C0ding Workshop Wk04: Yuta Nakayama

Week03:Assignment1
Rolling Ball machine

Design a rolling ball machine

- ***Size within w:42cm h: 30cm d:30cm (A4 paper)***
- **A mechanism to lift up balls**
- **Minimum 3 Gimmicks using switches or sensors**
  http://www.oobject.com/category/15-videos-of-amazing-rolling-ball-machines/
-

Particles(daito manabe/motoi ishibashi 2011):

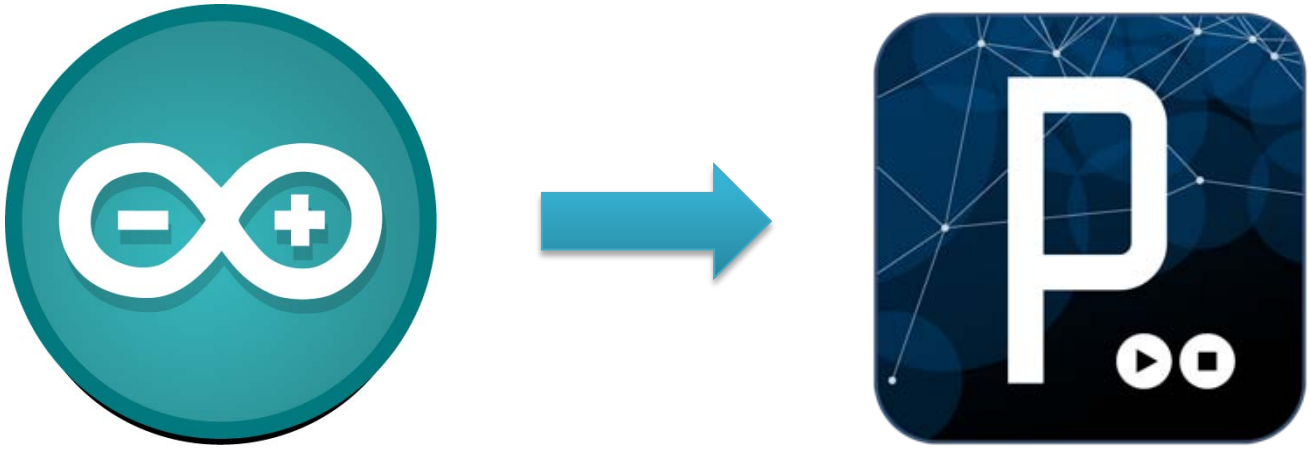
http://www.youtube.com/watch?v=xE8uy_L9dLw

