



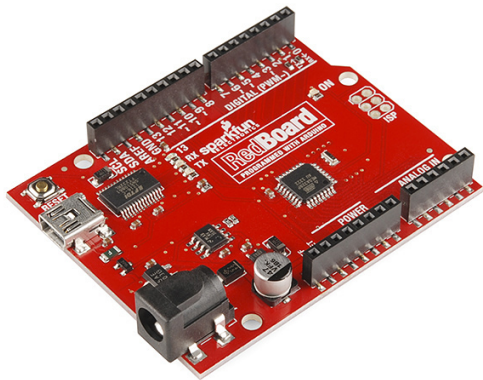
# **Simple Tools to Enhance Student Engagement through Processing**

SparkFun Electronics

Derek Runberg, Angela Sheehan & Jeff  
Branson

# About us...

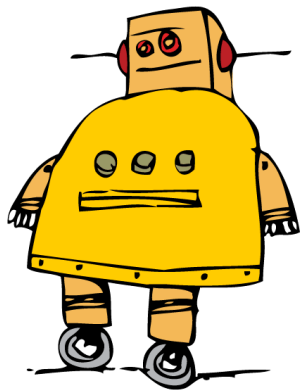
Founded in 2003 by Nathan Seidle.



[This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 United States License.](https://creativecommons.org/licenses/by-sa/3.0/us/)

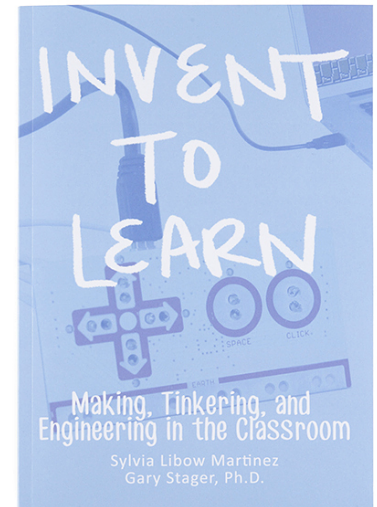
# Inspiration for this workshop...

## “Teachers that Make, Make Great Teachers”



**instructables.com**

THE WORLD'S BIGGEST SHOW & TELL



[This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 United States License.](#)

# Where can I get this later?

You can find this presentation at:

[http://bit.ly/NSTA14\\_SFE\\_P](http://bit.ly/NSTA14_SFE_P)

As well as all materials used at our resources  
page:

<https://learn.sparkfun.com/resources/69>



[This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 United States License.](#)

# Goals for today...

1. Learn some basics of programming
2. Build digital tools that empower student
3. discuss how you could incorporate this into your classroom



# What is Processing and Why do I care?

Processing is a open source programming language designed with ***you*** in mind.

Designed for creating visuals, dashboards, computational art

Simple, easy to use, easy to read and free!



[This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 United States License.](https://creativecommons.org/licenses/by-sa/3.0/)



# What is Processing and Why do I care?

Giving students the opportunity to:

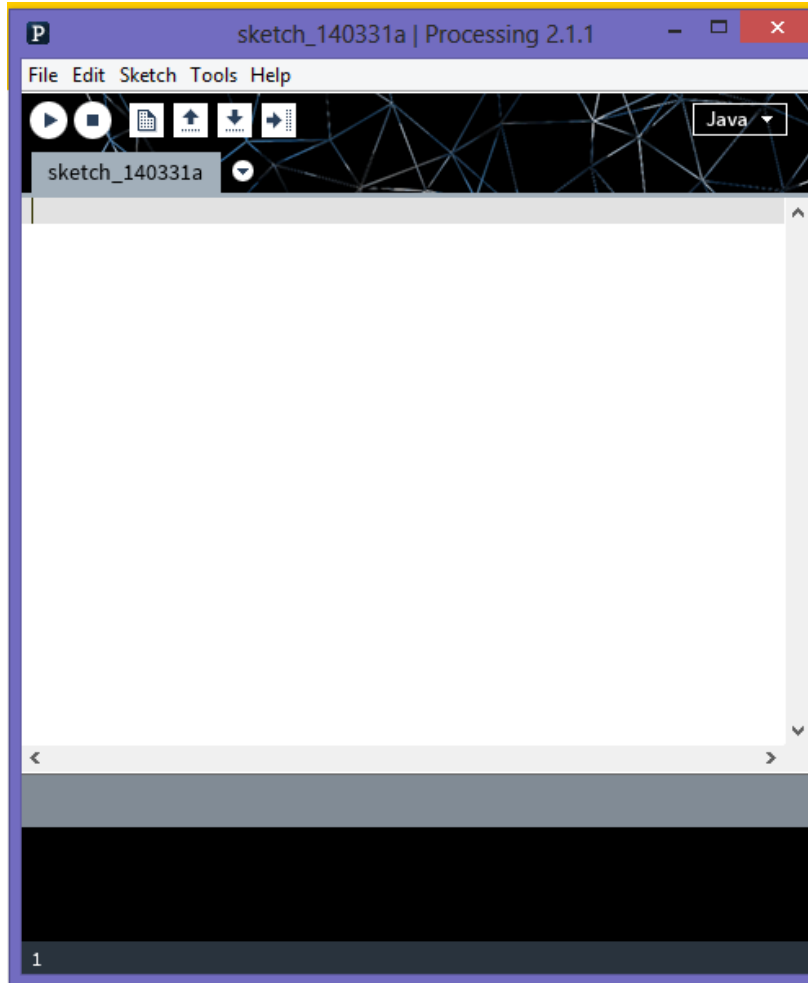
- Design and create their own tools
- Create and manage instrumentation that they need
- Integrate NGSS engineering standards into their inquiry



