

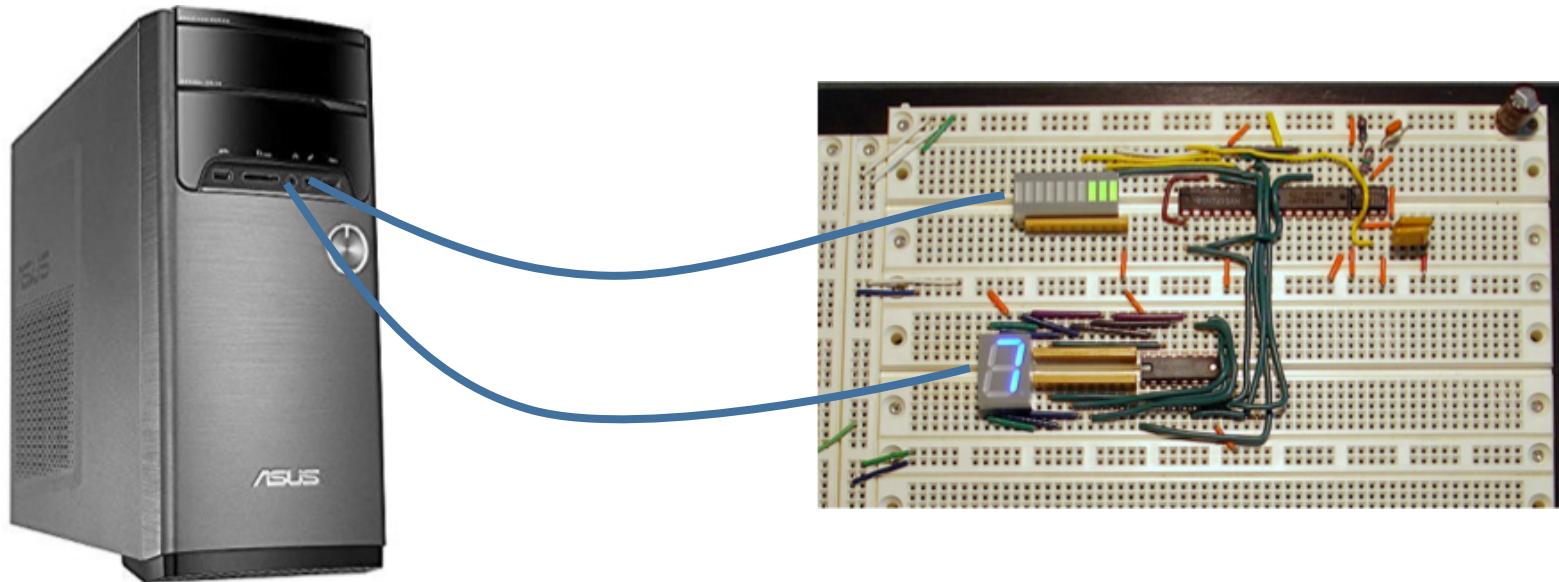
Arduino Nugget

Span Spanbauer (many slides by Sarah Bricault)

Motivation

If computers are digital logic machines, why can't we easily connect our own electronics to them?

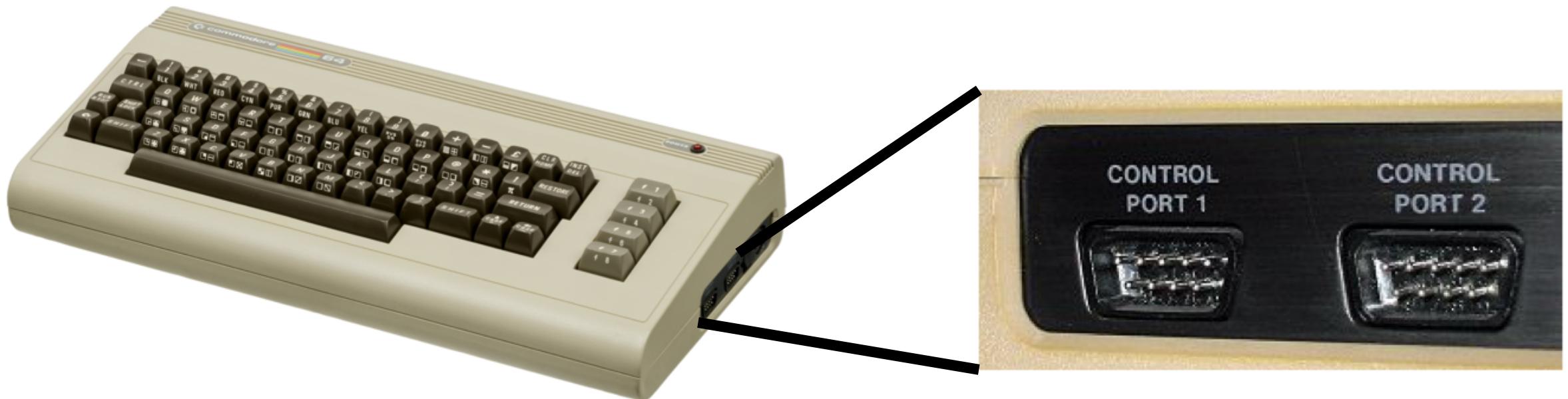
When I was growing up, I looked for a solution but couldn't find one. Eventually I ended up using the computer's microphone and headphone ports as analog input and output.



History

Actually, it's only recently that this has been a challenge.

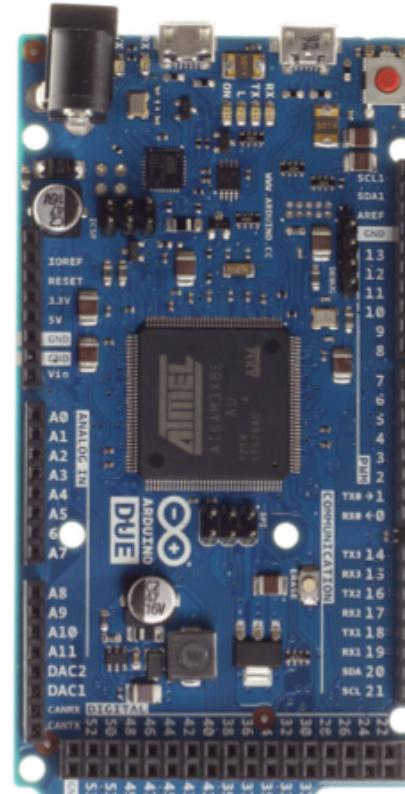
For example, consider the best selling computer model of all time, the Commodore 64. Introduced in 1982, it had four reasonably accessible analog inputs, and a variety of digital inputs and outputs.



History

This accessibility was lost as computers evolved towards complex protocols, such as USB.

Luckily there are now better ways. By 2007 Arduino boards were widely available, and now there are a variety of similar systems available for hobby electronics.