Quick Overview:Week04

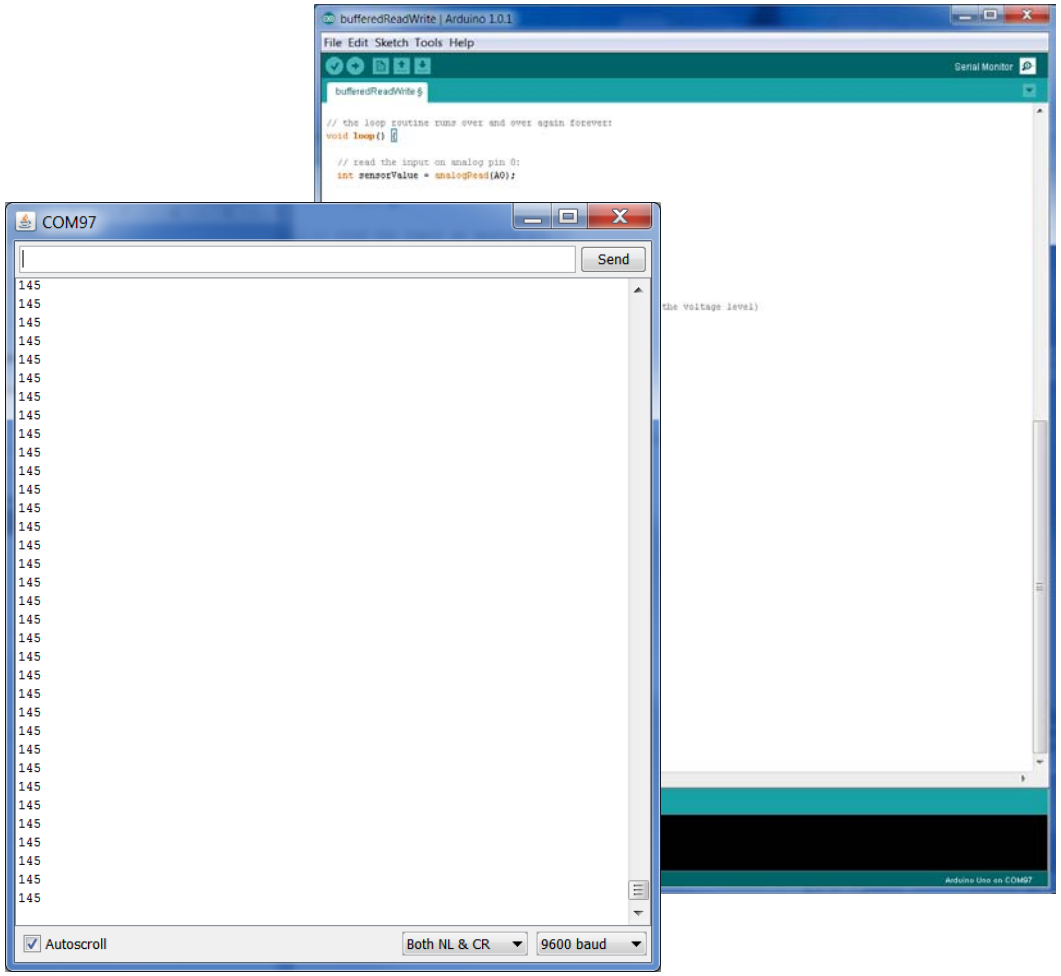
Introduction to Communication/Network

- ***4-1 Arduino <–> Processing***
  - ***Serial Communication***
- ***4-2 Arduino <–> Arduino***
  - ***InfraRed Communication***
- **Circuit Bending Exercise (2hr)**

4-1 Arduino -> Processing

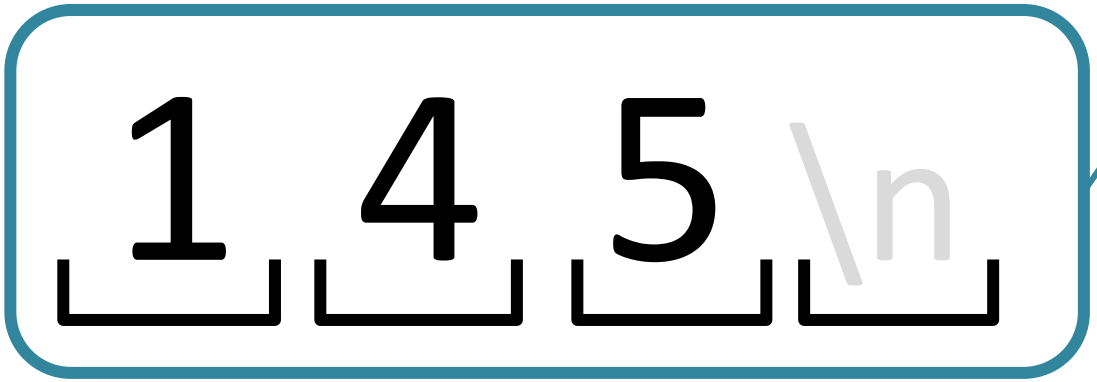


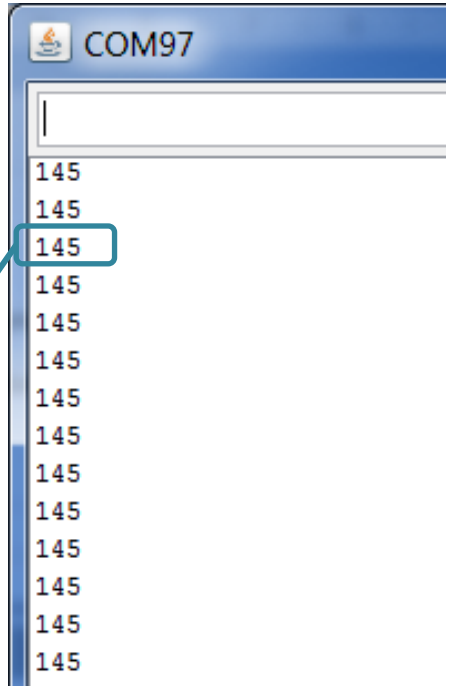
Serial Monitor



Serial Communication

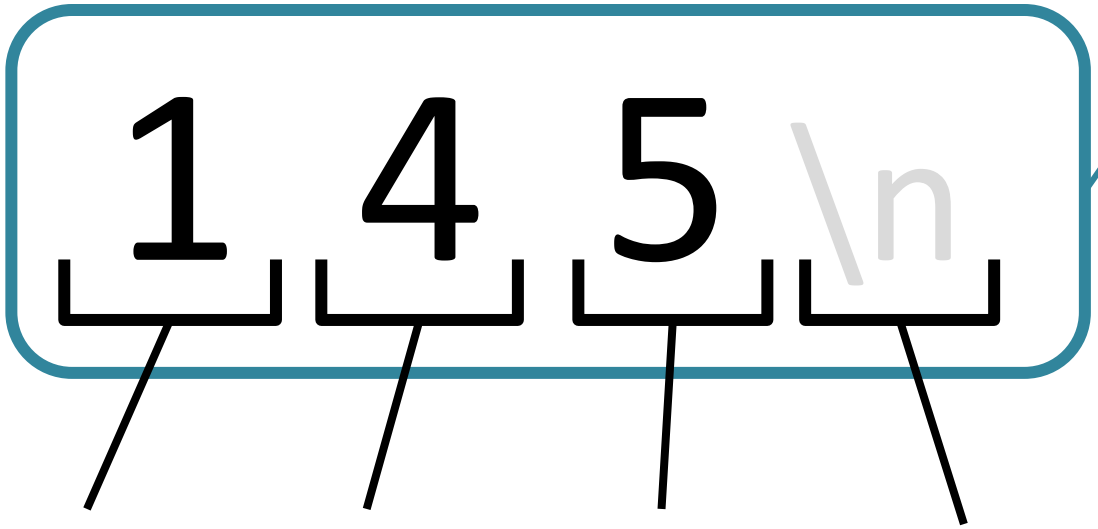
Sending /Receiving value by sequences of byte(0-255) numbers.