[This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 United States License.](https://creativecommons.org/licenses/by-sa/3.0/)




# Jumping Right in!



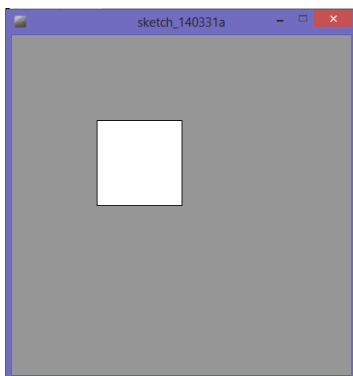
[This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 United States License.](https://creativecommons.org/licenses/by-sa/3.0/)



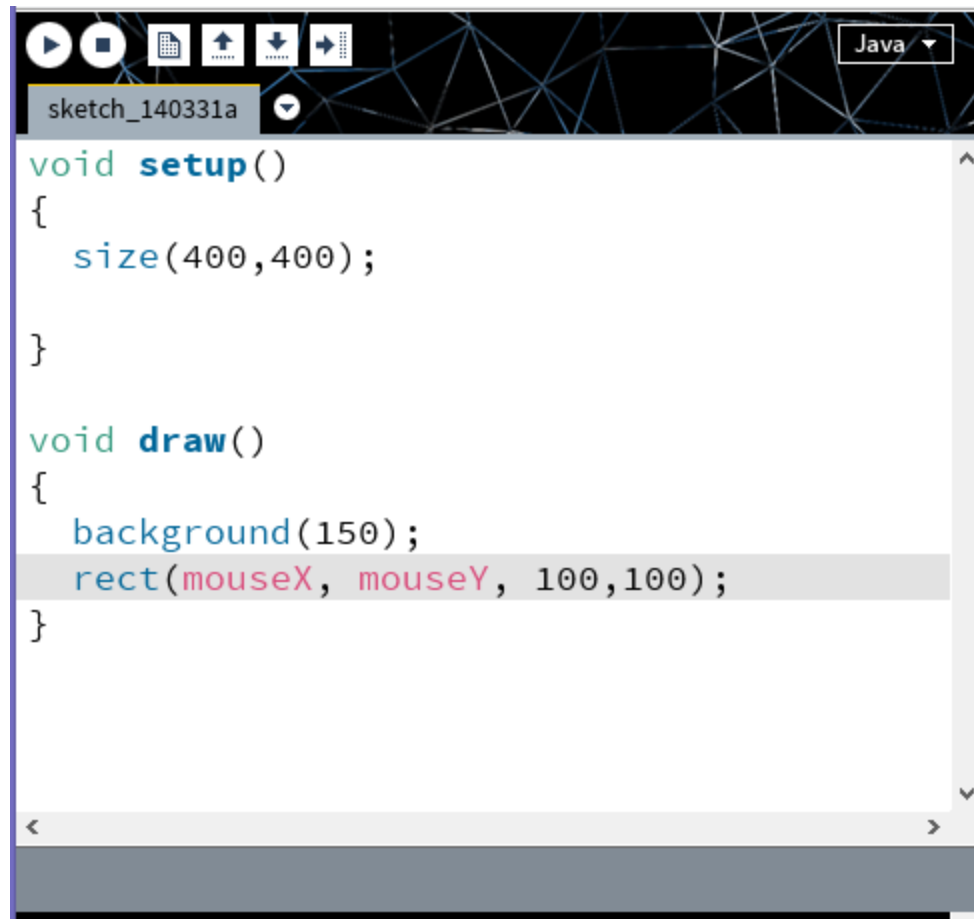
# Basic Rectangle

The image shows the top toolbar of the Processing IDE. It contains icons for running (play), stopping (square), opening a file (document), saving (floppy disk), and other standard development tools. Below the icons is a text field containing the name of the current sketch, 'sketch\_140331a', and a dropdown arrow.

```
void setup()  
{  
  size(400,400);  
  background(150);  
}  
  
void draw()  
{  
  rect(100,100,100,100);  
}
```

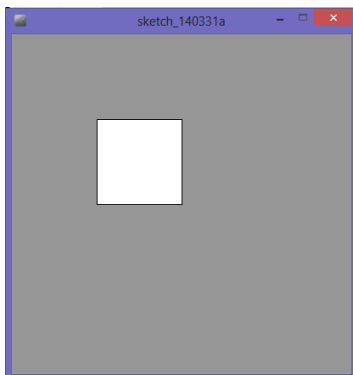


# Motion with variables




```
void setup()
{
  size(400,400);
}

void draw()
{
  background(150);
  rect(mouseX, mouseY, 100,100);
}
```

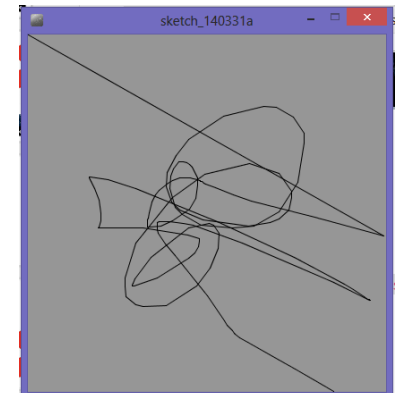


# A simple drawing tool with Variables




```
void setup()
{
  size(400,400);
  background(150);
}

void draw()
{
  line(mouseX,mouseY,mouseX, mouseY);
}
```



# Movement through Computation!



```
sketch_140331a
int x=0;

void setup()
{
  size(400,400);
}

void draw()
{
  background(150);
  rect(x, mouseY, 100,100);
  x++;
  if(x>=400)
  {
    x=0;
  }
}
```

16



# Where can we get more data?

In Processing we have been using variables that are easily available through the computer...