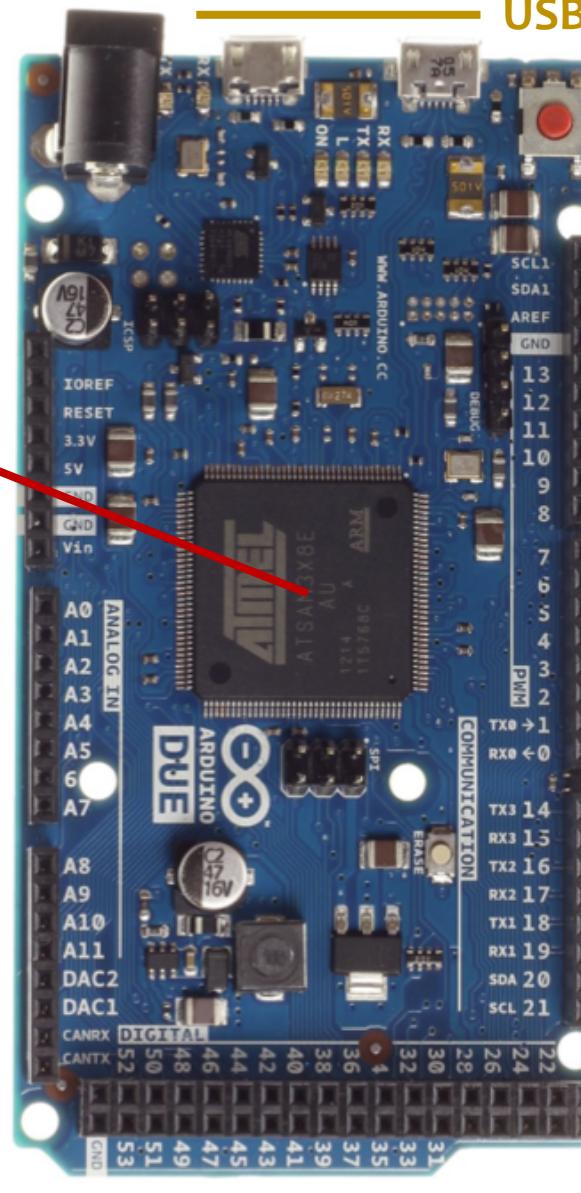
Specs

Arduino Due

84MHz microcontroller

12-bit ADCs

12-bit DACs



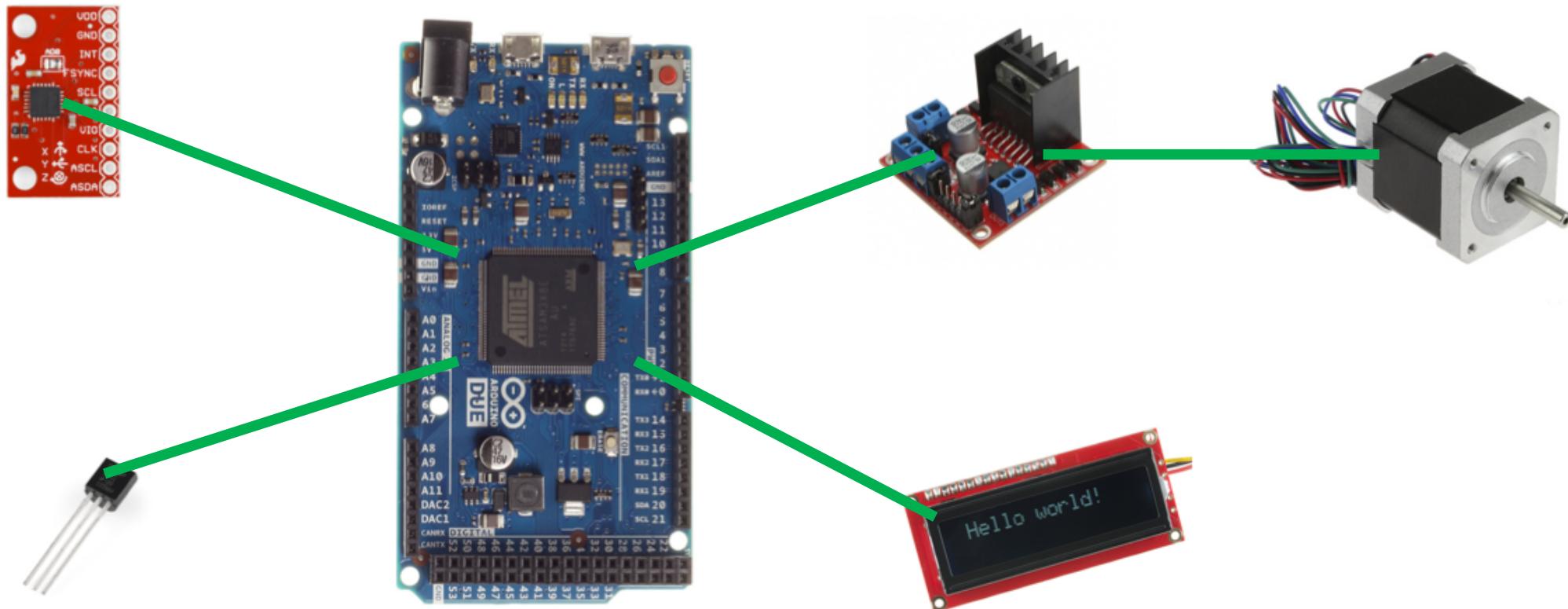
USB communication with computer (virtual serial port)

Digital I/O

Digital communication

Strategy

Use the Arduino as a hub to control sensors and actuators, record data to computer



Advantages and Disadvantages

Pro

Very easy to get up and running

Sufficient for many tasks

Good debugging tools

Deterministic execution (unlike an OS)