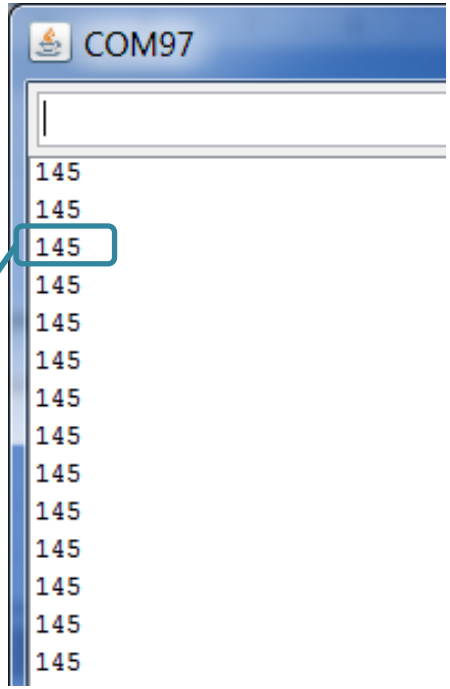




Serial Communication

Sending/Receiving value by sequences of byte(0-255) numbers.





‘1’ = 49 ‘4’ = 52 ‘5’ = 53 ‘\n’ = 10 (line feed)

ASCII Characters

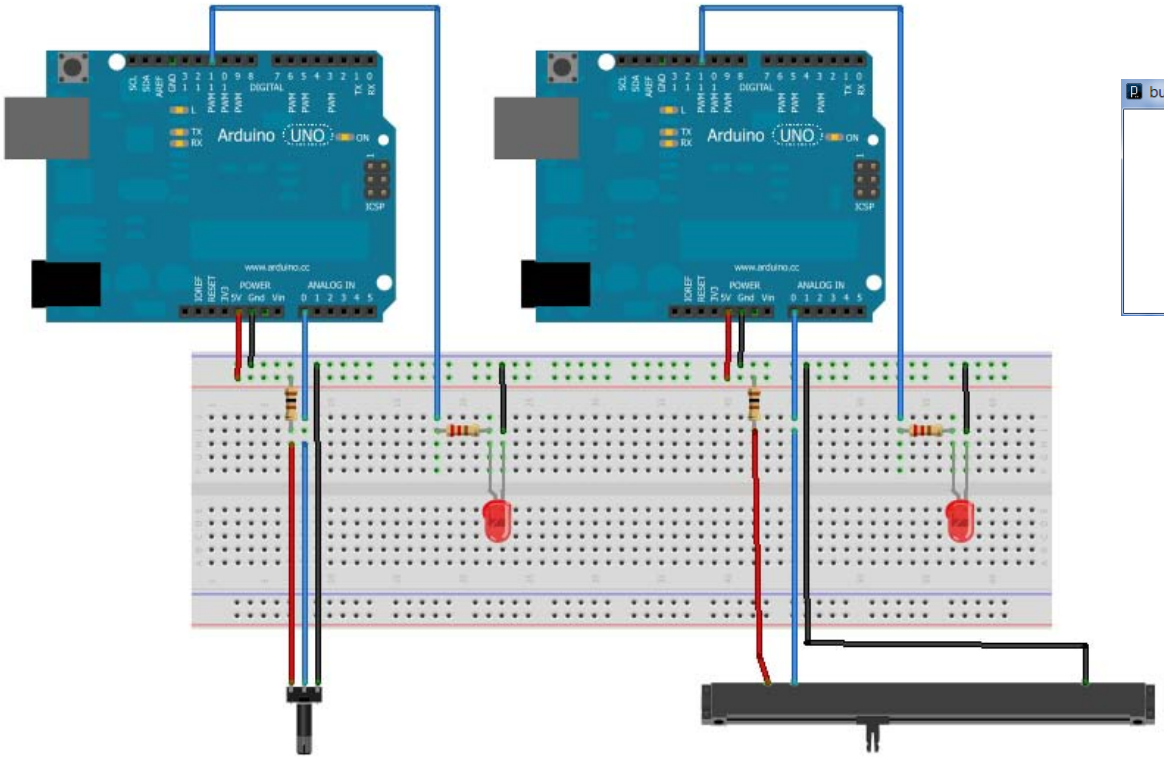
http://arduino.cc/en/Reference/ASCIIchart