**What are some other sources of data?**

***Where can we get useful physical data?***



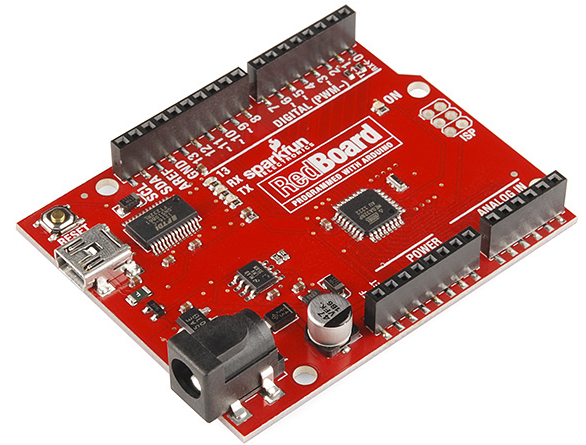
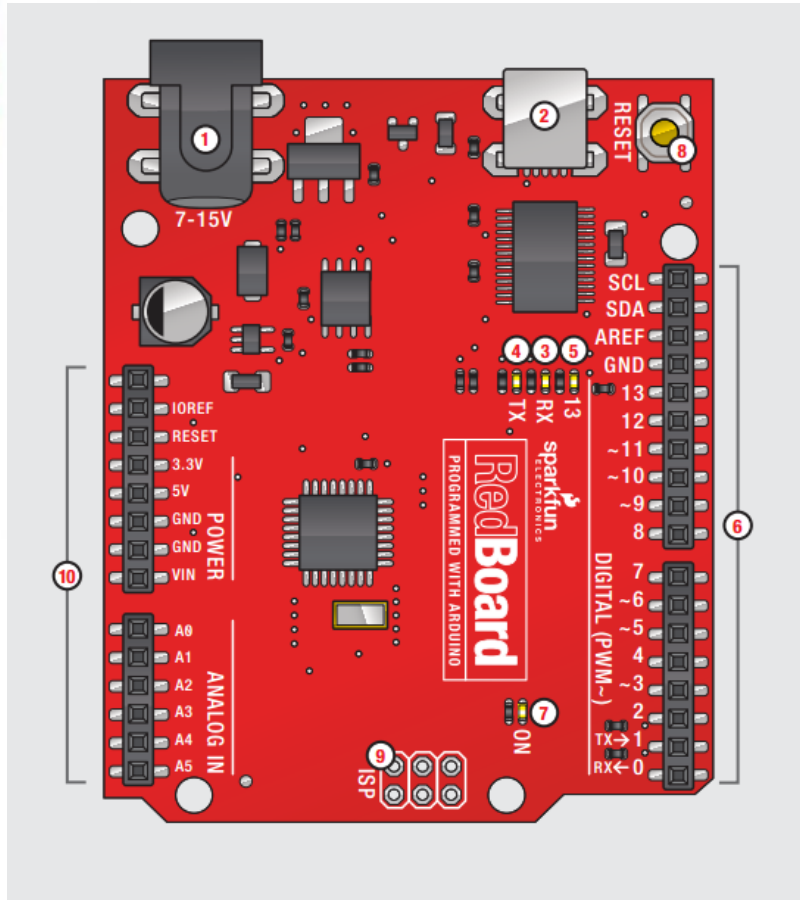


# Building Instrumentation

- Foundational understanding of what they are measuring
- Problem and inquiry based  
(Engineering and Science)
- Student build their knowledge through building (Constructionism)



