Using Arduino With Vernier Sensors

Vernier + Arduino

Objective of Module

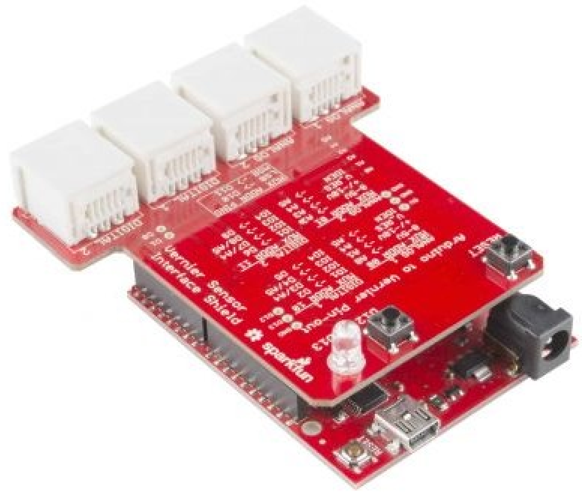
Here we look at how to use analog Vernier sensors with Arduino. (Currently most Vernier sensors appear to be analog). The Vernier.com website has a lot of information about this topic. There are three ways to connect Vernier probes to an Arduino:

1. Buy a Vernier shield from Sparkfun
2. Buy connector break-out boards that are compatible with your Arduino breadboards
3. Buy inexpensive connectors and attach your wiring directly to the connectors.

In this module we illustrate all three techniques, and include a simple project that uses a Vernier light gauge as the basis for an analog light gauge.

1. Sparkfun Shield

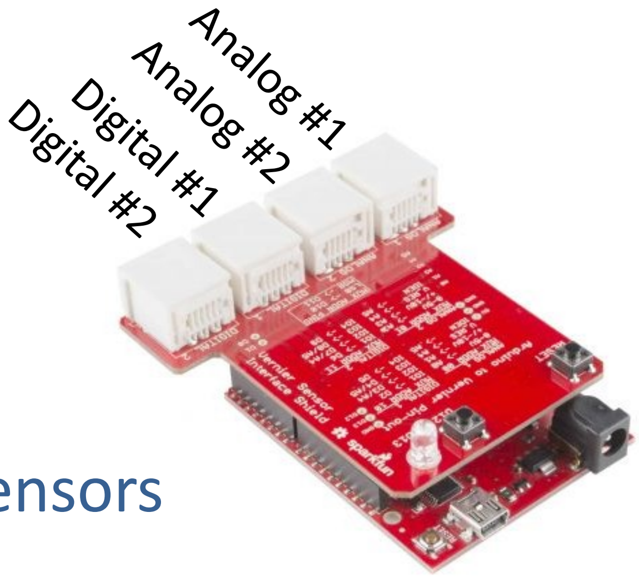
- Two analog ports
- Two digital ports
- Blocks all access to Arduino
- Idea: Plug Vernier sensors into shield. Write Arduino sketch to process sensor values



This figure shows the shield plugged into an Arduino

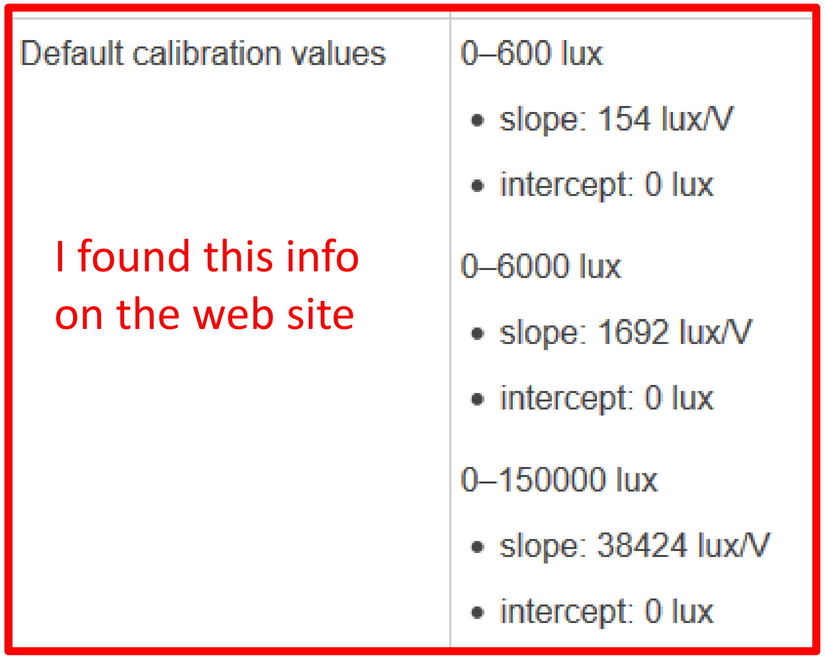
Shield Analog Ports

- Vernier sensors plug into:
  - Analog port 1:
    - Connects to Arduino pin A0
  - Analog port 2:
    - Connects to Arduino pin A2
  - Digital ports 1 and 2 are for digital sensors
    - I didn't look at these due to time
- Special purpose (not used much?)
  - Arduino pins A1 and A3 used for 30V Vernier sensor
  - V-Res is on A4 and IDEN is on A5



Using Light Sensor with Shield

- For test case we look at Vernier light sensor
- Technical data for sensors at: https://www.vernier.com/support/manuals/
- We will need Calibration Values (slope and intercept) for our Arduino sketch:



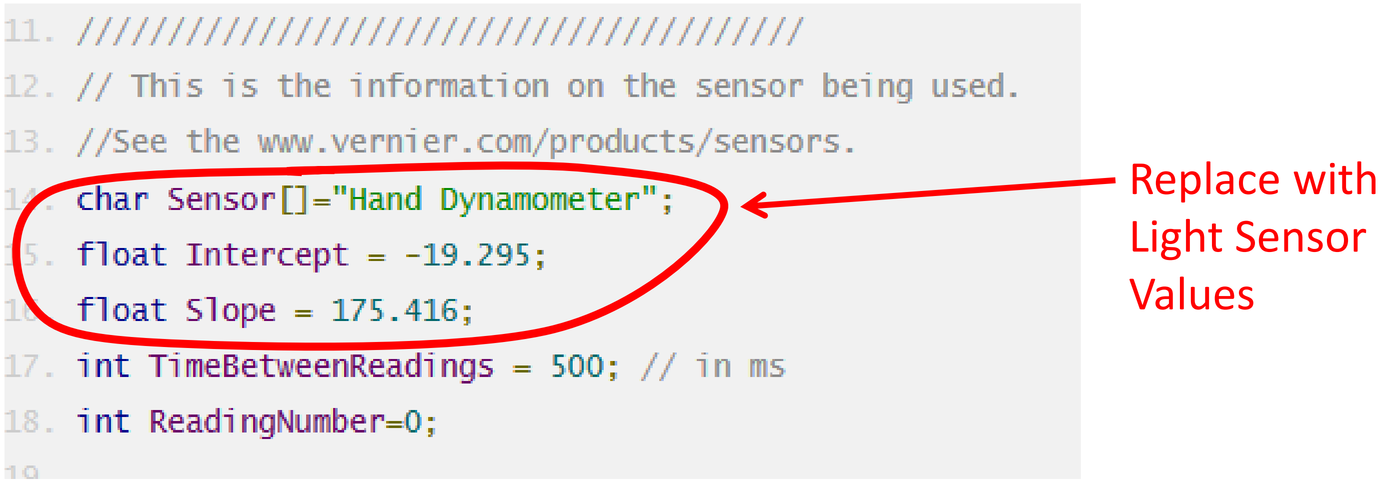
Default calibration values	0–600 lux • slope: 154 lux/V • intercept: 0 lux 0–6000 lux • slope: 1692 lux/V • intercept: 0 lux 0–150000 lux • slope: 38424 lux/V • intercept: 0 lux

I found this info on the web site

Vernier Web Page:

"Reading Data from Sensors with Linear Calibrations"