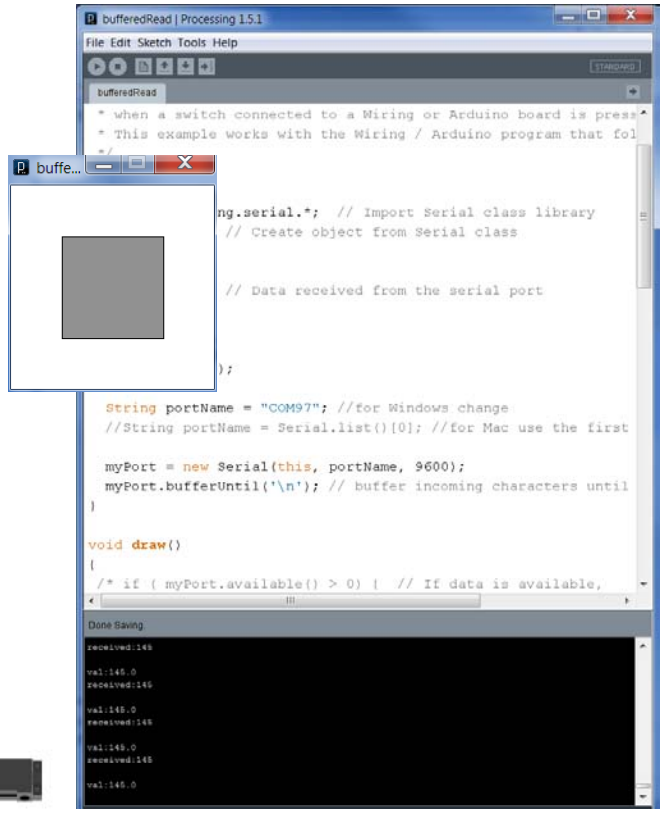
byte val	:	(type of characters)
[0 ~ 32]	:	Special Characters (‘\n’, ’\r’, ‘ESC’)
[48 ~ 59]	:	‘0’, ‘1’, ‘2’ …. ‘9’ (digits)
[65 ~ 90]	:	‘A’, ’B’, ‘C’….’Z’ (upper-case alphabet)
[97 ~ 122]	:	‘a’, ‘b’, ‘c’….’z’ (lower-case alphabet)

Ex 01: 00_Arduino->Processing

/ 01_simpleReadWrite //sending value using Serial.println();

/ 02_bufferdReadWrite //buffered reading example





Arduino Serial Functions (send)

```
Serial.begin(speed);  //start serial communication.

Serial.write(value);  //send RAW byte value(0-255).
      (value = 145 -> 145).

Serial.print(value);
    // send value as a sequence of characters
      (value = 145 -> '1' '4' '5' -> 49 52 53 ).

Serial.println(value);
    // send value as a sequence of characters with line feed.
      (value = 145 -> '1' '4' '5' '\r' '\n' ).
```

Processing Serial Commands(send)

```
import processing.serial.*;  // Import Serial class library
Serial myPort;               // Create object from Serial class

myPort = new Serial(this, portName, speed);
                             //start serial communication.

myPort.available(); // check how many characters have received in the
    buffer

byte value = myPort.read(); // read one byte character. (‘1’ -> 53)

String value = myPort.readString();
    // read incoming byte sequence as a  string (‘1’ ‘4’ ‘5’ -> “145”).
```

Read SensorValue(0-1023)

map in to a byte range(0-255)

Serial.println(val);

"145\n"

mySerial.available() > 0

s = mySerial.readStringUntil("\n") ;

String s= "145\n"

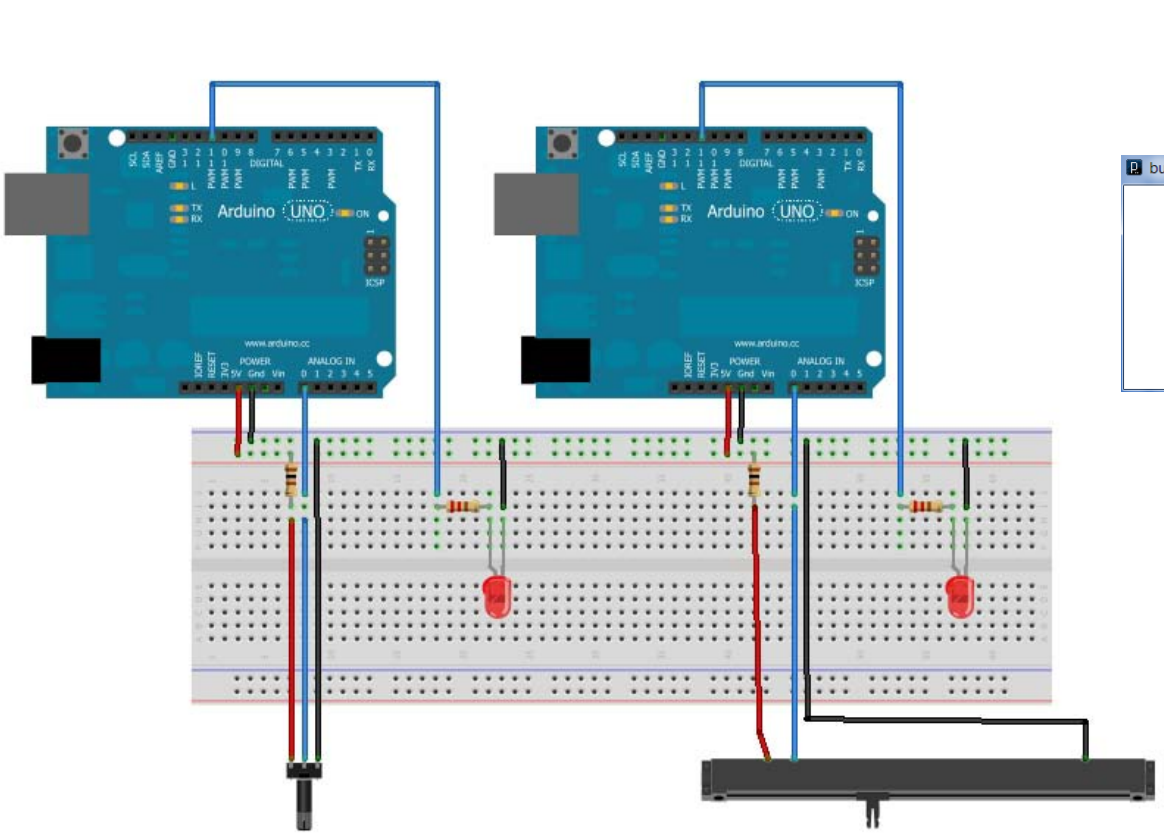
v = float(s);
Convert String to a float value