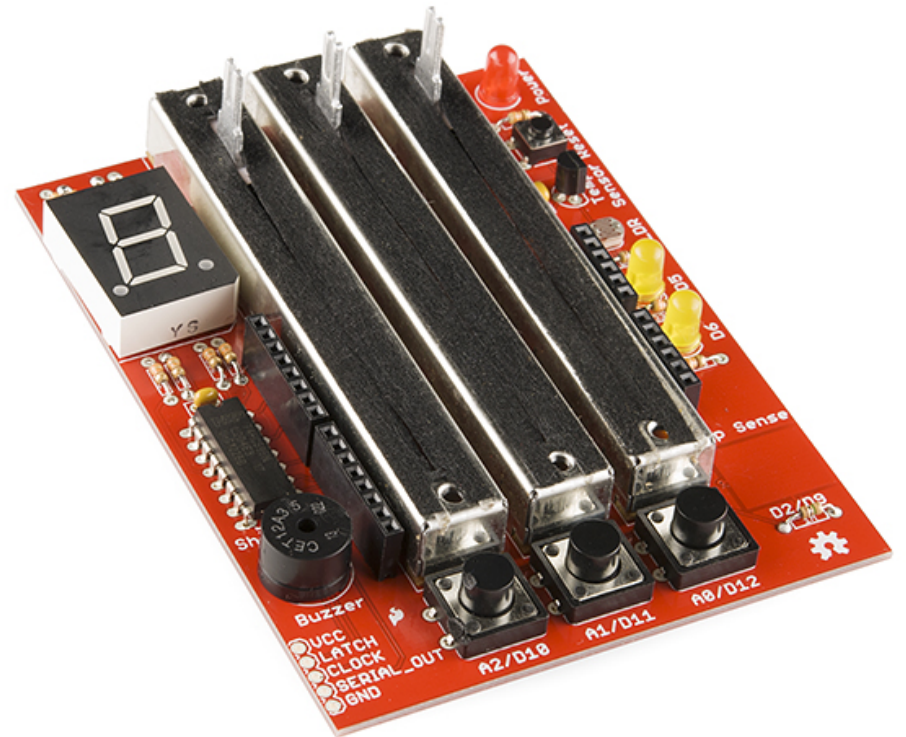
# The Arduino!



[This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 United States License.](https://creativecommons.org/licenses/by-sa/3.0/us/)

# The Danger Shield

- 3 Sliders
- 3 Buttons
- 1 light sensor
- 1 temperature sensor
- 1 7-segment
- 2 LEDs
- 1 buzzer





# What do we get from the danger shield?

- 3 slider values
- temperature value
- light value
- 3 button values

Through serial communication as a comma separated string of values





# NSTA\_Example1

- Find and open NSTA\_Examples on the Desktop
- Open NSTA\_Example1
- Make sure your Arduino/ Danger Shield is plugged in!



# NSTA\_Example1

```
NSTA_Example1 SerialEvent
import processing.serial.*; // Import the Serial Library

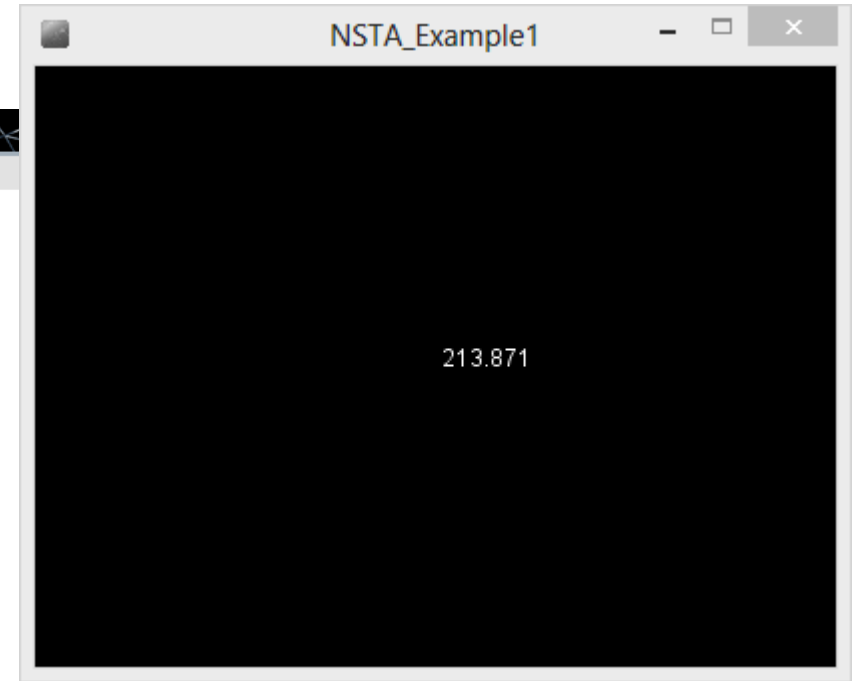
Serial myPort;           // The serial port

int xPos = 0;
int x2Pos= 0; // horizontal position of the graph
int x3Pos= 0;

float slider1=0; //variable for slider 1
float slider2 = 0; //variable for slider 2
float slider3 = 0; //variable for slider 3
float temp= 0; //variable for temperature sensor
float light= 0; //variable for light sensor
float button1 = 0; //variable for button 1
float buttton2= 0; //variable for button 2
float button3= 0; //variable for button 3

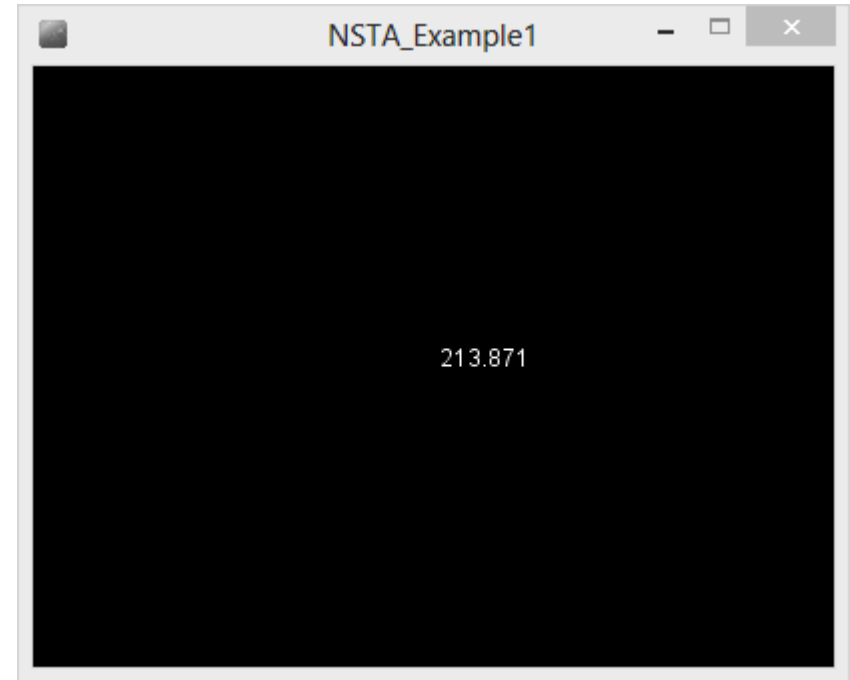
void setup ()
{
  size(400, 300); // set the window size
  println(Serial.list()); //print out the list of serial ports
  myPort = new Serial(this, Serial.list()[0], 9600); //connect to given serial port (change 0 to your port number)
  myPort.bufferUntil('\n');
}

void draw ()
{
  background(0); //black background
  fill(255); //fill white
  text(light,width/2,height/2); //text readout with sensor value (change the value to different variables from above)
}
```