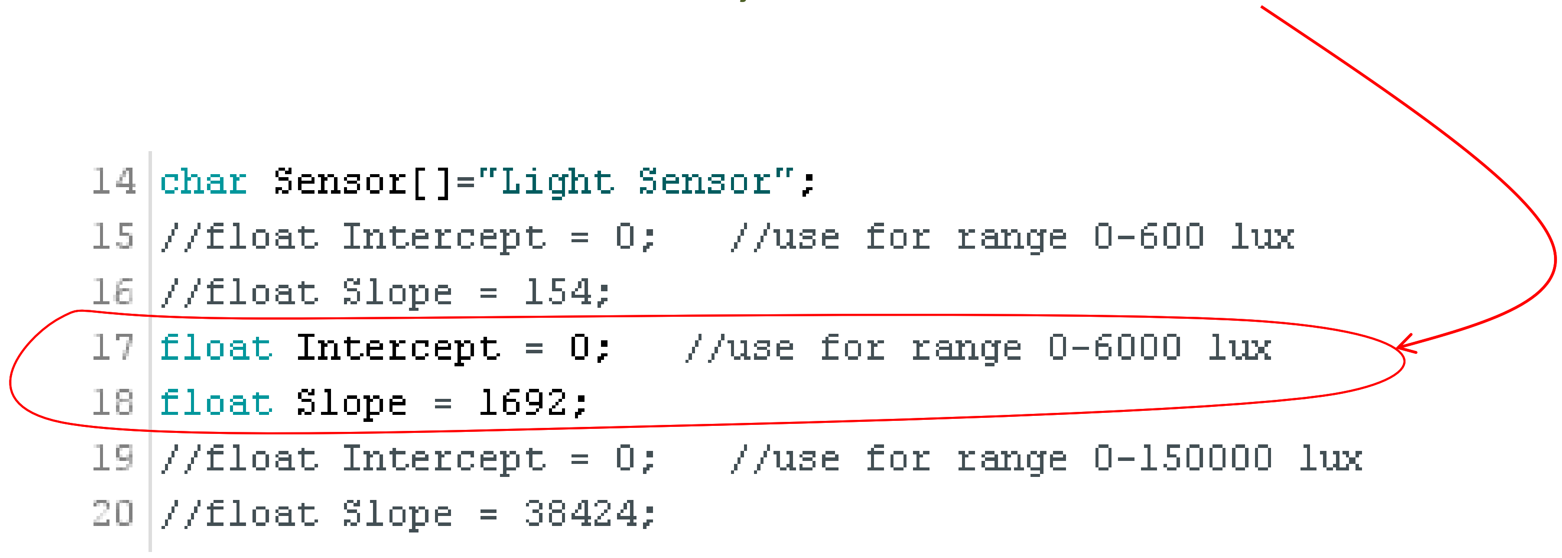
- Vernier code example at: https://www.vernier.com/engineering/arduino/analog-sensors/linear/
- We can use their example code, but it requires minor modifications to use our particular sensor



```
/////////////////////////////////////////
// This is the information on the sensor being used.
//See the www.vernier.com/products/sensors.
char Sensor[]="Hand Dynamometer";
float Intercept = -19.295;
float Slope = 175.416;
int TimeBetweenReadings = 500; // in ms
int ReadingNumber=0;
```

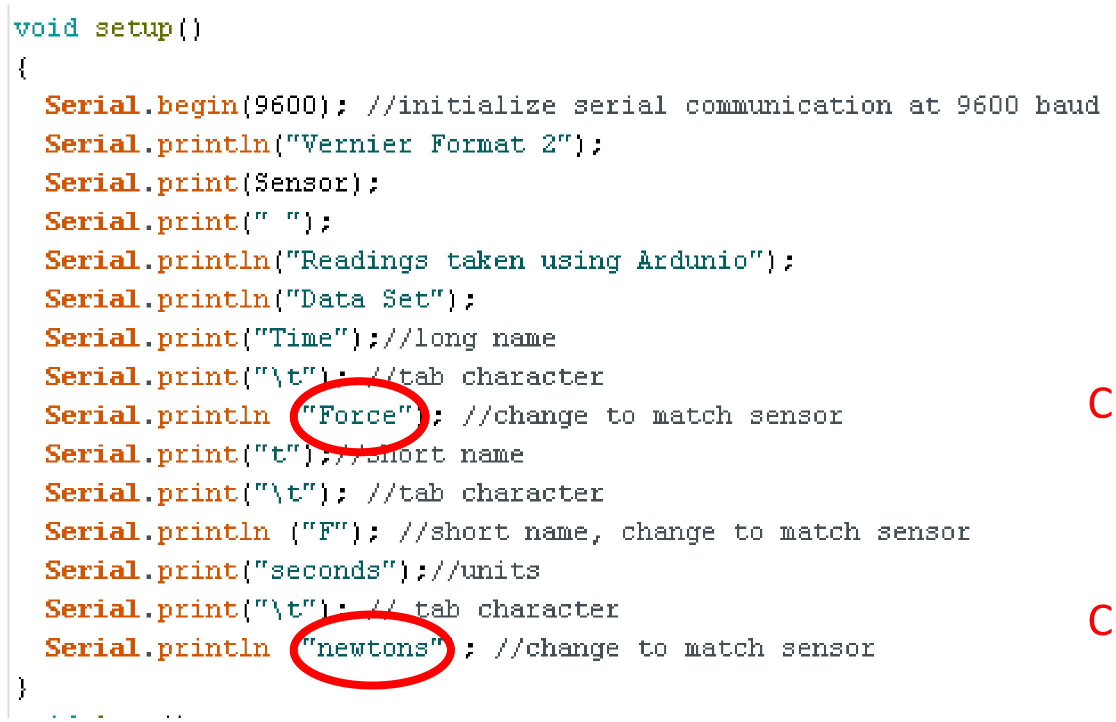
Example continued

- Modified code for Light Sensor
- The light sensor has 3 light ranges selected by a switch. For now, assume “0-6000” is selected



```
char Sensor[]="Light Sensor";
//float Intercept = 0;    //use for range 0-600 lux
//float Slope = 154;
float Intercept = 0;    //use for range 0-6000 lux
float Slope = 1692;
//float Intercept = 0;    //use for range 0-150000 lux
//float Slope = 38424;
```

Modify setup()