Con

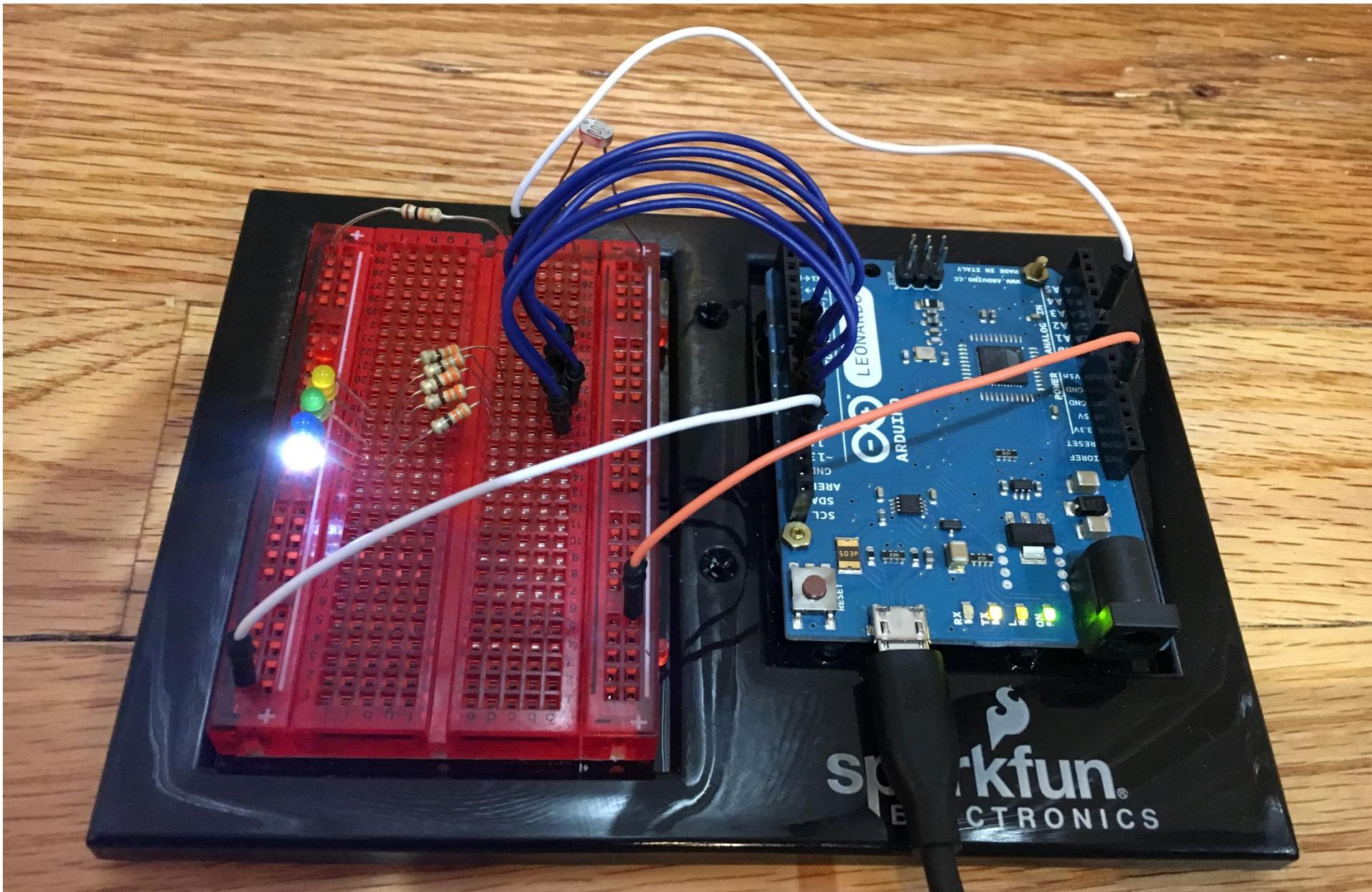
Somewhat slow if code is not optimized

Moderate cost

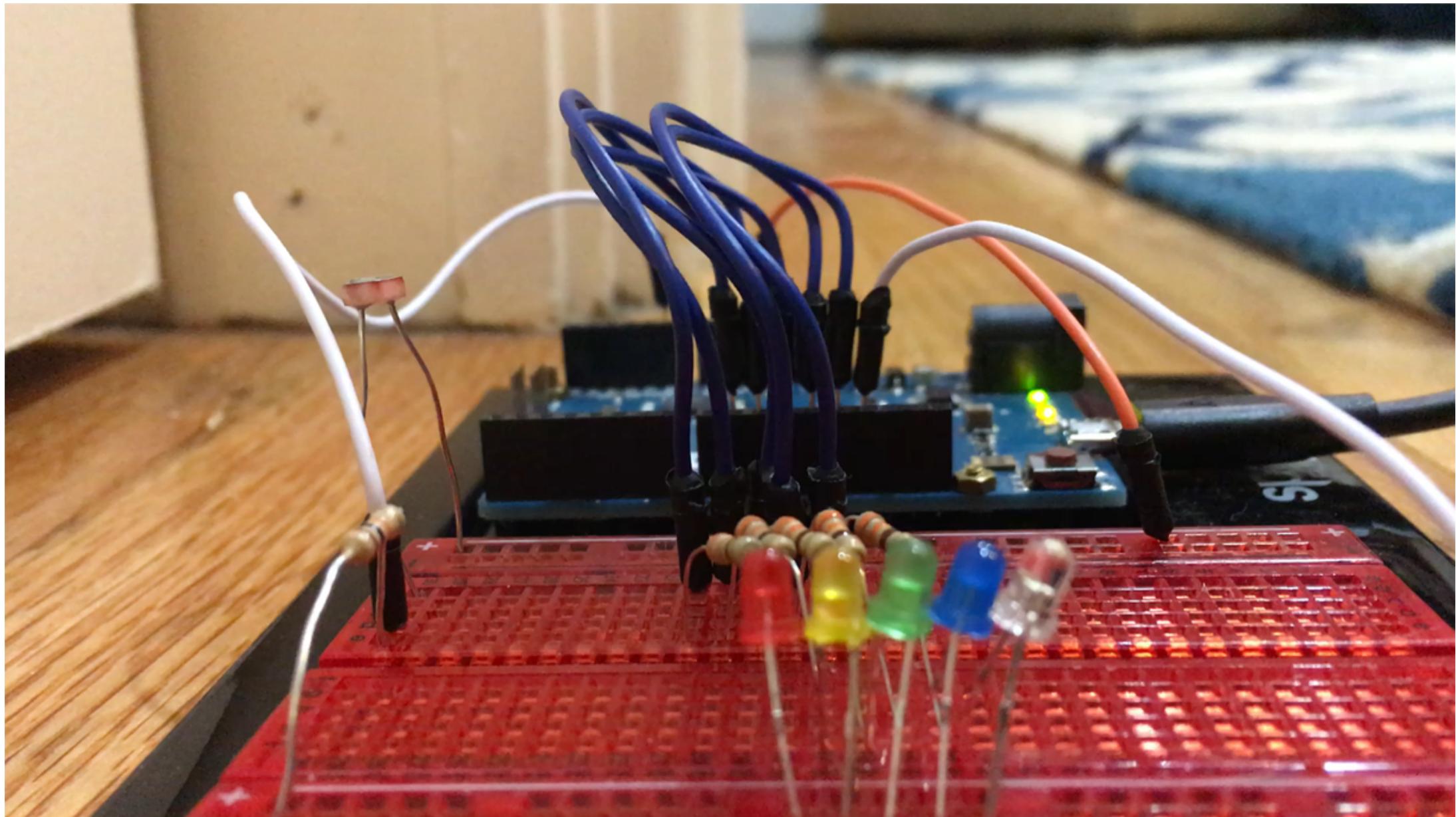
Connote “hobbyist quality”

Bottom line: Fantastic for prototyping when it is sufficiently fast.

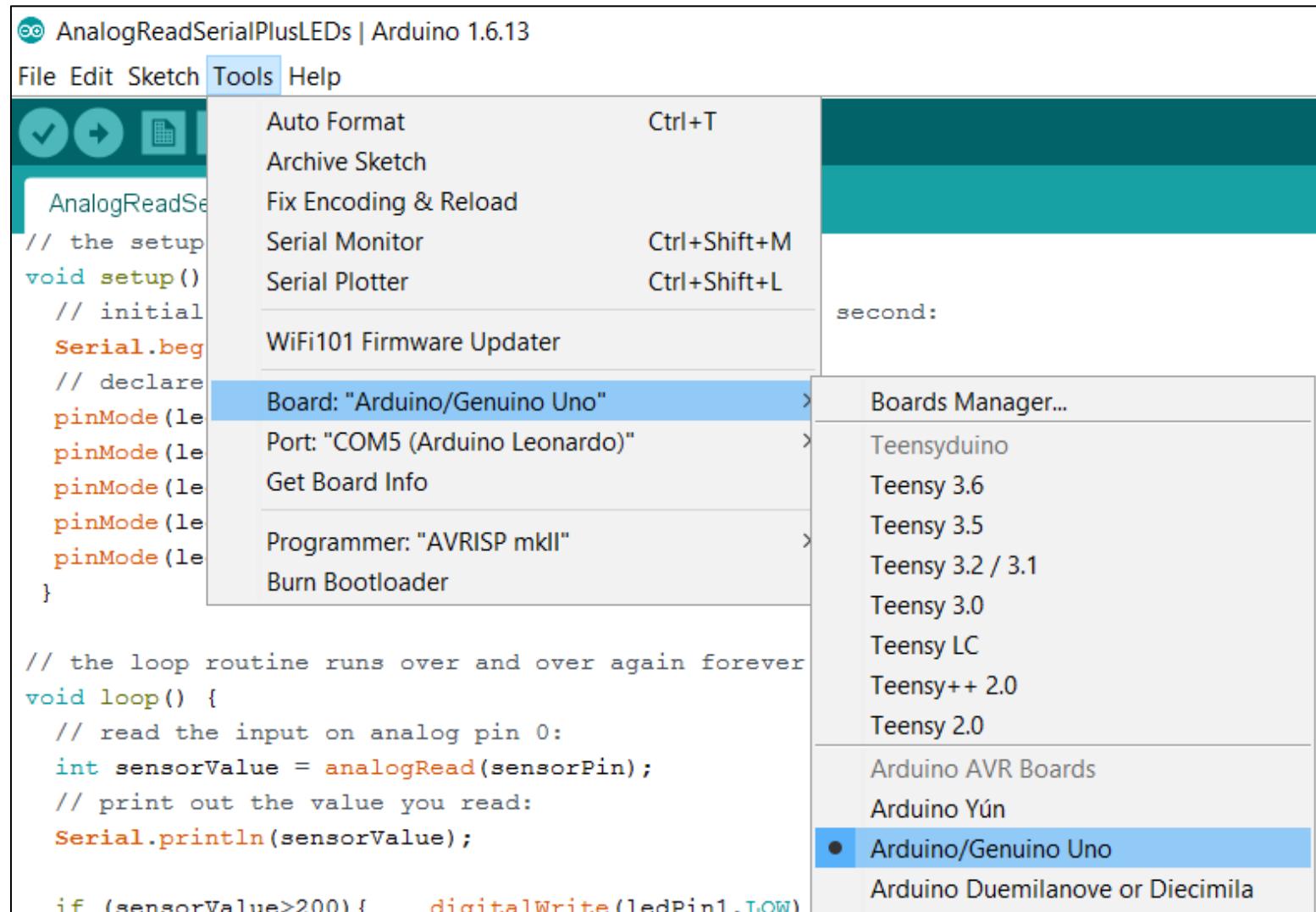
Goal



Goal



Setup: Select your board



The screenshot shows the Arduino IDE interface. The title bar reads "AnalogReadSerialPlusLEDs | Arduino 1.6.13". The menu bar includes File, Edit, Sketch, Tools (which is selected and highlighted in blue), and Help. The main window displays a portion of an Arduino sketch. The code includes setup and loop functions for reading analog input and printing to the serial monitor. A conditional statement at the bottom controls a digital LED output based on sensor value. The Tools menu is open, showing options for Auto Format, Archive Sketch, Fix Encoding & Reload, Serial Monitor, Serial Plotter, WiFi101 Firmware Updater, and Board selection. The Board option is expanded, showing a list of available boards: Arduino/Genuino Uno, Teensyduino, Teensy 3.6, Teensy 3.5, Teensy 3.2 / 3.1, Teensy 3.0, Teensy LC, Teensy++ 2.0, Teensy 2.0, Arduino AVR Boards, Arduino Yún, Arduino/Genuino Uno (which is selected and highlighted in blue), and Arduino Duemilanove or Diecimila.