# NSTA\_Example1

Substitute any variable from the list in the text function!



```
slider1=0; //variable for slider 1
slider2 = 0; //variable for slider 2
slider3 = 0; //variable for slider 3
temp= 0; //variable for temperature sensor
light= 0; //varibale for light sensor
button1 = 0; //variable for button 1
buttton2= 0; //variable for button 2
button3= 0; //variable for button 3
```



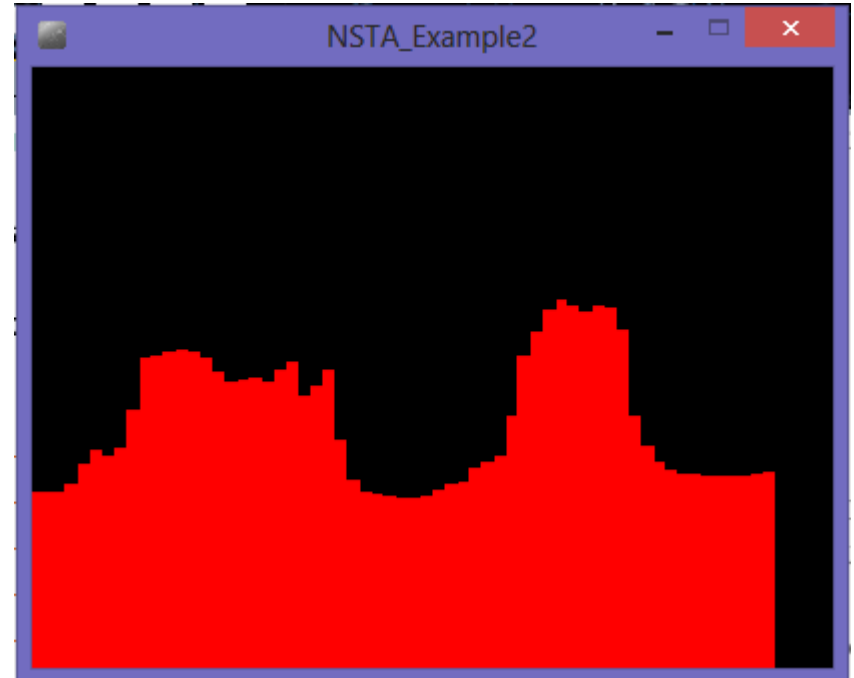
# NSTA\_Example2

```
void setup () {  
  
  size(400, 300); // set the window size  
  println(Serial.list()); //print list of serial ports  
  myPort = new Serial(this, Serial.list()[9], 9600); //select your serial port (change  
  myPort.bufferUntil('\n');  
  background(0);  
}  
void draw () {  
  // everything happens in the serialEvent()  
  
  stroke(255,0,0); //change line color to red  
  line(xPos,height,xPos,height-light); //create a lines that moves across the screen, c  
  
  xPos++; //incremenet the x position of the line across the screen  
  
  if(xPos>=width) //if line gets to edge of screen, wipe the screen and move back to th  
  {  
    background(0);  
    xPos=0;  
  }  
}
```