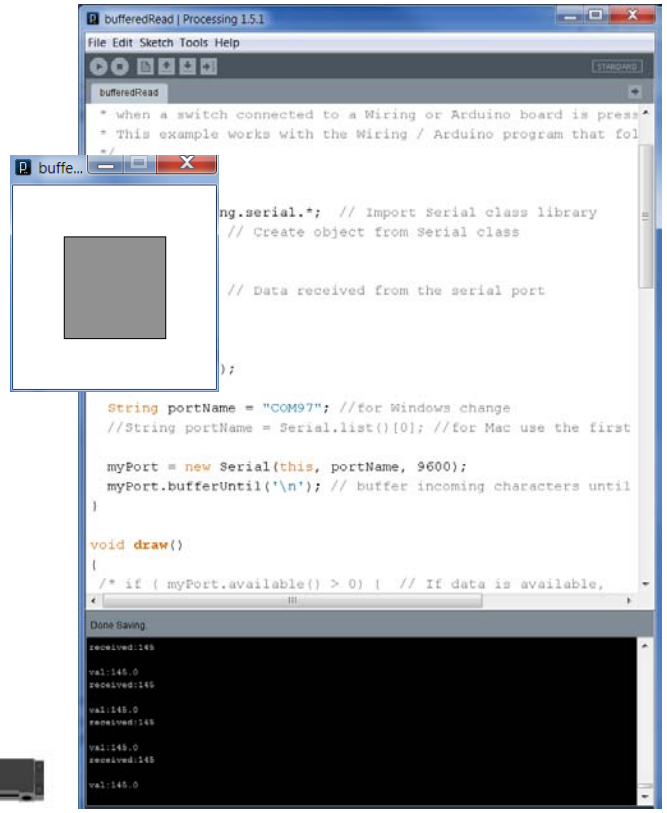
float v= 145;

fill(v);
change color for filling

Ex 02: 01_Arduino<-Processing

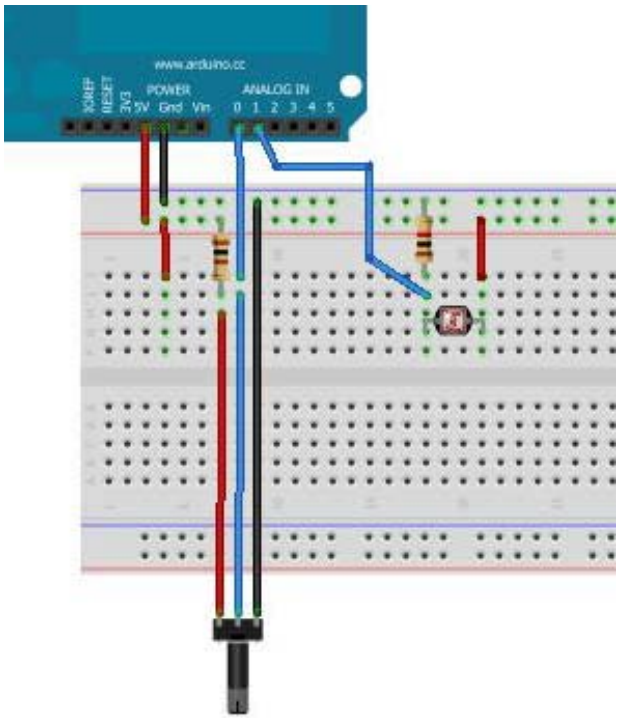
/ 03_simpleWrite





Sending Multiple Values

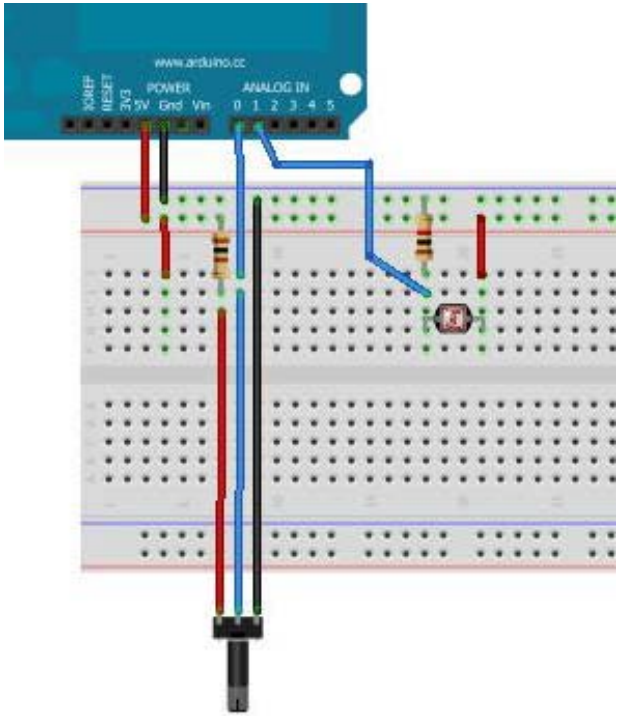
- What if we have connected
  more than one sensors to Arduino?