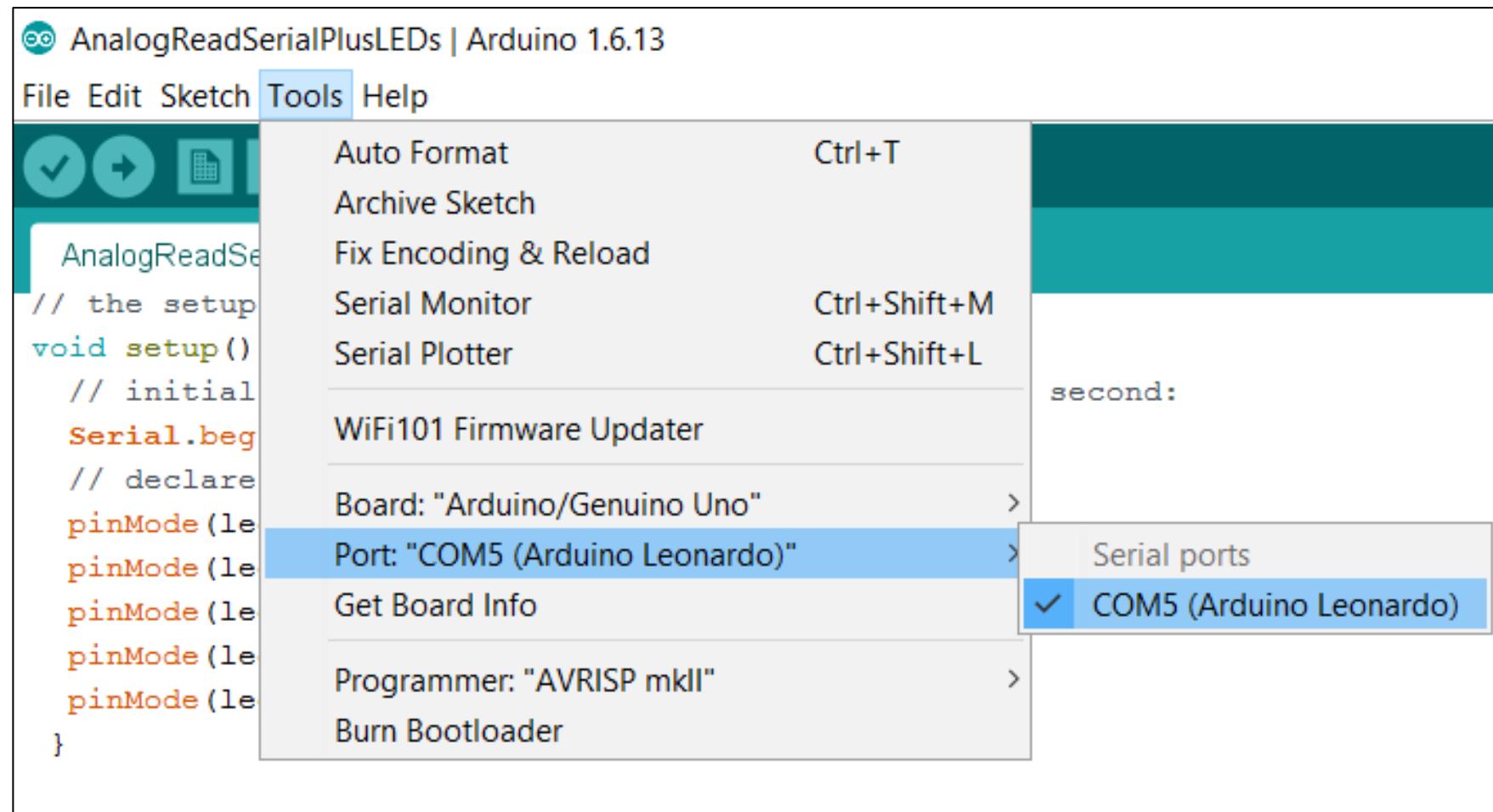
```
// the setup
void setup() {
    // initial
    Serial.begin();
    // declare
    pinMode(ledPin1, OUTPUT);
    pinMode(ledPin2, OUTPUT);
    pinMode(ledPin3, OUTPUT);
    pinMode(ledPin4, OUTPUT);
    pinMode(ledPin5, OUTPUT);
}

// the loop routine runs over and over again forever
void loop() {
    // read the input on analog pin 0:
    int sensorValue = analogRead(sensorPin);
    // print out the value you read:
    Serial.println(sensorValue);

    if (sensorValue>200) {      digitalWrite(ledPin1,LOW);
                                digitalWrite(ledPin2,LOW);
                                digitalWrite(ledPin3,LOW);
                                digitalWrite(ledPin4,LOW);
                                digitalWrite(ledPin5,LOW);
    }
}
```

Connect your Arduino to the USB cord and attach to your computer.

Setup: Select your COM port



Basics of an Arduino Program

This arrow
uploads your
program

```
BareMinimum | Arduino 1.6.13
File Edit Sketch Tools Help
BareMinimum §

void setup() {
    // put your setup code here, to run once:
}

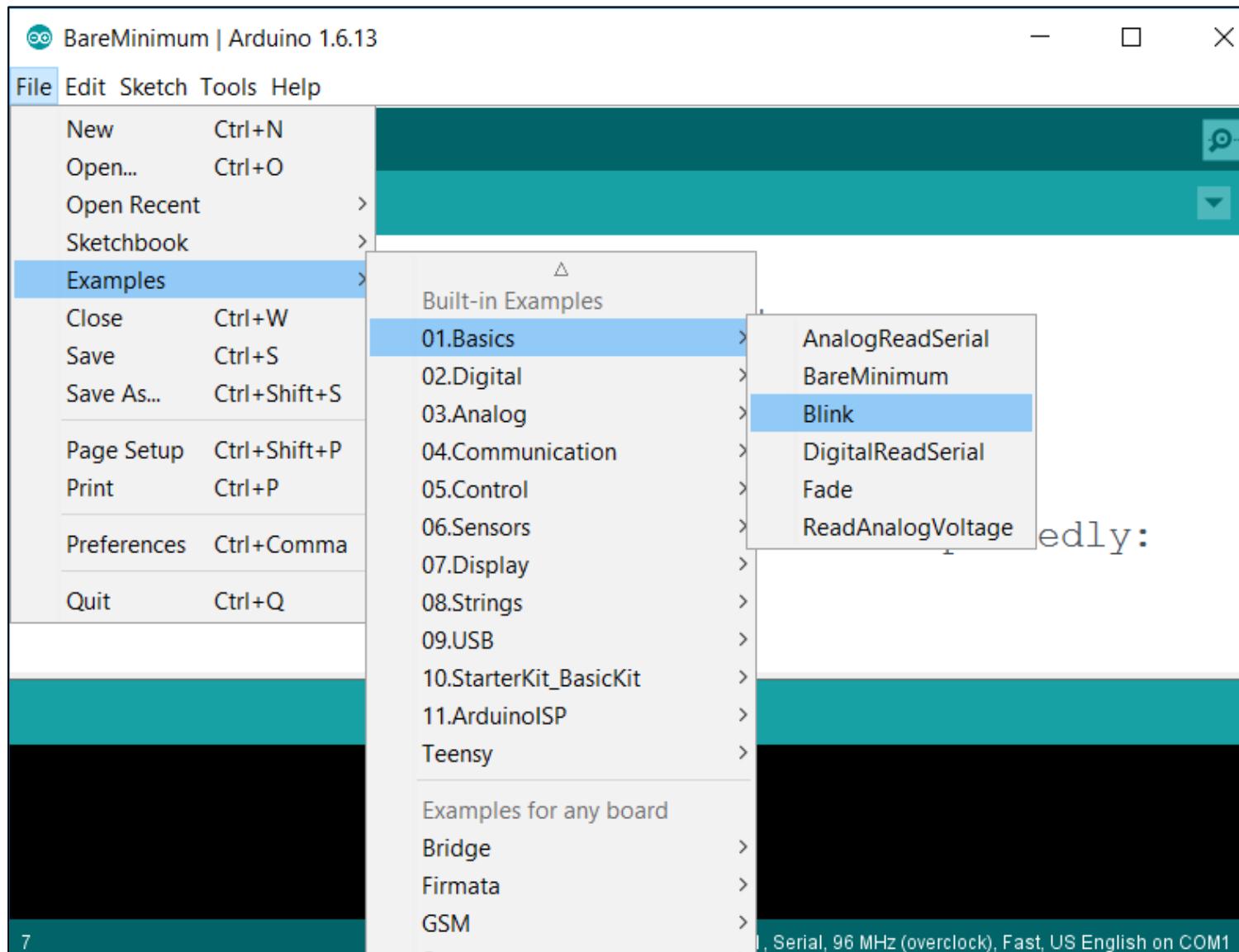
void loop() {
    // put your main code here, to run repeatedly:
}
```