# NSTA\_Example2

Change the “light” variable in the line() function to any other variable

Change the RGB value of the stroke() function  
`stroke(Red, Green, Blue);`





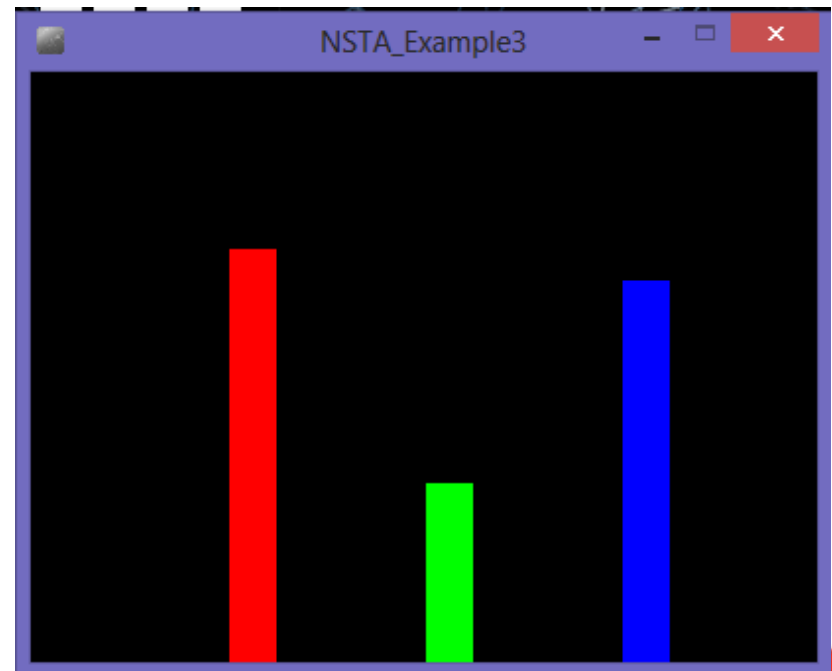
# NSTA\_Example3

```
void setup ()
{
  size(400, 300); // set the window size
  println(Serial.list()); //print out list of serial ports
  myPort = new Serial(this, Serial.list()[9], 9600); //change 0 to your serial port nu
  myPort.bufferUntil('\n');
}

void draw ()
{
  background(0); //set background to black
  fill(255,0,0); //fill red
  rect(100,height,25,-light); //bar graph for light (x,y,w,h)
  fill(0,255,0); //fill green
  rect(200,height,25,-slider2); //bar graph for slider 2
  fill(0,0,255); //fill blue
  rect(300,height,25,-slider3); //bar graph for slider 3
}
```

# NSTA\_Example3

Find the variables used  
in the rect() function.



Change them to a variable of your choice

Change the fill color to a color of your choice

How could you also add text? hint: example1





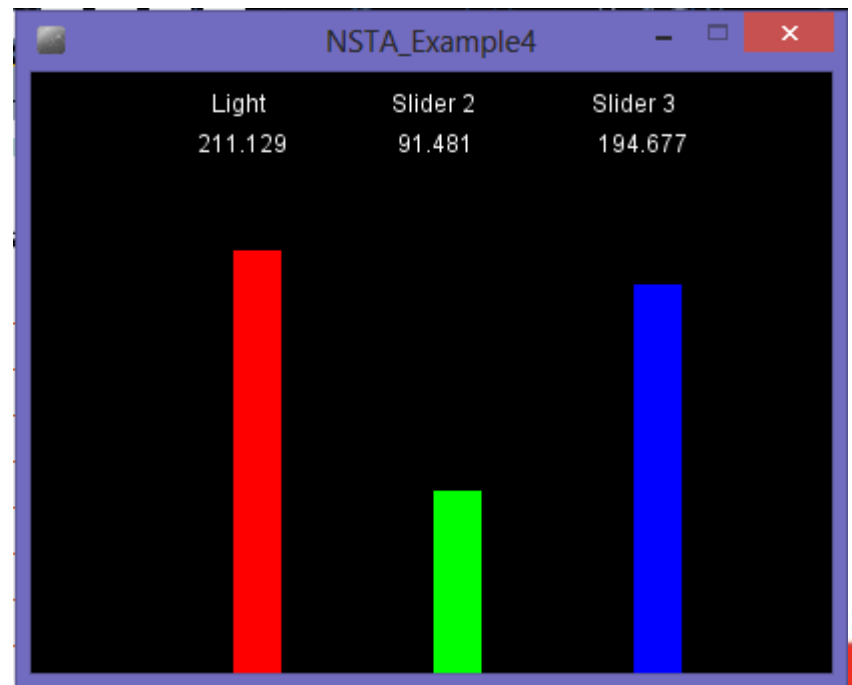
# NSTA\_Example4

```
void setup ()
{
  size(400, 300); // set the window size
  println(Serial.list()); //print out list of serial ports
  myPort = new Serial(this, Serial.list()[9], 9600); //select your port (char
  myPort.bufferUntil('\n');
}

void draw ()
{
  background(0); //background set to black
  fill(255, 0, 0); //fill red
  rect(100, height, 25, -light); //bar graph for light sensor
  fill(255); //fill white for text
  text(light, 80, 40); //light value print out
  text("Light", 90, 20); //label of light
  fill(0, 255, 0); //fill green
  rect(200, height, 25, -slider2); //bar graph for slider 2
  fill(255); //fill white for text
  text("Slider 2", 180, 20); //slider 2 label
  text(slider2, 180, 40); //slider2 value print out
  fill(0, 0, 255); //fill blue
  rect(300, height, 25, -slider3); //slider for slider 3
  fill(255); //fill white for text
  text("Slider 3", 280, 20); //label for slider 3
  text(slider3, 280, 40); //print out of slider 3
}
```

# NSTA\_Example4

Change the slider variables to slider1 and slider2.



Change the labelling to “slider 1” and “slider 2”

change the fill value for the bar graphs to the variable they are reading out!