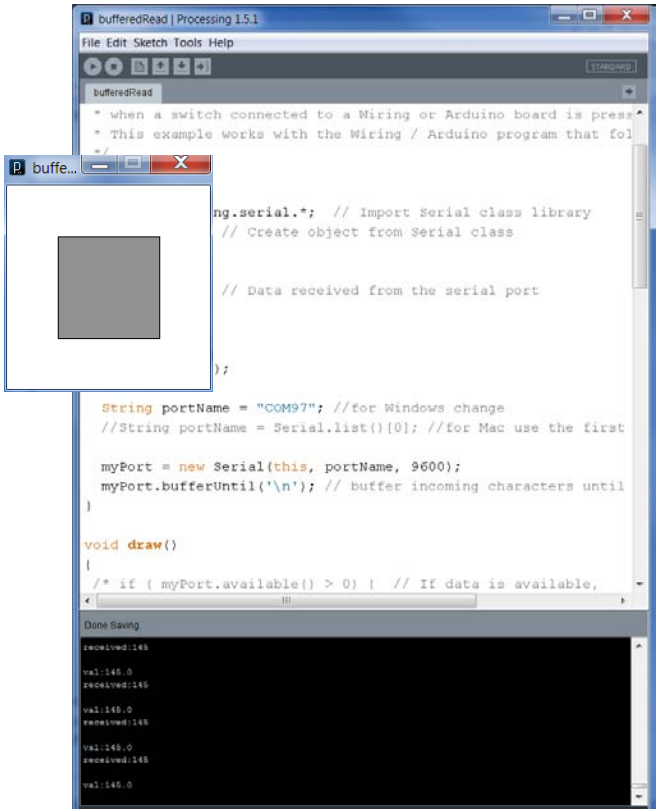


Potentiometer: 300
Light sensor : 200

Ex 03: 00_Arduino->Processing

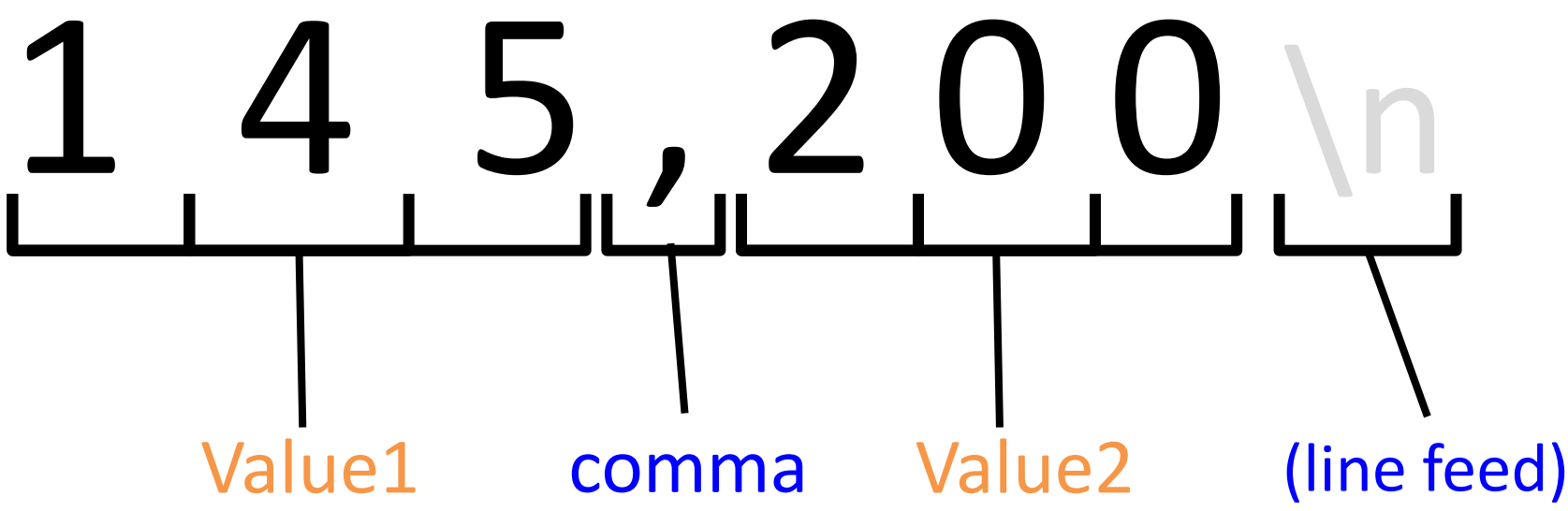
/ 03_multiValueRead //sending multiple values





Comma Separated Value

Sending/Receiving values separated by comma.