Teensy 3.2 / 3.1, Serial, 96 MHz (overclock), Fast, US English on COM1

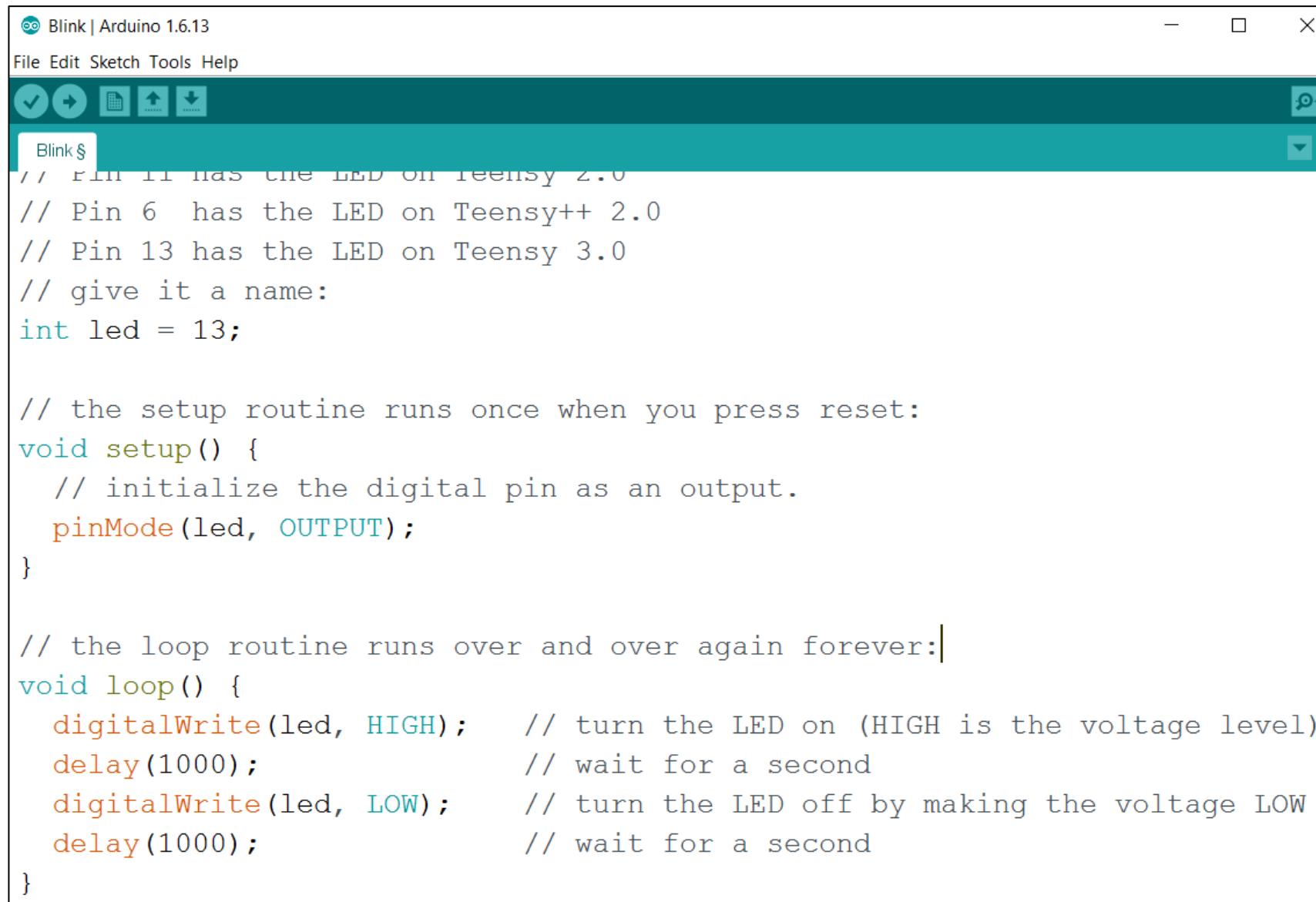
Upload errors
will appear
down here

// indicates a comment,
or notation that will not
affect how the code runs

Simple Program: Blink



Simple Program: Blink



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.6.13". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for Save, Run, Upload, and Download. The main window displays the "Blink" sketch code. The code is as follows:

```
// Pin 11 has the LED on Teensy 2.0
// Pin 6 has the LED on Teensy++ 2.0
// Pin 13 has the LED on Teensy 3.0
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
    // initialize the digital pin as an output.
    pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
    digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);                // wait for a second
    digitalWrite(led, LOW);      // turn the LED off by making the voltage LOW
    delay(1000);                // wait for a second
}
```

Variables

```
int inputVariable1;  
int inputVariable2 = 0;      // both are correct
```

Some basic data types

- Integer (int)
- String
- Float

If / else statement

```
if (pinFiveInput < 500)
{
    // action A
}
else
{
    // action B
}
```

Useful syntax			
&&	and	!=	not equal to
!	not	==	test equality
	or	>=	greater than or equal to
		<=	less than or equal to

For loop

```
for (initialization; condition; increment) {  
//statement(s);  
}
```

```
for (int i=0; i <= 255; i++){  
//statement(s);  
}
```

Useful syntax

continue; go to next iteration of loop

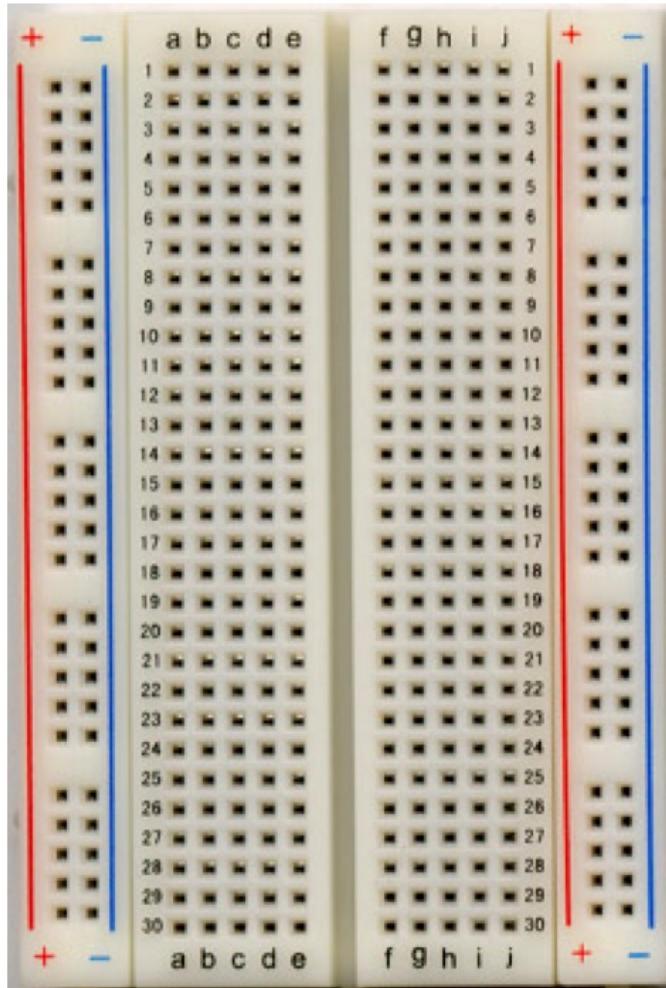
While loop

```
while(expression){  
    // statement(s)  
}
```

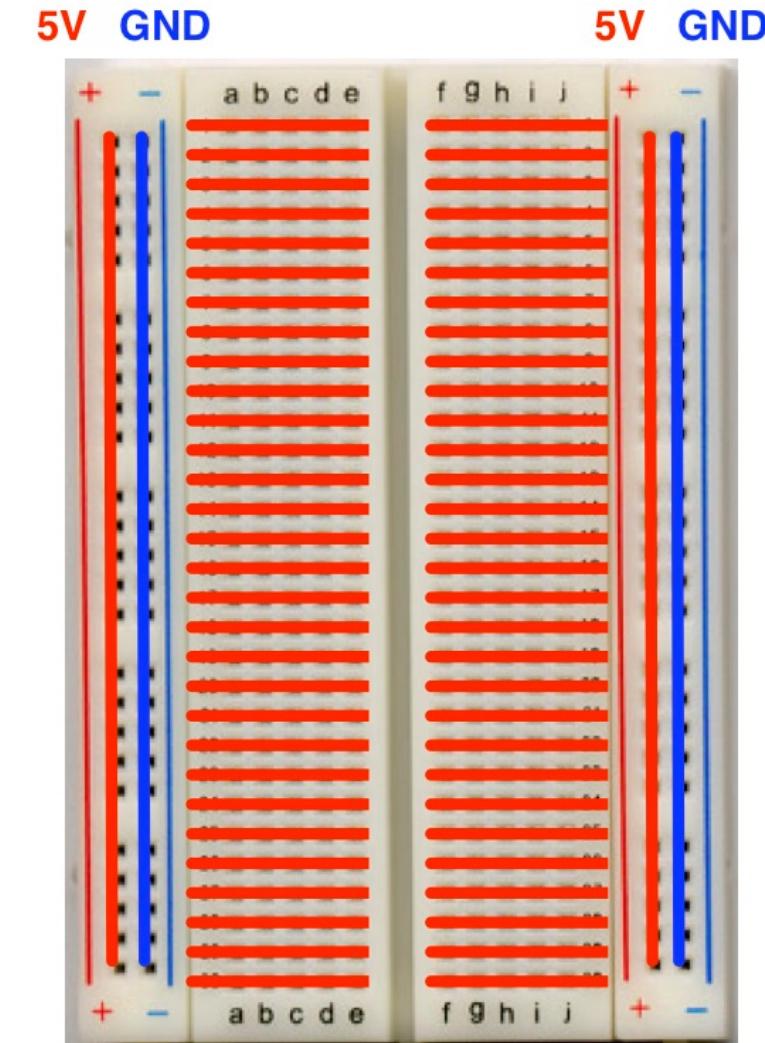
```
var = 0;  
while(var < 200){  
    // do something repetitive 200 times  
    var++;  
}
```