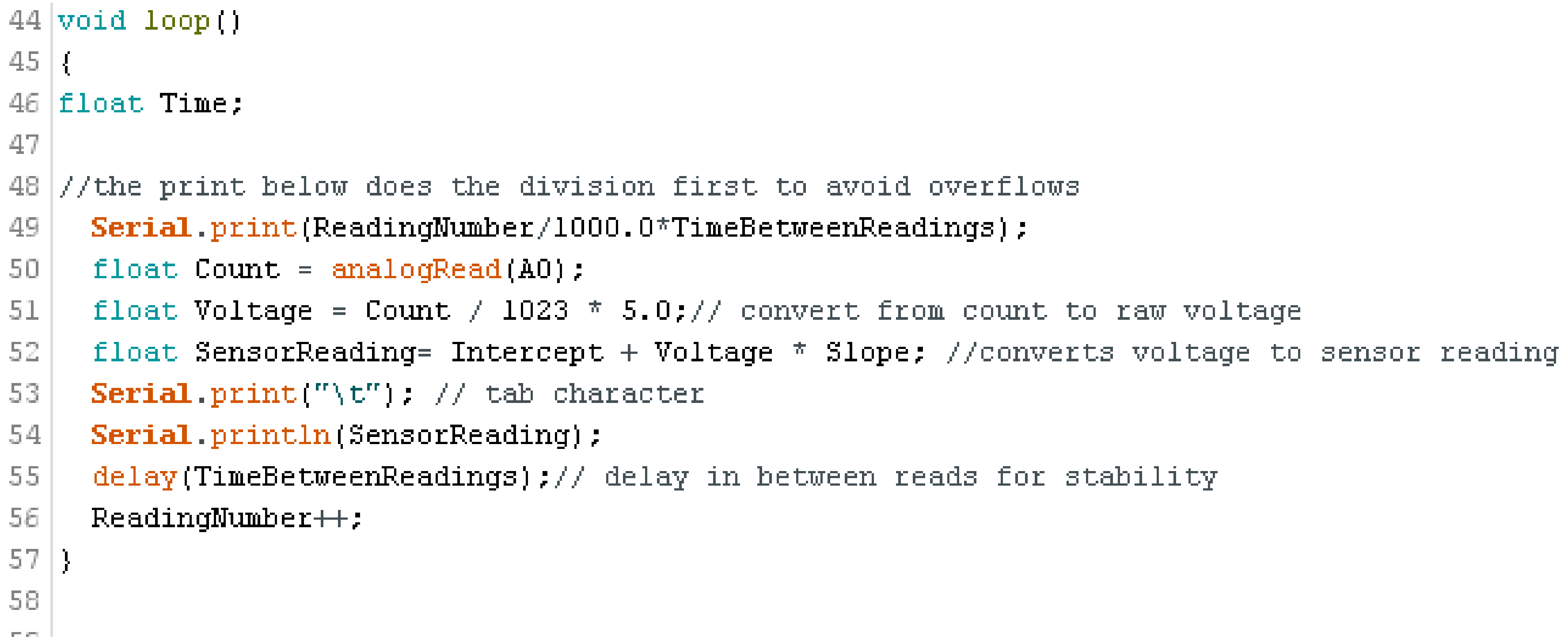


```
void setup()
{
  Serial.begin(9600); //initialize serial communication at 9600 baud
  Serial.println("Vernier Format 2");
  Serial.print(Sensor);
  Serial.print(" ");
  Serial.println("Readings taken using Ardunio");
  Serial.println("Data Set");
  Serial.print("Time");//long name
  Serial.print("\t"); //tab character
  Serial.println ("Force"); //change to match sensor
  Serial.print("t");//short name
  Serial.print("\t"); //tab character
  Serial.println ("F"); //short name, change to match sensor
  Serial.print("seconds");//units
  Serial.print("\t"); // tab character
  Serial.println ("newtons") ; //change to match sensor
}
```

Change to “Light”

Change to “Lux”

No Changes for loop() required

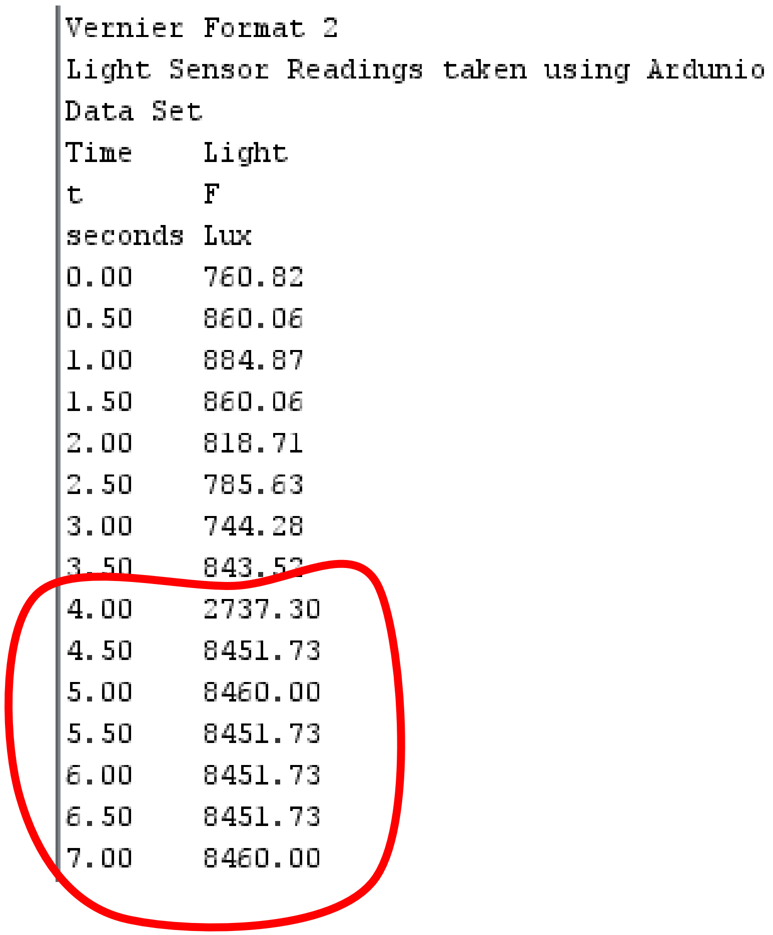


```
void loop()
{
float Time;

//the print below does the division first to avoid overflows
  Serial.print(ReadingNumber/1000.0*TimeBetweenReadings);
  float Count = analogRead(A0);
  float Voltage = Count / 1023 * 5.0;// convert from count to raw voltage
  float SensorReading= Intercept + Voltage * Slope; //converts voltage to sensor reading
  Serial.print("\t"); // tab character
  Serial.println(SensorReading);
  delay(TimeBetweenReadings);// delay in between reads for stability
  ReadingNumber++;
}

```

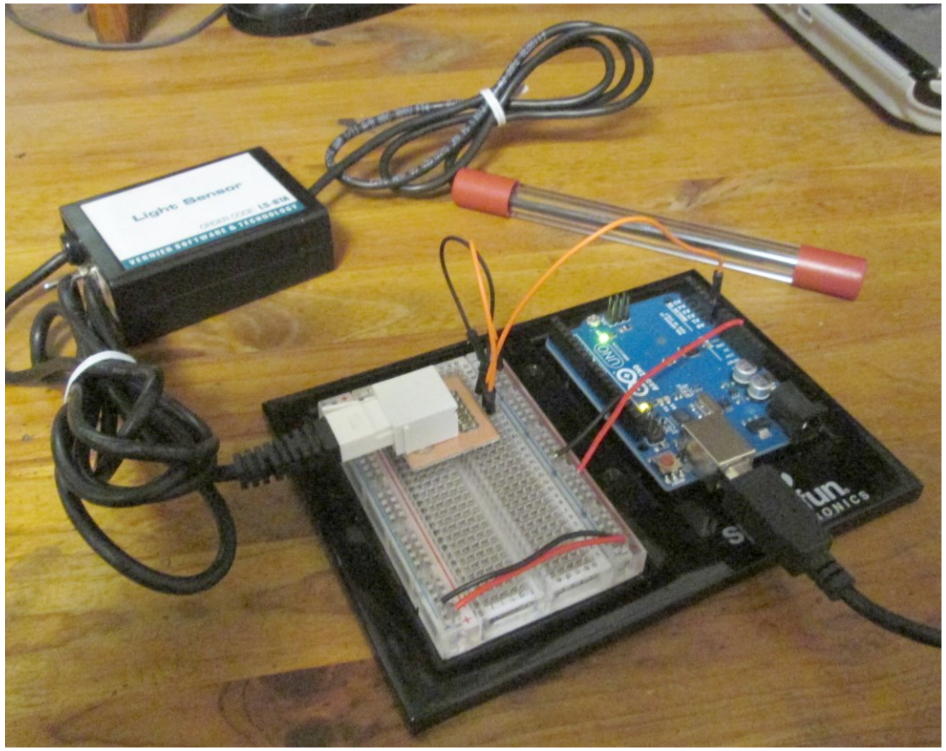
Output as expected



```
Vernier Format 2
Light Sensor Readings taken using Ardunio
Data Set
Time    Light
t       F
seconds Lux
0.00    760.82
0.50    860.06
1.00    884.87
1.50    860.06
2.00    818.71
2.50    785.63
3.00    744.28
3.50    843.52
4.00    2737.30
4.50    8451.73
5.00    8460.00
5.50    8451.73
6.00    8451.73
6.50    8451.73
7.00    8460.00
```

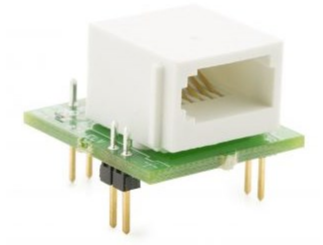
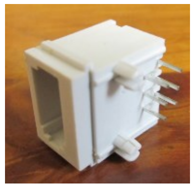
Bright Light was Turned on Here

2. Using Breadboard rather than Shield



Using Breadboard rather than Shield

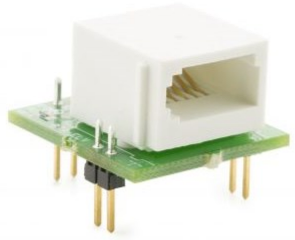
- Advantage over shield:
  - You can access Arduino pins!!
  - Also, you can have more than 2 sensors (2 is max for the shield)
- There are two ways to do this…
  1. "Analog Protoboard Adapter"
     - compatible with standard breadboard
  2. Or get this BTA from Sparkfun
     - Warning: It's <u>not</u> Breadboard Compatible
     - Must supply your own breakout board!!!

