWrite(8, HIGH);   // turn the LED on (HIGH is the voltage level)
          digitalWrite(9, HIGH);    // turn the LED on (HIGH is the voltage level)
          digitalWrite(10, HIGH);    // turn the LED on (HIGH is the voltage level)
}
```

Compile your sketch and upload it to the Arduino. If everything is hooked up correctly, you should have all 3 LEDs on. If not, check your hookup. Make sure you have the correct polarity.

Now let's make our stoplight simulation:

Type in the following script:

```
// Stop light simulation

// the setup function runs once when you press reset or power the board
void setup() {
	// initialize digital pin 8, 9, and 10 as an output.
	pinMode(8, OUTPUT);
	pinMode(9, OUTPUT);
	pinMode(10, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
	//Green
	digitalWrite(10, HIGH);   // turn the LED on (HIGH is the voltage level)
	delay(6000);
	digitalWrite(10, LOW);   // turn the LED off (LOW is the voltage level)


	//Yellow
	digitalWrite(9, HIGH);   // turn the LED on (HIGH is the voltage level)
	delay(2000);
	digitalWrite(9, LOW);    // turn the LED off (LOW is the voltage level)

	//Red
	digitalWrite(8, HIGH);   // turn the LED on (HIGH is the voltage level)
	delay(6000);
	digitalWrite(8, LOW);    // turn the LED off (LOW is the voltage level)

}
```


UNO
Arduino
DIGITAL (PWM~)
ANALOG IN
POWER
ICSP
ICSP2
RESET
ON
L
TX
RX