Breadboard

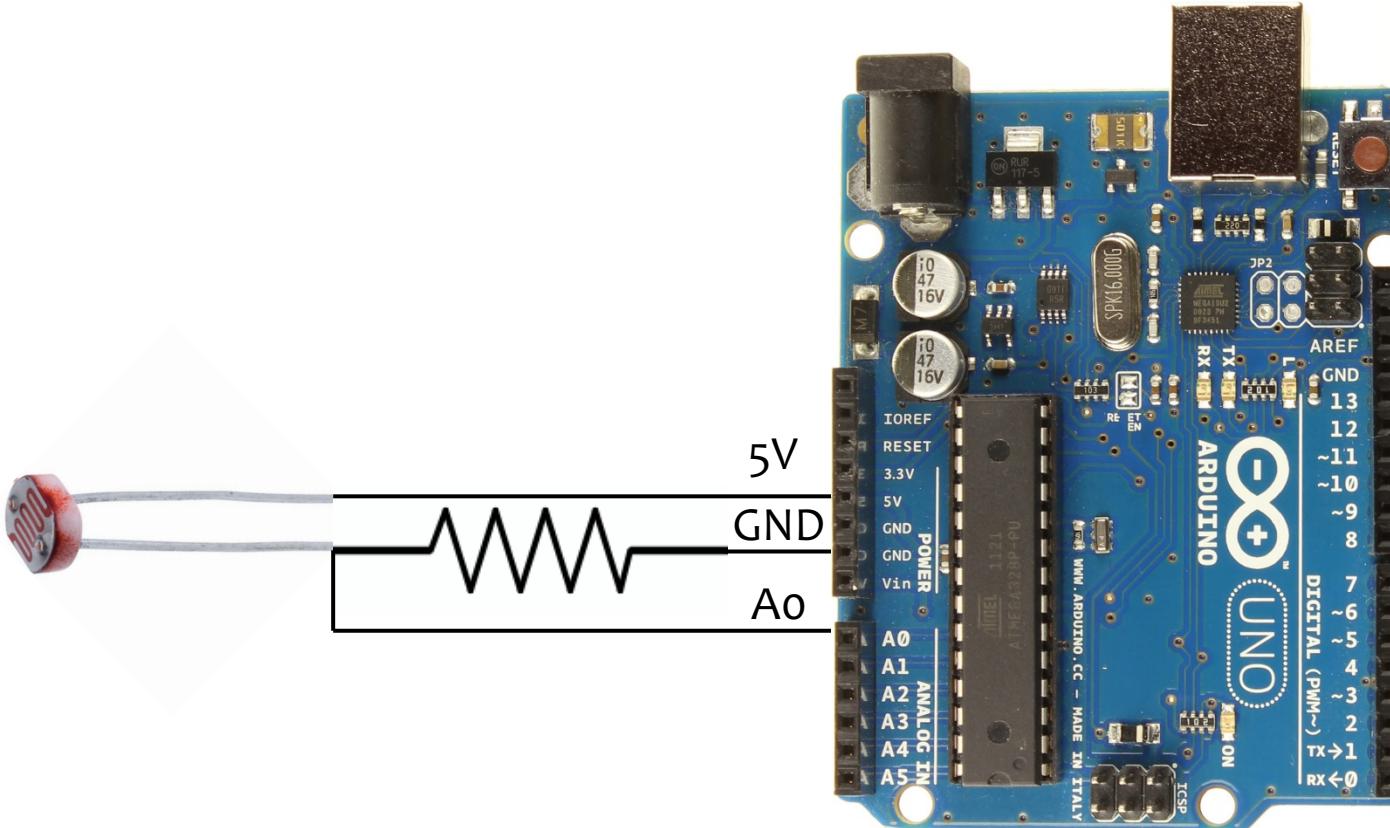
Breadboard (photo)



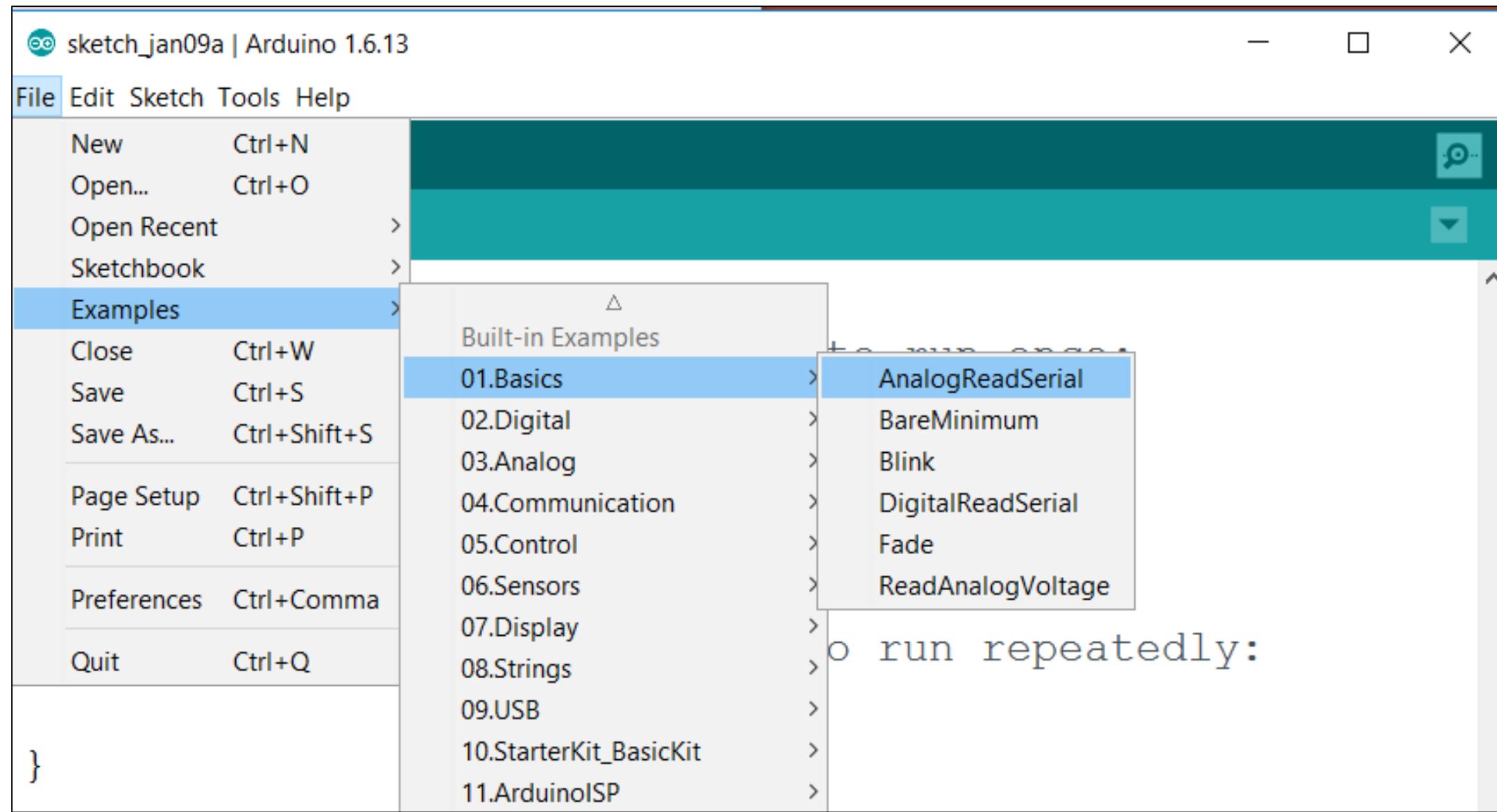
Breadboard (schematic)