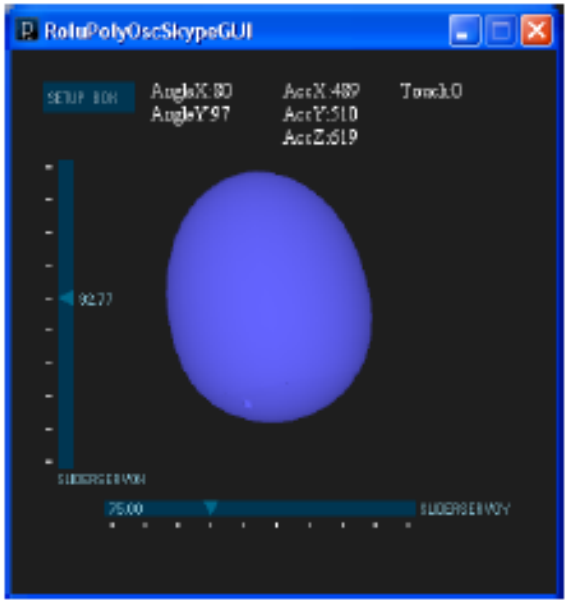
Sending Multiple Values(Arduino)

```
value1 = analogRead(A0);
value2 = analogRead(A1);

 value1 = map(value1,0,1023,0,255);
 value2 = map(value2,0,1023,0,255);

  // print out comma separated values.
  Serial.print(value1); //send value1
  Serial.print(',');        // insert comma.
  Serial.print(value2); //send value2
  Serial.print('\n');       // end with line feed.

  delay(10);              // wait for a while
```

Receiving Multiple Values(multivalueRead.pde)

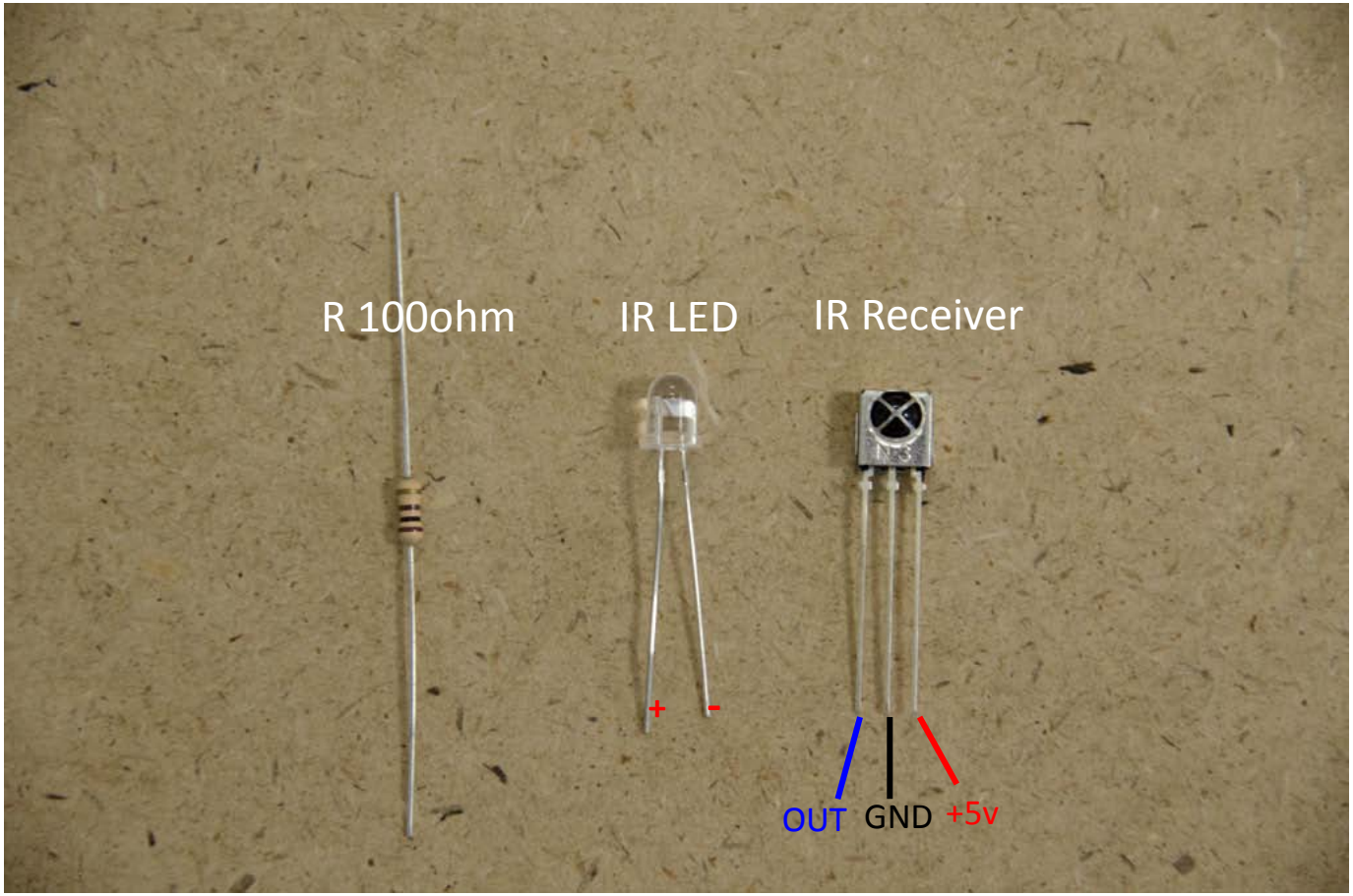
```
//serialEvent function is called when incoming character reaches '\n'
void serialEvent(Serial p){
  String s = "";       // string value received.
  String[] values;     // string array for splitted string values.
  if(p.available() > 0){  // check number of characters in the serial buffer
    s = p.readString();   // read string.
    if(s != null){
      println("received:"+s);
      values = s.split(",");      // separate values with ',' characters.
      println(values);
      val1 = int(values[0]);  // convert string value to int
      val2 = int(values[1]);  // convert string value to int
    }
  }
}
```