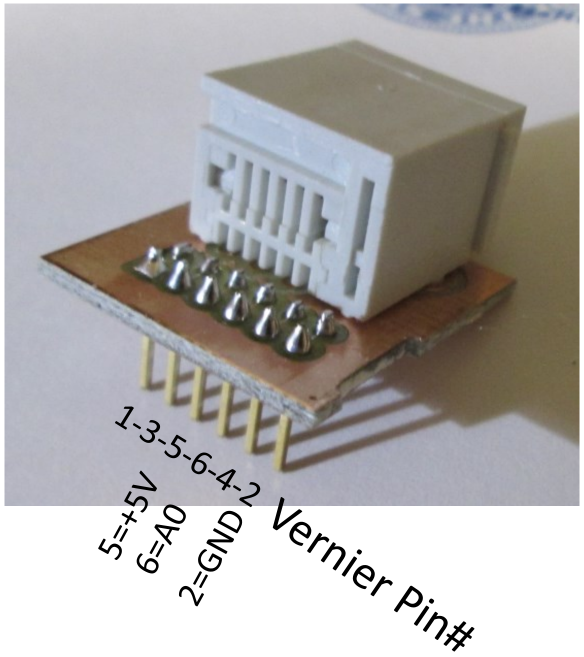
UW Protoboard

- We built our own protoboards
  - Since we had 3 connectors
- Protoboard pin #6 goes to an Analog pin (your choice: A0, A1, etc.)
  - Connect pin 5 to +5V, pin 2 to GND
- Modify "analogRead" to refer to chosen analog pin
- No more changes required
- Let us know if you're interested in these

Vernier protoboard

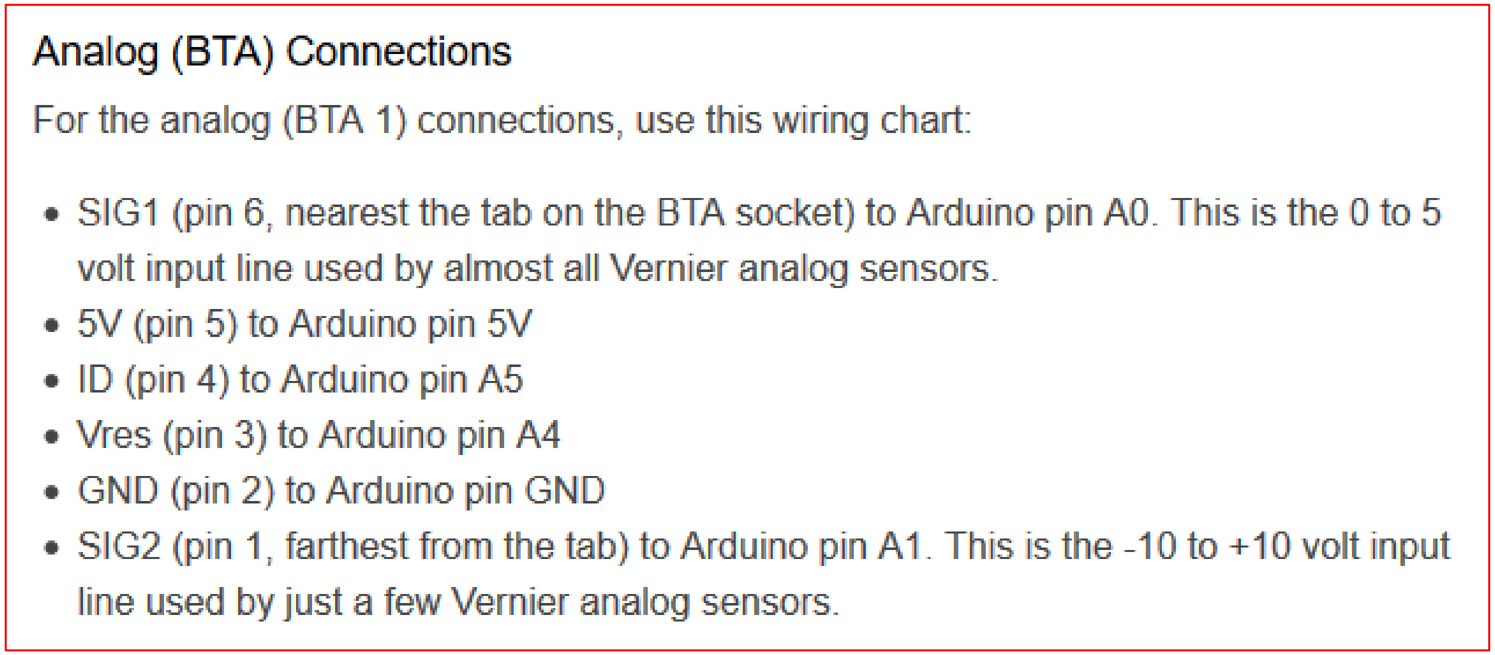


UW protoboard



Vernier Instructions

- If you have Vernier's protoboard check here:
  - https://www.vernier.com/engineering/arduino/connect/breadboard/
  - Here is excerpt from this web page:



Analog (BTA) Connections

For the analog (BTA 1) connections, use this wiring chart:

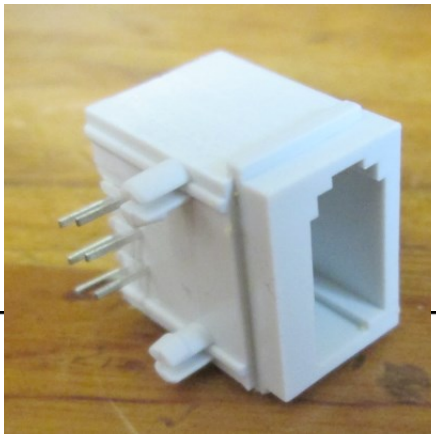
- SIG1 (pin 6, nearest the tab on the BTA socket) to Arduino pin A0. This is the 0 to 5 volt input line used by almost all Vernier analog sensors.
- 5V (pin 5) to Arduino pin 5V
- ID (pin 4) to Arduino pin A5
- Vres (pin 3) to Arduino pin A4
- GND (pin 2) to Arduino pin GND
- SIG2 (pin 1, farthest from the tab) to Arduino pin A1. This is the -10 to +10 volt input line used by just a few Vernier analog sensors.