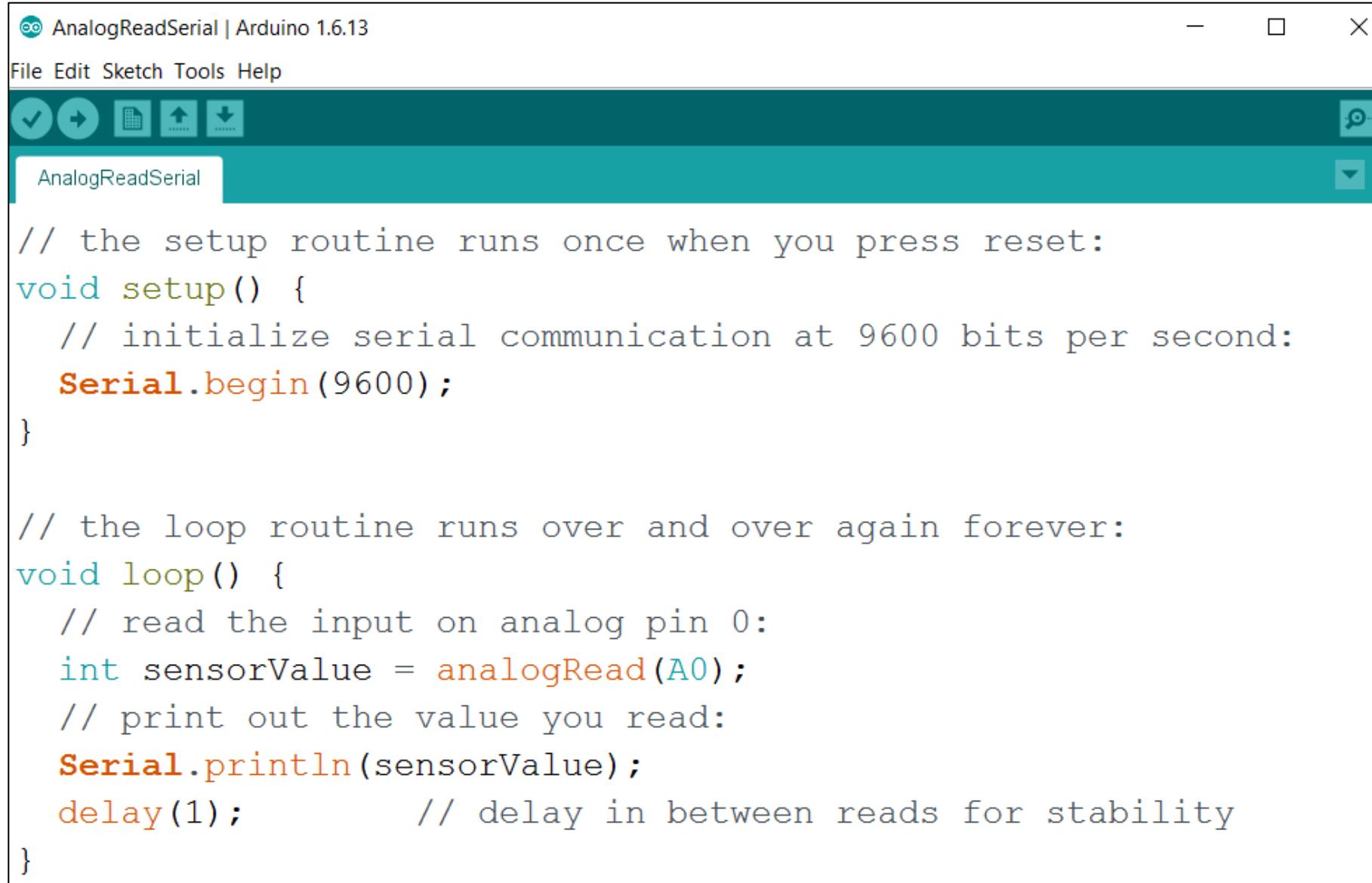
Your first circuit: the diagram



Your first circuit: the program



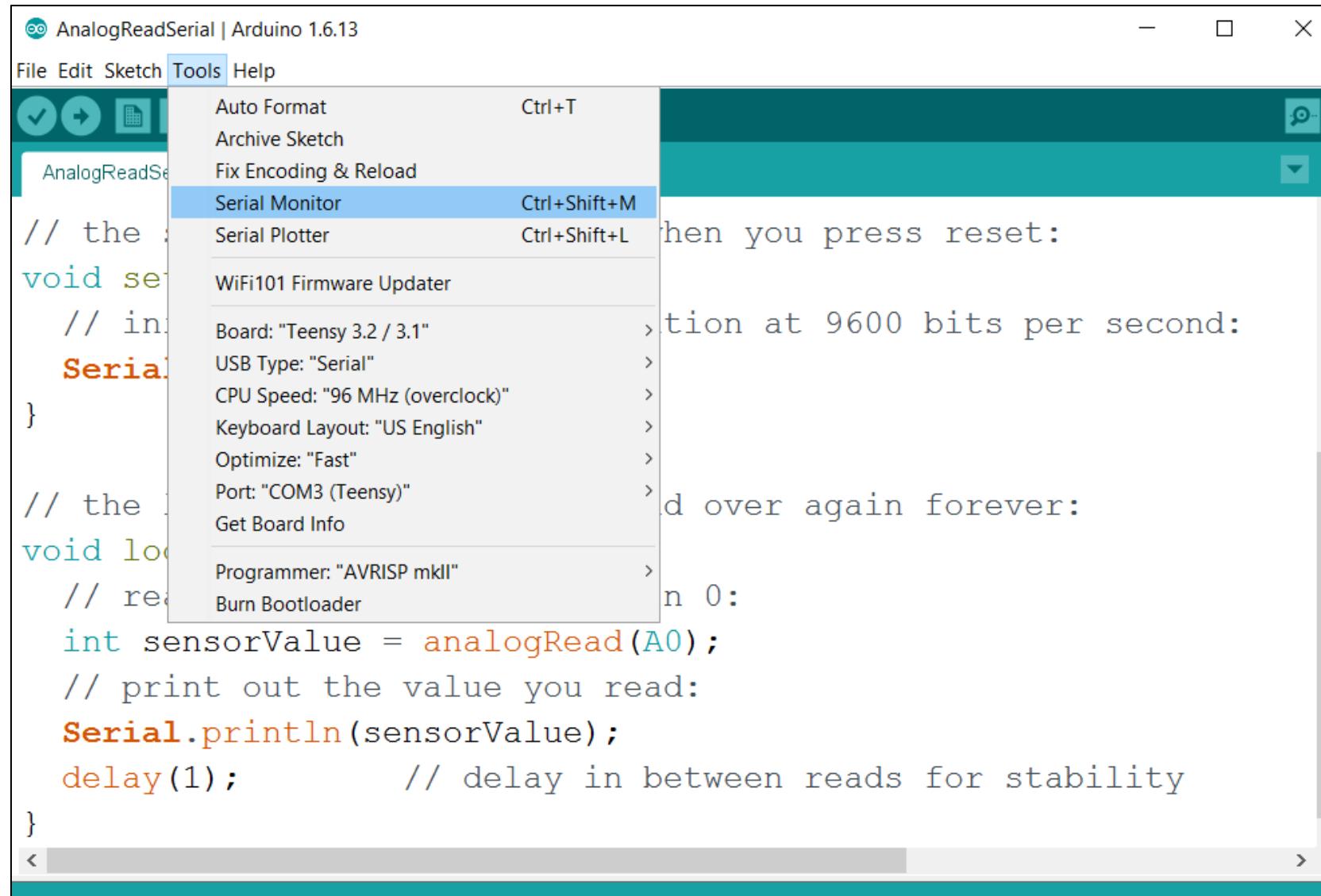
Your first circuit: the program



The screenshot shows the Arduino IDE interface with the title bar "AnalogReadSerial | Arduino 1.6.13". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for save, upload, and other functions. The main code editor window displays the following C++ code:

```
// 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(1);          // delay in between reads for stability  
}
```