# NSTA\_Example4

Example for substituting a variable for a static value.

```
fill(255, 0, 0); //fill red
rect(100, height, 25, -light); //bar graph for light sensor
fill(255); //fill white for text
text(light, 80, 40); //light value print out
text("Light", 90, 20); //label of light
```

Vs.

```
fill(light, 0, 0); //fill red
rect(100, height, 25, -light); //bar graph for light sensor
fill(255); //fill white for text
text(light, 80, 40); //light value print out
text("Light", 90, 20); //label of light
```



# NSTA\_Example5

```
void draw ()
{
    for(int x=width;x>=0;x-=50)    //used to create grid lines
    {
        stroke(255);    //white stroke color
        strokeWeight(1);    //line thickness
        line(x,0,x,height);    //vertical lines
        text(x,5,height-x);    //grid labels
        line(0,x,width,x);    //horizontal lines
    }

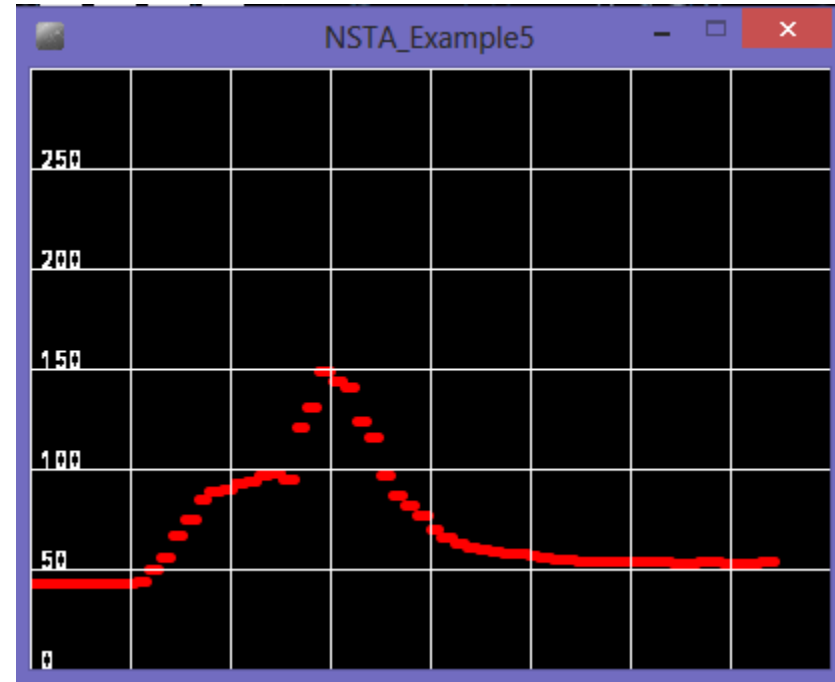
    stroke(255,0,0);    //red stroke color for point
    strokeWeight(5);    //size of point (5 pixels)
    point(xPos,height-light);    //point at xPos, light value
    xPos++;    //increment xPos

    if(xPos>=width)    //if xPos is off the screen reset to start
    {
        xPos=0;
        background(0);
    }
}
```

# NSTA\_Example5

Change the  
variable to a  
different one

Add a second  
channel!



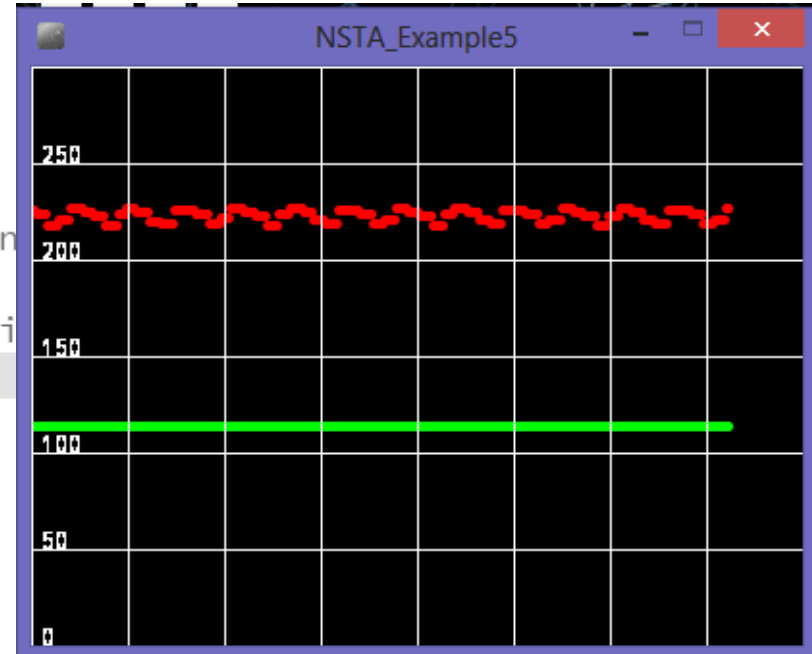
# NSTA\_Example5

Adding a second channel! Find this...

```
stroke(255,0,0); //red stroke color for point
strokeWeight(5); //size of point (5 pixels)
point(xPos,height-light); //point at xPos, light value
xPos++; //increment xPos
```

Hack it to this:

```
stroke(255,0,0); //red stroke color for point
strokeWeight(5); //size of point (5 pixels)
point(xPos,height-light); //point at xPos, li
stroke(0,255,0);
point(xPos,height-slider1);
xPos++; //increment xPos
```