How To Use Protoboards

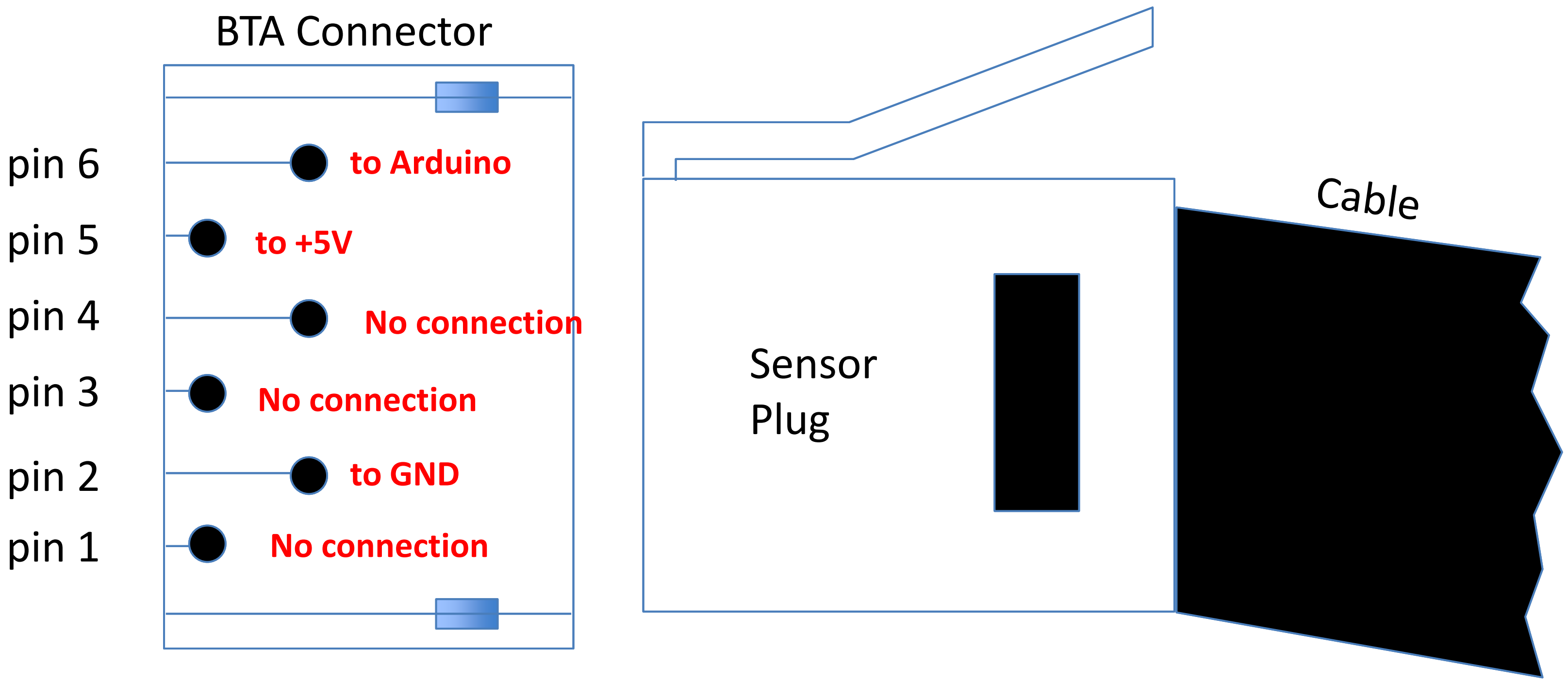
- Connect protoboard pin 6 to A0 (or A1, A2, etc.)
  - Arduino Uno has 6 analog inputs so you can have up to 6 protoboards and 6 Vernier sensors
  - If you have more sensors, get an Arduino Mega!
  - Or you can mix Vernier sensors with typical Arduino sensors (e.g., temperature, light, etc.)
- Software is same as that for Shield.
  - Except now you can utilize other Arduino pins for your own purposes.

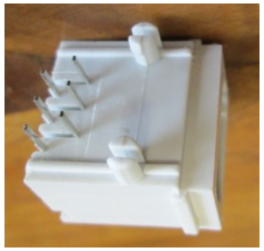
3. Cheapskate Method

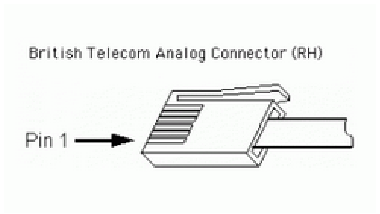
- Only requires the BTA connector from Sparkfun
- We need to connect to following pins:
  - pin 2: Ground
  - pin 5: +5 Volts
  - pin 6: Analog sensor output. Connect this to Arduino pin A0
    (or A1 or A2 or whatever)



Bottom View of BTA Connector







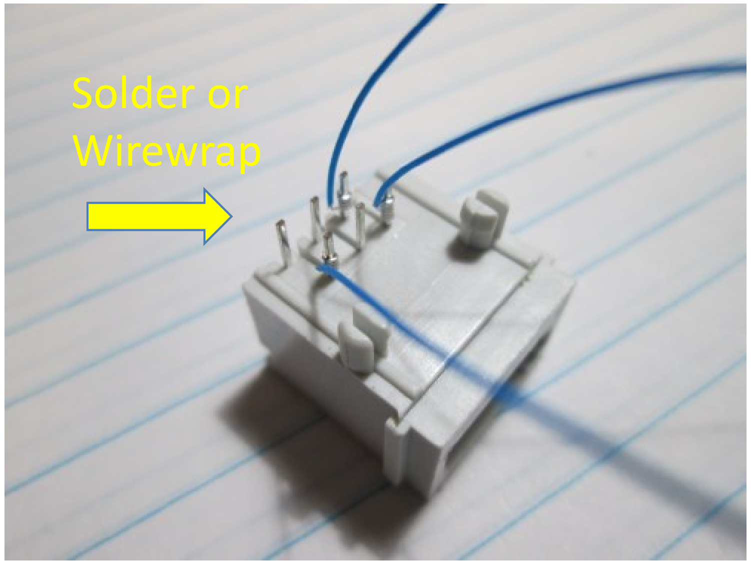
British Telecom Analog (BTA) – Right Hand

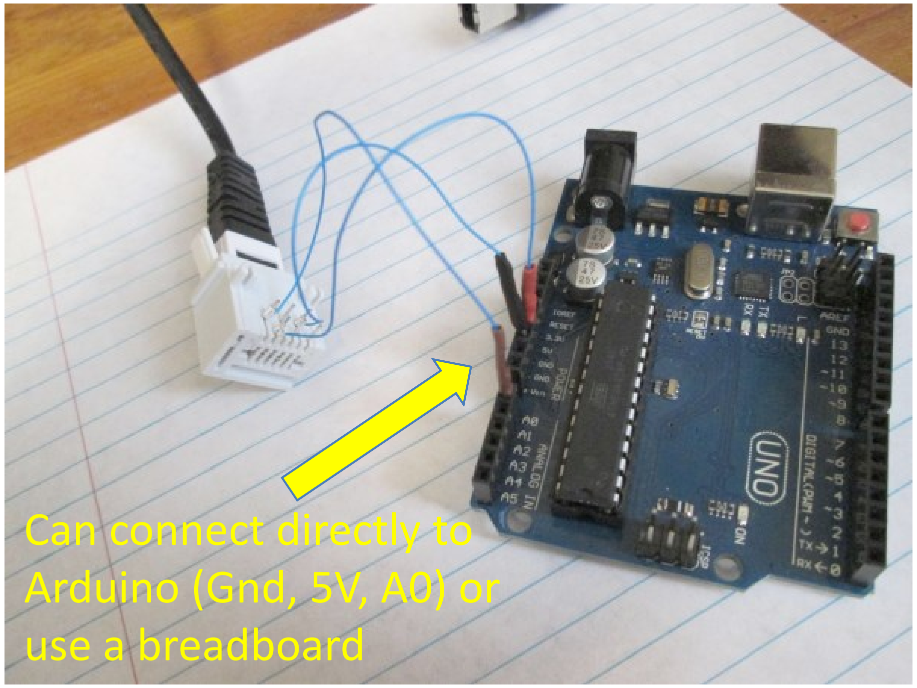
Pin 1 = Sensor output (+/-10V)
Pin 2 = GND
Pin 3 = Vres (resistance reference)
Pin 4 = AutoIDENT (not supported on all sensors)
Pin 5 = Power (+5VDC)
Pin 6 = Sensor output (0-5V)

