**Introdução ao Arduino e Processing - Sparkfun Eletronics - Atlas 2013**

**Processing Installation**

**For Linux:**

- Download the .tar.gz file to your home directory, then open a terminal window and type:

Tar xvfz processing-xxxx.tgz

(replace xxxx with the rest of the file’s name, which is the version number)

- This will create a folder named processing-1.5 or something similar. Then change to that directory:

cd processing-xxxx

and run processing:

./processing

**For Mac:**

Double-click the .dmg file and drag the Processing icon from inside this file to your applications folder, or any other location on your computer. Double click the Processing icon to start Processing.

**For Windows:**

Double-click the .zip file and drag the folder inside labeled Processing to a location on your hard drive. Double click the Processing icon to start Processing.

If you are stuck go to http://wiki.processing.org/index.php/Troubleshooting for help.

**Processing Introduction**

- Processing is a free, open source, cross-platform programming language and environment for people who want to create images, animations, and interactions.

- Created in 2001 by Casey Reas and Ben Fry at the MIT Media Lab.

- Downloads, updates, reference, forums, etc. at: http://processing.org

- A sketch is a file or project you create in Processing. When you first open up a new sketch it will be completely blank

- Tools buttons:

run / stop / new / save

- Help for native functions:

highlight a function or a word, right click and select "Find in Reference"

**Arduino Code**

// Serial variables

char START\_BYTE = '\*';

char DELIMITER = ',';

char END\_BYTE = '#';

// General Variables

int sensValue = 0;

int bt1Value = 0;

int bt2Value = 0;

// Define pins

int led\_pin = 3;

int b1\_pin = 4;

int b2\_pin = 5;

int sens\_pin = 0;

// Wait for byte received

void establishContact()

{

while(Serial.available() <= 0)

{

Serial.println("Hello"); // string Hello with '\n' at the end

delay(300); // 300 ms

}

}

void setup()

{

// Define pins' modes

pinMode(b1\_pin, INPUT\_PULLUP);

pinMode(b2\_pin, INPUT\_PULLUP);

pinMode(led\_pin, OUTPUT);

// start serial communication

Serial.begin(9600,SERIAL\_8N1);

// wait for byte

establishContact();

}

void loop()

{

// Get sensors and buttons values

sensValue = analogRead(sens\_pin);

bt1Value = digitalRead(b1\_pin);

bt2Value = digitalRead(b2\_pin);

// send array to serial port

Serial.write(START\_BYTE);

Serial.print(DELIMITER);

Serial.print(sensValue);

Serial.print(DELIMITER);

Serial.print(bt1Value);

Serial.print(DELIMITER);

Serial.print(bt2Value);

Serial.print(DELIMITER);

Serial.write(END\_BYTE);

Serial.println();

}

**Processing Code**

import processing.serial.\*;

int[] sensors = null;

int sens1 = 0;

int bt1 = 1;

int bt2 = 1;

boolean firstContact = false;

Serial usbPort;

String usbString;

String PortName;

void setup ()

{

size(700, 500);

PortName = Serial.list()[0];

usbPort = new Serial (this, PortName, 9600);

usbPort.bufferUntil ('\n');

}

void draw ()

{

background(200);

dotDrawer(sensor1);

}

void dotDrawer(int slider1)

{

int slider1\_Scaled = int (map (slider1, 0, 1023, 0, width)) ;

point(slider1\_Scaled, 350);

ellipse(slider1\_Scaled, 350, 100, 100);

}

void serialEvent (Serial usbPort)

{

usbString = usbPort.readStringUntil ('\n');

if (usbString != null)

{

usbString = trim(usbString);

if (firstContact == false)

{

if (usbString.equals("Hello"))

{

usbPort.clear();

firstContact = true;

usbPort.write('A');

println("contact");

}

}

else

{

int sensors[ ] = int(split(usbString, ','));

sensor1 = sensors[1];

button1 = sensors[2];

button2 = sensors[3];

} // end else

} // end if

} // end serialEvent