Your first circuit: reading input



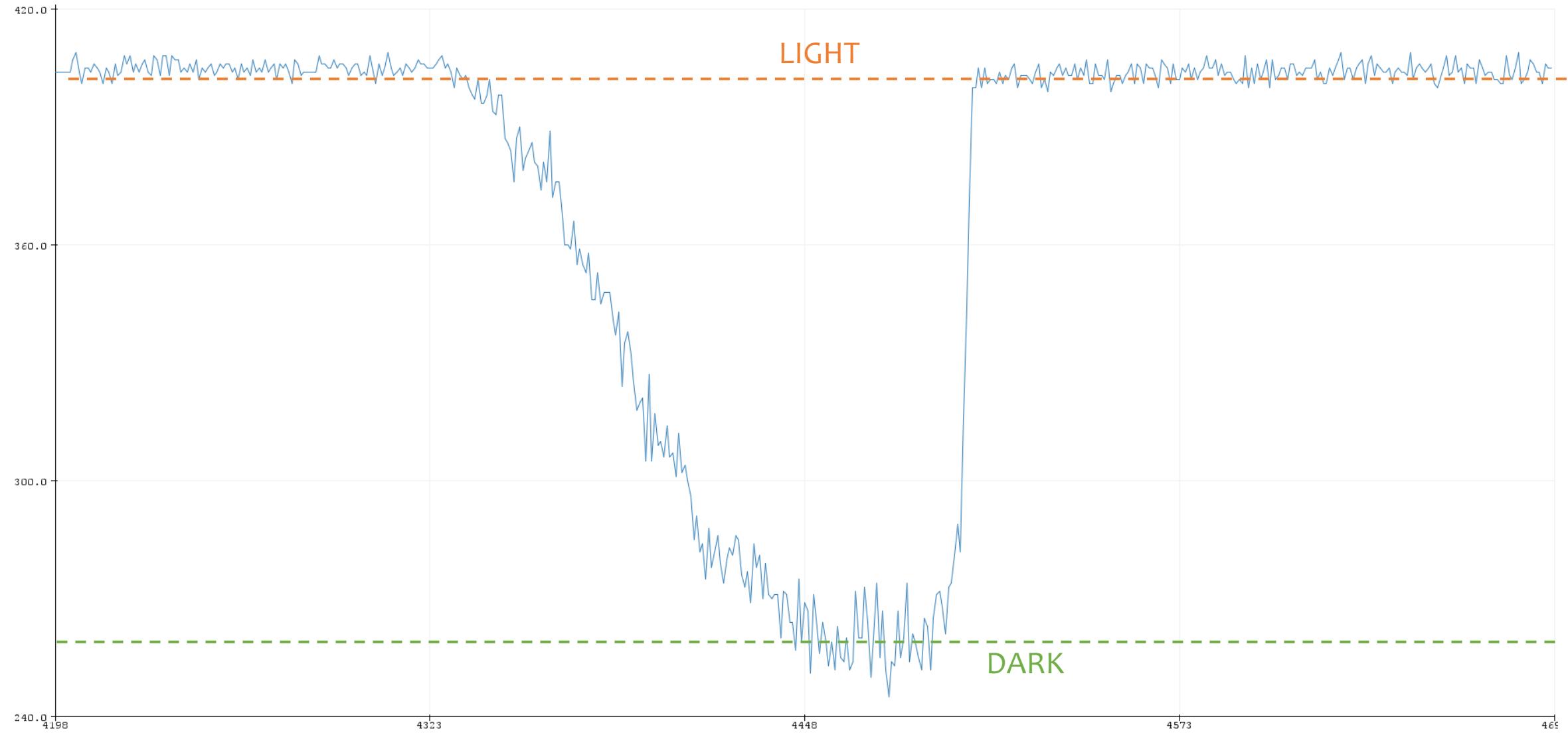
The screenshot shows the Arduino IDE interface. The title bar reads "AnalogReadSerial | Arduino 1.6.13". The menu bar includes "File", "Edit", "Sketch", "Tools" (which is currently selected and highlighted in blue), and "Help". The main window displays a portion of an Arduino sketch named "AnalogReadSerial". The code includes comments explaining the setup of pins A0 and A1, the creation of a Serial object, and a loop that reads analog values from pin A0, prints them to the Serial Monitor, and adds a one-second delay between reads. The Serial Monitor window is visible in the background, showing the output of the code.

```
// the pins we're using
void setup() {
    // initialize serial communication at 9600 bits per second:
    Serial.begin(9600);
}

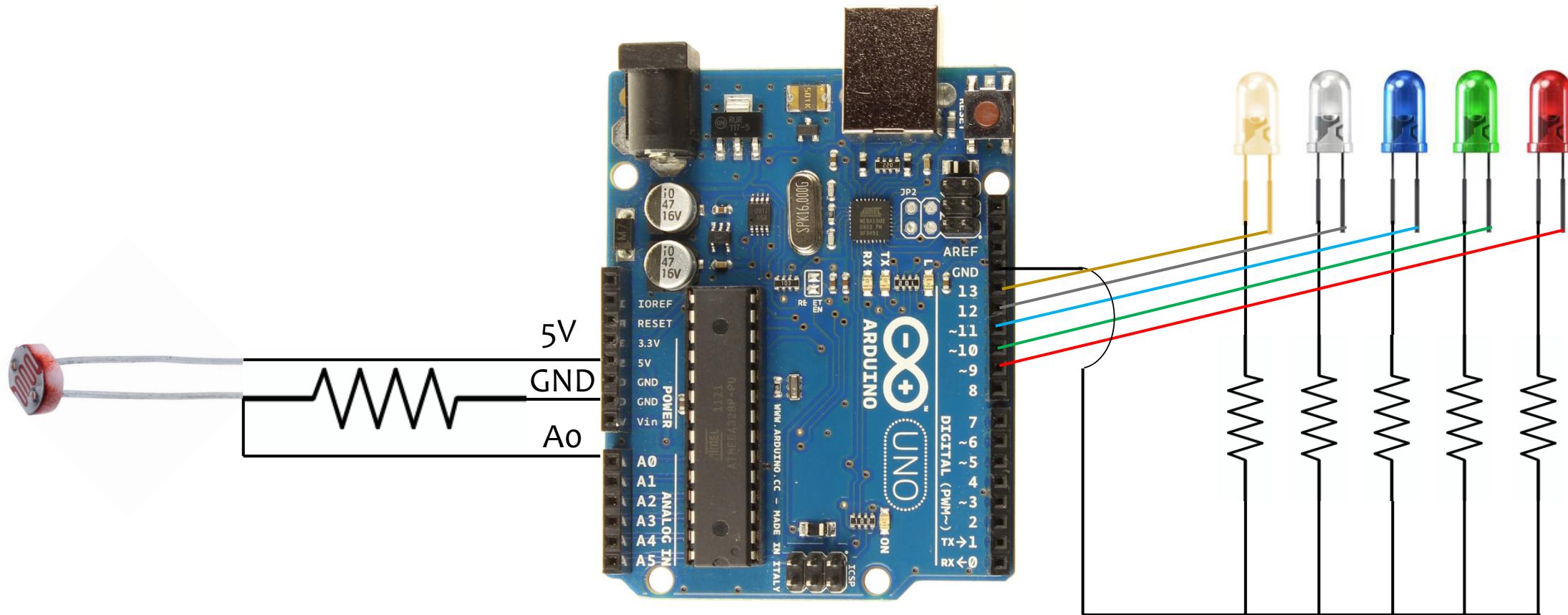
// the loop
void loop() {
    // read the analog signal:
    int sensorValue = analogRead(A0);

    // print out the value you read:
    Serial.println(sensorValue);
    delay(1);          // delay in between reads for stability
}
```

Your first circuit: serial plotter



Your second circuit: the diagram



Your second circuit: the program

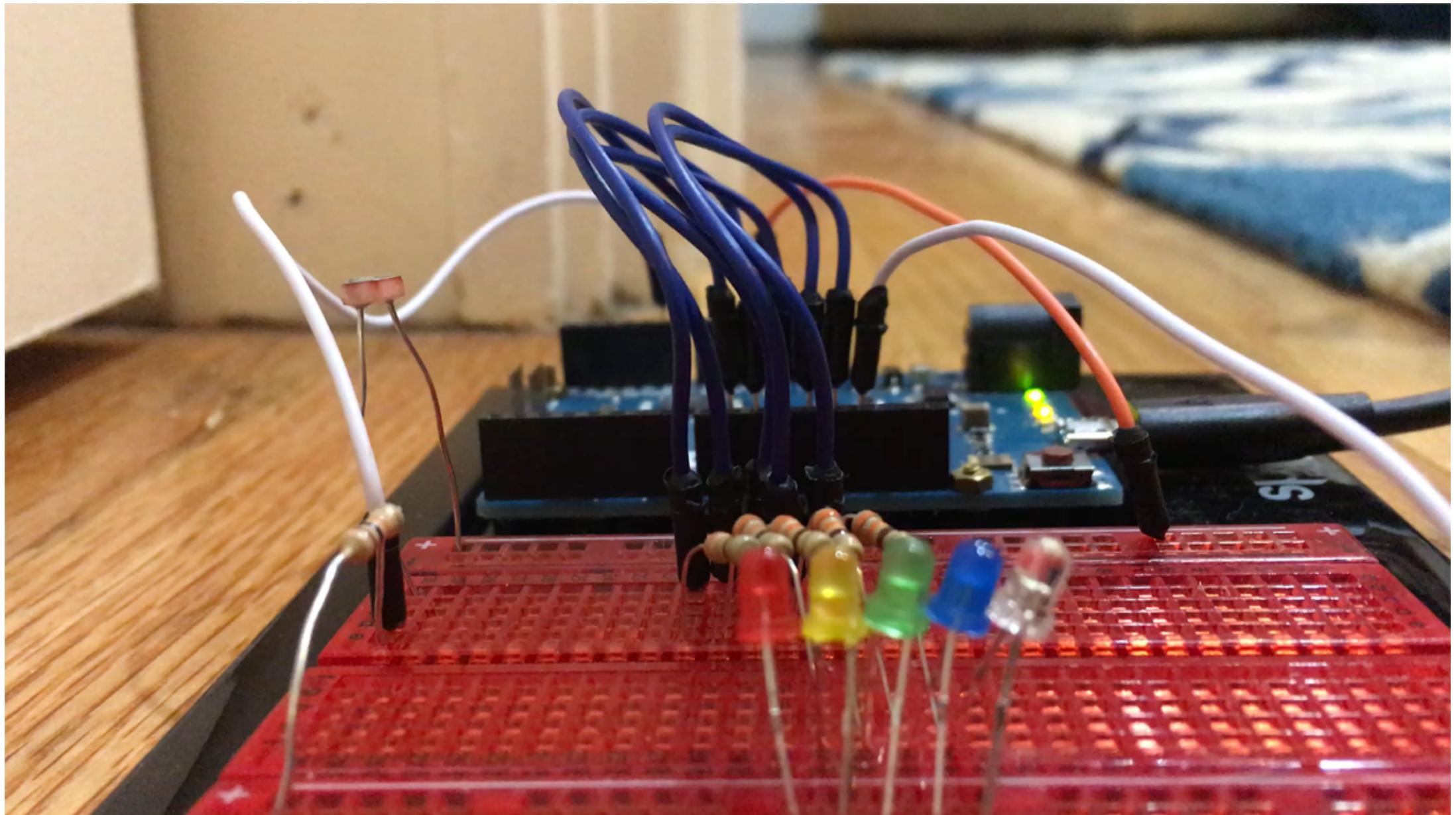
```
int sensorPin = A0;
int ledPin1=9;
int ledPin2=10;
int ledPin3=11;
int ledPin4=12;
int ledPin5=13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // declare the ledPins as OUTPUT:
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin4, OUTPUT);
  pinMode(ledPin5, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(sensorPin);
  // print out the value you read:
  Serial.println(sensorValue);

  if (sensorValue>200){ digitalWrite(ledPin1,LOW); }
  else{ digitalWrite(ledPin1,HIGH); }
  if (sensorValue>250){ digitalWrite(ledPin2,LOW); }
  else{ digitalWrite(ledPin2,HIGH); }
  if (sensorValue>300){ digitalWrite(ledPin3,LOW); }
  else{ digitalWrite(ledPin3,HIGH); }
  if (sensorValue>350){ digitalWrite(ledPin4,LOW); }
  else{ digitalWrite(ledPin4,HIGH); }
  if (sensorValue>400){ digitalWrite(ledPin5,LOW); }
  else{ digitalWrite(ledPin5,HIGH); }
  delay(1);           // delay in between reads for stability
}
```

Your second circuit





Thanks !