# NSTA\_Example6

```
void draw ()
{
    for(int x=width;x>=0;x-=grid) //create grid lines
    {
        stroke(255);
        strokeWeight(1);
        line(x,0,x,height);
        text(x,5,height-x);
        line(0,x,width,x);
    }

    stroke(255,0,0); //red stroke color
    strokeWeight(5); //point thickness
    point(xPos,height-light); //plot the point using the light
    xPos++; //increment x position of plot

    if(xPos>=width) //if x postion is off screen, reset
    {
        xPos=0;
        background(0);
    }

    output.println(light); //log the value of light
}

void keyPressed() //if a key is pressed close the log and
{
    output.flush();
    output.close();
    exit();
}
```



# NSTA\_Example6

The createWriter() function:

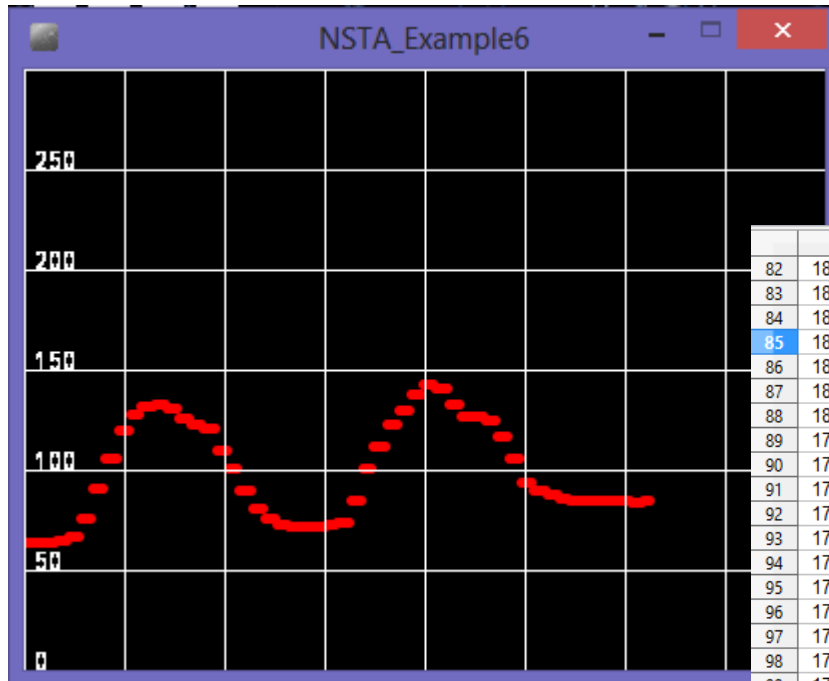
Logs whatever data we write to it for the entire time we have the sketch open.

Creates a .csv file which can be opened with MS Excel or OpenOffice





# NSTA\_Example6

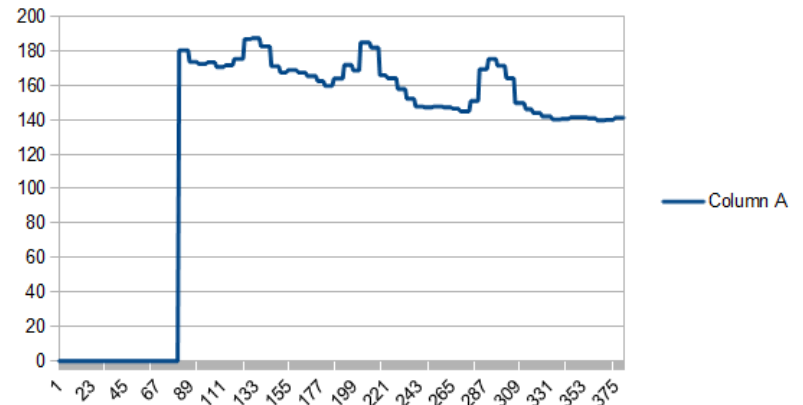


	A
82	180.21994
83	180.21994
84	180.21994
85	180.21994
86	180.21994
87	180.21994
88	180.21994
89	173.48973
90	173.48973
91	173.48973
92	173.48973
93	173.48973
94	173.48973
95	172.24341
96	172.24341
97	172.24341
98	172.24341
99	172.24341
100	172.24341
101	173.24048
102	173.24048
103	173.24048
104	173.24048
105	173.24048
106	173.24048
107	170.49854
108	170.49854
109	170.49854
110	170.49854

ne Share View

↑ NSTA\_Processing\_Master ▸ NSTA\_Example6

Name	Date modified
NSTA_Example6	3/28/2014 12:01 PM
SerialEvent	3/27/2014 2:54 PM
Values	3/31/2014 10:25 AM



[This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 United States License.](https://creativecommons.org/licenses/by-sa/3.0/us/)

# NSTA\_Example6

1. Run the sketch for roughly 1 minute  
(30 fps\* 60 seconds= 1800 data points)
2. Tap a key on the keyboard to stop logging and write to file
3. Click on **Sketch>Show Sketch Folder...**
4. Double click on Values.csv to open in OpenOffice



# Processing Examples

Pulling data from other sources:

- Vernier Shield
- NOAA Data From the Web
- High Altitude Balloon Example
- Sparkfun Stock Graph



[This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 United States License.](https://creativecommons.org/licenses/by-sa/3.0/us/)



Questions?!

[Curriculum@sparkfun.com](mailto:Curriculum@sparkfun.com)

