How Processing reads the Arduino data

@Written by Mahshid mzeinaly@sfu.ca

From: http://processing.org/reference/libraries/serial/index.html

Assume that Arduino prints one integer per line.

```
//****Arduino:

int a=134; //example value
Serial.println(a);
```

To read this information in Processing, we could use: port.read();. However, this function reads one character at a time. So it reads our value as '1' '3' '4'. And also because in Arduino we wrote println, there is an "end of line" character at the end of the data.

To test this, you could run the following code and see what it prints:

```
// ***** Arduino
(put this code in “setup” because we want to test one example of data.)

int a=analogRead(ledPin);
Serial.println(b);

// ***** Processing:

if(port.available() >0){
int a =port.read();
println(a);
}
```

For our example it will print :
49
51
52
13
10

If you go to the ASCII table:
http://www.asciitable.com/

49 == '1'
51 == '3'

52 == '4'
13 == end of a data
10 == end of a line (println produces this one)

Now that we know what format it is that Arduino sent and Processing port.read() reads, we can extract the information that we want to use. Below is a sample code for that purpose.

```
//****Processing:

int newData=0; //define in global scope before setup;
if(port.available() >0){
int a=port.read();
if (a !=13 && a!=10){
newData=newData*10+(a-'0');
}
else if (a!=10){
value=newData; //value is ready to use if fill()
newData=0;
}
}
```

* We also could use another function of Processing to read the information that Arduino sent. Since we know each line contains one integer:

```
//**** Processing:

String st = port.readStringUntil('\n'); //read each line
if (st != null){
st = trim(st);
value = int(st); //value is ready to use!
}

//An alternative solution:

String st = port.readStringUntil('\n'); //read each line
if (st != null){
st = trim(st);
value = Integer.parseInt(st); //value is ready to use!
}
```