Options for Connecting to Pins

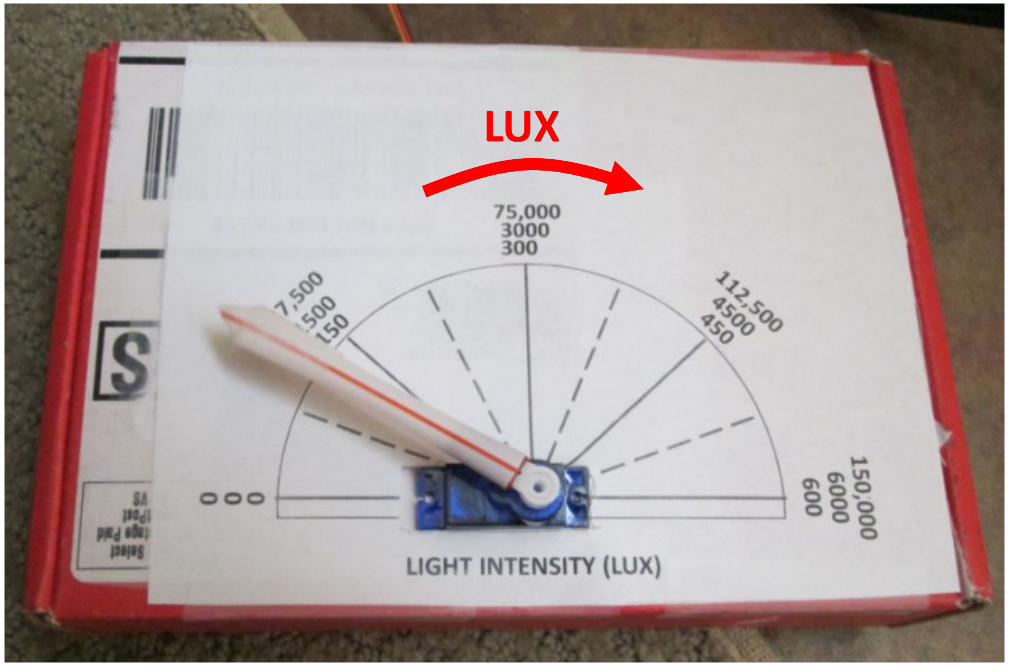
- Solder using small wire (26-30 Ga)
- <u>or</u> use a Wire Wrap tool (google it)
- As a precaution, protect result using
  - a dollop of epoxy
  - hot glue
  - shrink wrap for the small wires

Example using Wire Wrap





Project: Light Gauge

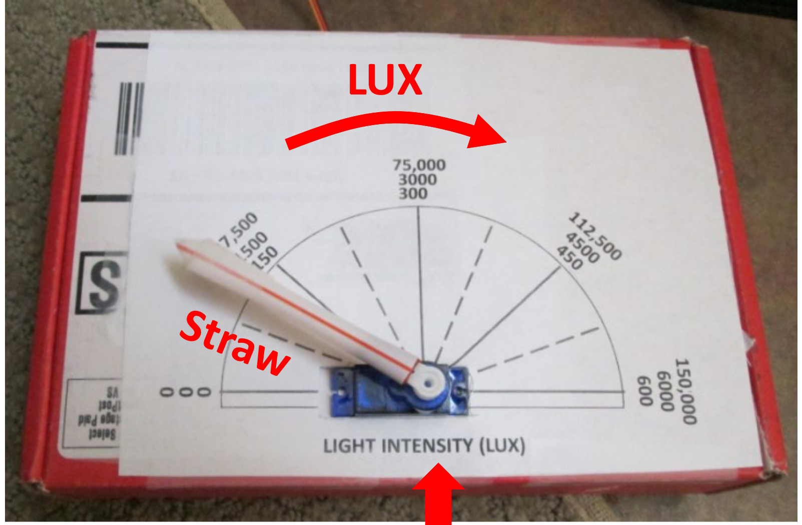


Example Application

Light Gauge with Dial

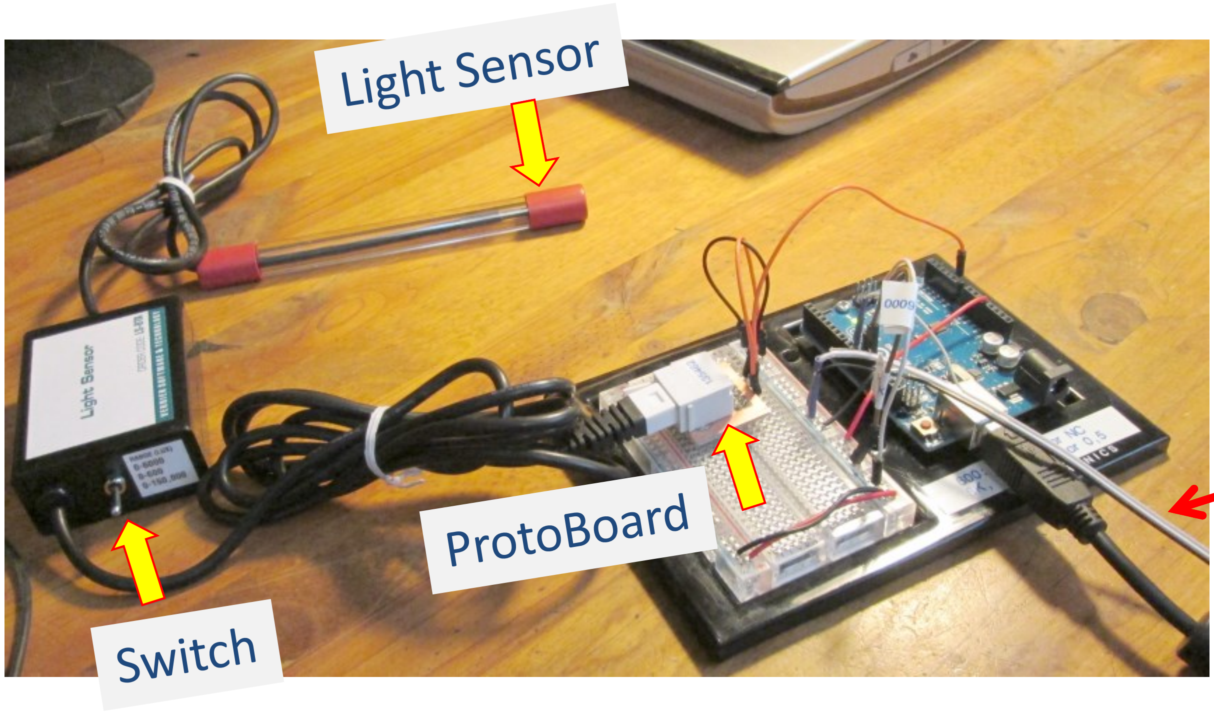
- Goal: Lets build a system based on the UW protoboard that utilizes the Vernier light sensor to create a Light gauge
  - The light gauge dial is based on a servo motor
  - User sets a jumper wire to tell software which Lux range has been selected on the Vernier sensor
- Protoboard allows us to access all Arduino pins, for great design flexibility!

Final Product



Sparkfun shipping box

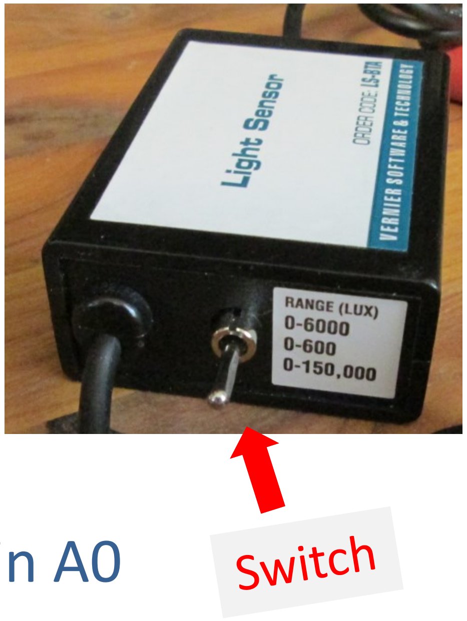
Servo Motor



Wires to Servo

LUX Ranges

- Vernier Light Sensor has three ranges
  - 0-6000 Lux (Dim conditions)
  - 0-600 Lux
  - 0-150000 Lux (Bright conditions)
- Arduino needs to know which range has been selected so it can compute LUX
  - Formula: LUX = Intercept + Voltage * Slope;
  - Where Voltage is determined from Arduino pin A0
  - Where Intercept and Slope depend on range selected
    - These values are found at the Vernier Web site:
    - https://www.vernier.com/support/manuals/



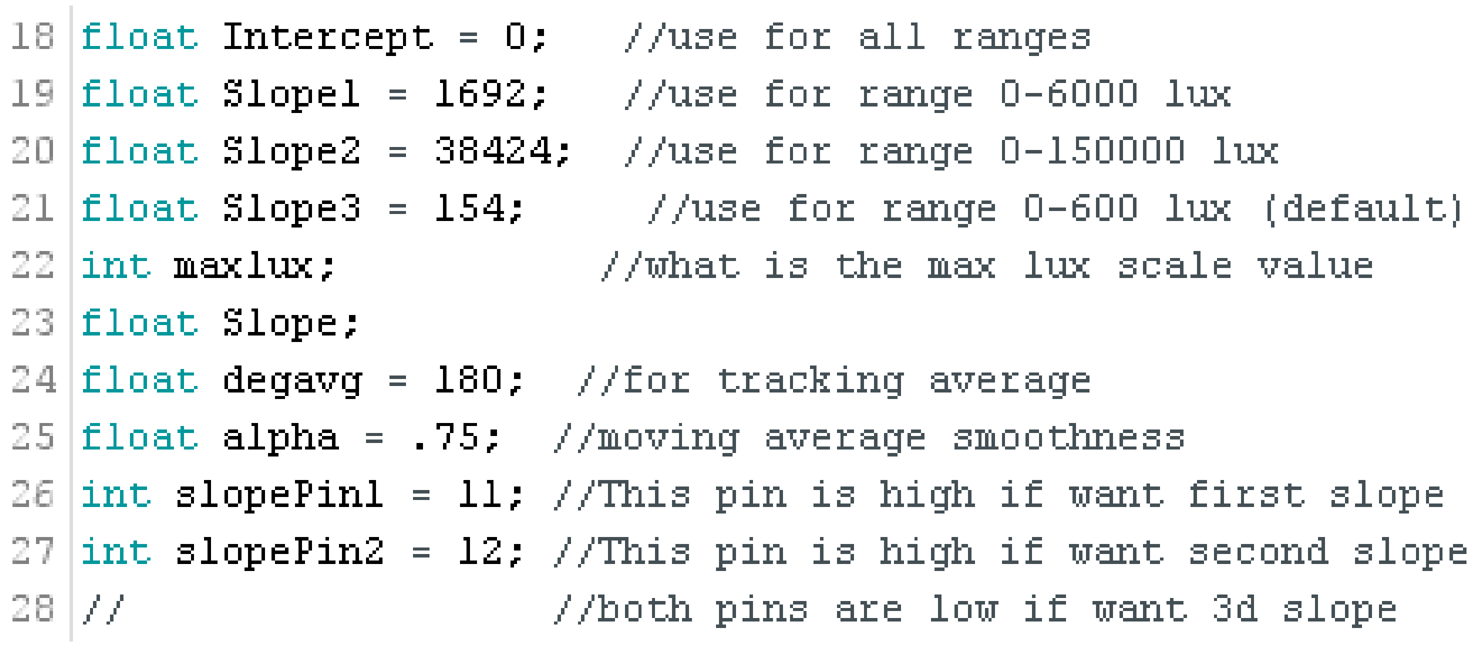
Using Jumpers to Select Range

Our program needs to know the switch setting on the light sensor in order to use the proper Slope and Intercept values. Here's one way:

- Jumpers on Arduino pins 11 and 12 indicate selected Lux range
- Connect jumpers to either +5 V or to GND as follows
  - **"0-6000" range**: pin 11 = 5V, pin 12 = GND:
  - **"0-150000" range**: pin 12 = 5V, pin 11 = GND:
  - **"0-600" range**: pin 11 and pin 12 unconnected **or**
    pin 11 = 5V, pin 12 = 5V **or**
    pin 11 = GND, pin 12 = GND:
- If the Vernier shield is used rather than protoboard, Arduino pins are not accessible. The software will detect that pins 11 and 12 are not connected and default to Case 3.

Declarations

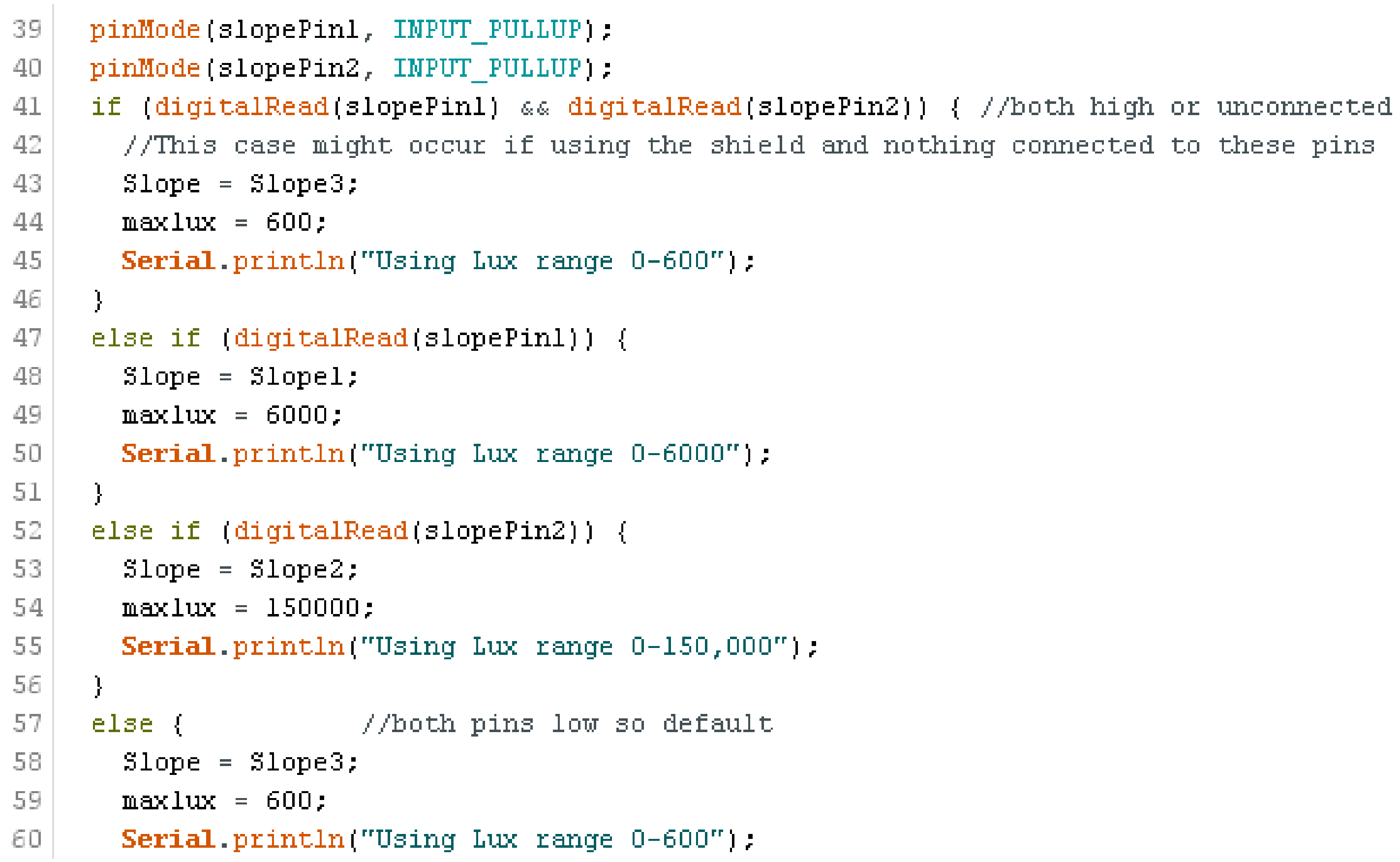
- Here we define the slope and intercept values that were gleaned from the Vernier datasheet



```
float Intercept = 0;   //use for all ranges
float Slope1 = 1692;   //use for range 0-6000 lux
float Slope2 = 38424;  //use for range 0-150000 lux
float Slope3 = 154;     //use for range 0-600 lux (default)
int maxlux;           //what is the max lux scale value
float Slope;
float degavg = 180;  //for tracking average
float alpha = .75;  //moving average smoothness
int slopePin1 = 11; //This pin is high if want first slope
int slopePin2 = 12; //This pin is high if want second slope
//                  //both pins are low if want 3d slope
```

setup()

setup() determines which Lux range has been set



```
pinMode(slopePin1, INPUT_PULLUP);
pinMode(slopePin2, INPUT_PULLUP);
if (digitalRead(slopePin1) && digitalRead(slopePin2)) { //both high or unconnected
  //This case might occur if using the shield and nothing connected to these pins
  Slope = Slope3;
  maxlux = 600;
  Serial.println("Using Lux range 0-600");
}
else if (digitalRead(slopePin1)) {
  Slope = Slope1;
  maxlux = 6000;
  Serial.println("Using Lux range 0-6000");
}
else if (digitalRead(slopePin2)) {
  Slope = Slope2;
  maxlux = 150000;
  Serial.println("Using Lux range 0-150,000");
}
else {           //both pins low so default
  Slope = Slope3;
  maxlux = 600;
  Serial.println("Using Lux range 0-600");
```

Using Servo Motor as a Dial

- Include the servo library:
  - #include <Servo.h>
- Create a servo object called 'myservo' (call it anything you want):
  - Servo myservo;
- In setup() assign pin 9 (ServoPin) to the servo
  - myservo.attach(ServoPin);
- In loop() compute the LUX value
  - float Voltage = Count / 1023 * 5.0;// convert from count to raw voltage
  - float SensorReading = Intercept + Voltage * Slope;
- Convert LUX to an angle (0 to 180 deg) for dial position
  - int deg = map(SensorReading, 0, maxlux, 180 , 0);
  - deg = constrain(deg, 0, 180); //value must stay between 0, 180
- Make the servo turn to this degree value
  - myservo.write(deg);

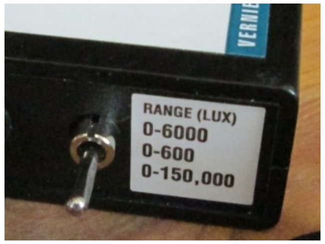
Remark #1

- Consider the instruction
    - int deg = map(SensorReading, 0, maxlux, 0, 180);
  - deg is # degrees to turn servo in <u>counter-clockwise</u> direction
  - However, we want dial to turn clockwise as Lux increases. So the map instruction is modified:
    - int deg = map(SensorReading, 0, maxlux, 180 , 0);
    - Now when SensorReading is minimum, the servo turns to 180 degrees (dial pointing left)
    - When SensorReading is maximum, servo turns to 0 degrees (dial all the way to right)

Remark #2

- Consider the instruction
    - myservo.write(deg);
  - This makes the servo turn to an angle of deg degrees
  - But sometimes deg is a little noisy and the dial appears “jerky”.
  - Can we make the action a little smoother?
- Using exponential averaging is an easy fix!
  - Declarations:
    - float degavg = 180; //initial value for average (set @ min brightness)
    - float alpha = .75; //values closer to 1.0 are smoother
  - In loop() compute the average as:
    - degavg = alpha * degavg + (1 - alpha) * deg; //running average
    - myservo.write(degavg);

User sets Arduino Pin 11, 12 to correspond to Light Sensor setting.

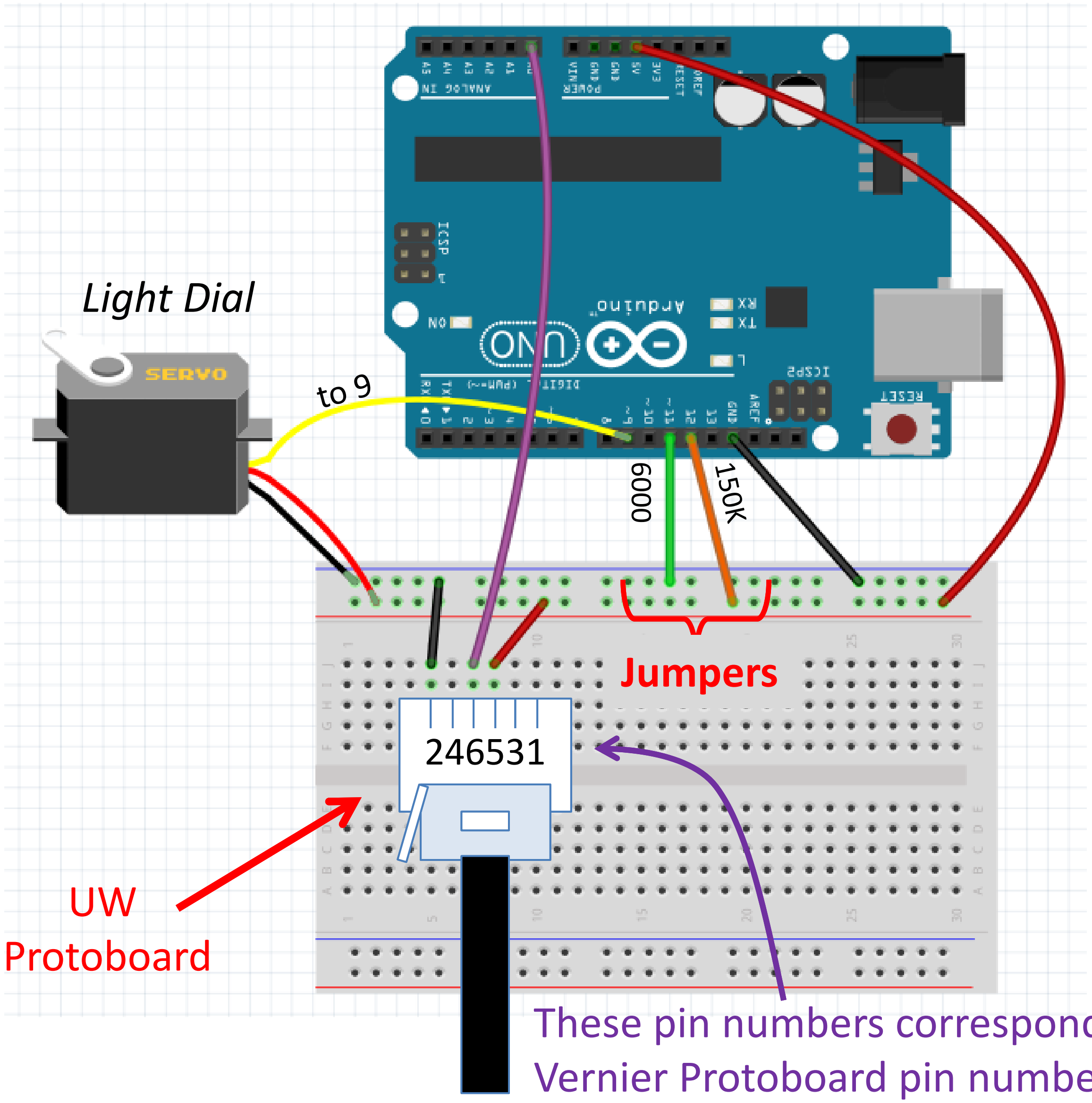


Jumpers:

0-6000: pin 11 to +5V
pin 12 to GND

0-150K: pin 12 to +5V
pin 11 to GND

0-600: pin 11 to GND
pin 12 to GND
or both unconnected
or both to +5V


Light Dial
SERVO
to 9
6000
150K
Jumpers
246531
UW Protoboard
These pin numbers correspond to Vernier Protoboard pin numbers

To Vernier Sensor