Roly Poly


RolyPolyOscSkypeGUI
SETUP BOX
AngleX:80
AngleY:97
AccX:489
AccY:510
AccZ:619
Touch:0
9277
25.00

4-2 Arduino <-> Arduino InfraRed Remote Control

IR Components.



4-2-2 InfraRed Remote Control

Arduino IR Remote Library

https://github.com/NUSID-C0DE/Arduino-IRremote

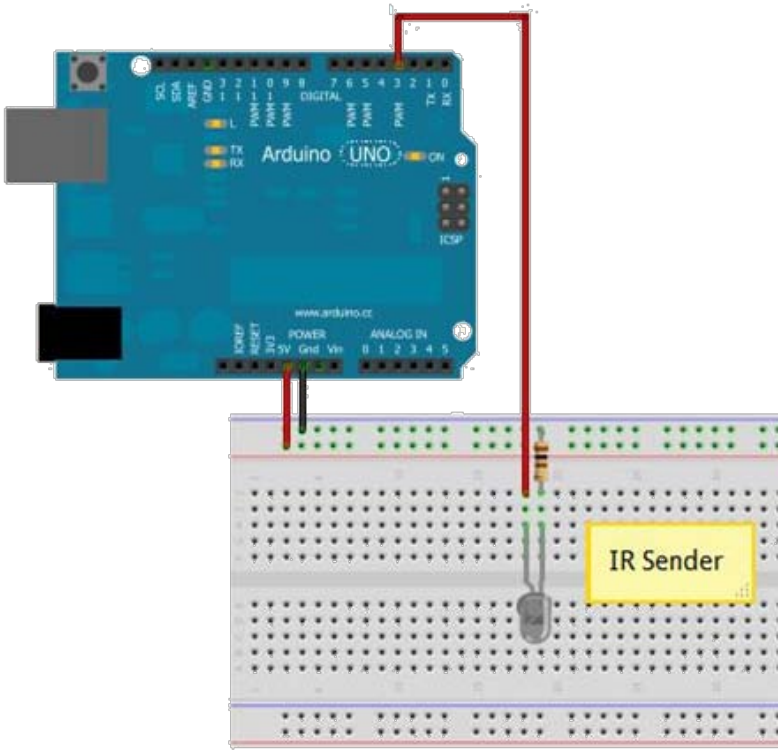
1. Download the library & extract
2. Rename the folder as :”IRremote”
3. Copy the folder into

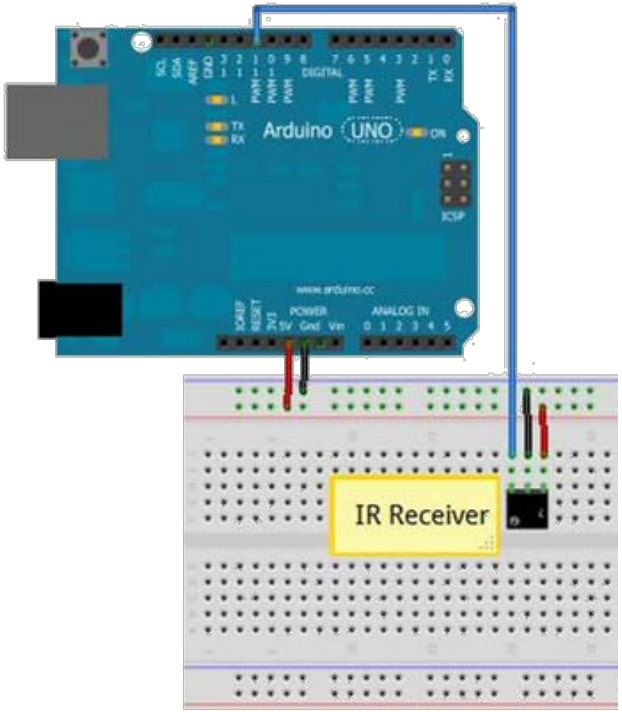
   Windows: Document\Arduino\libraries

   Mac OSX : Documents\Arduino\libraries
4. Restart Arduino and check “IRremote” appeared under File->Examples

Ex 03 :Simple IR

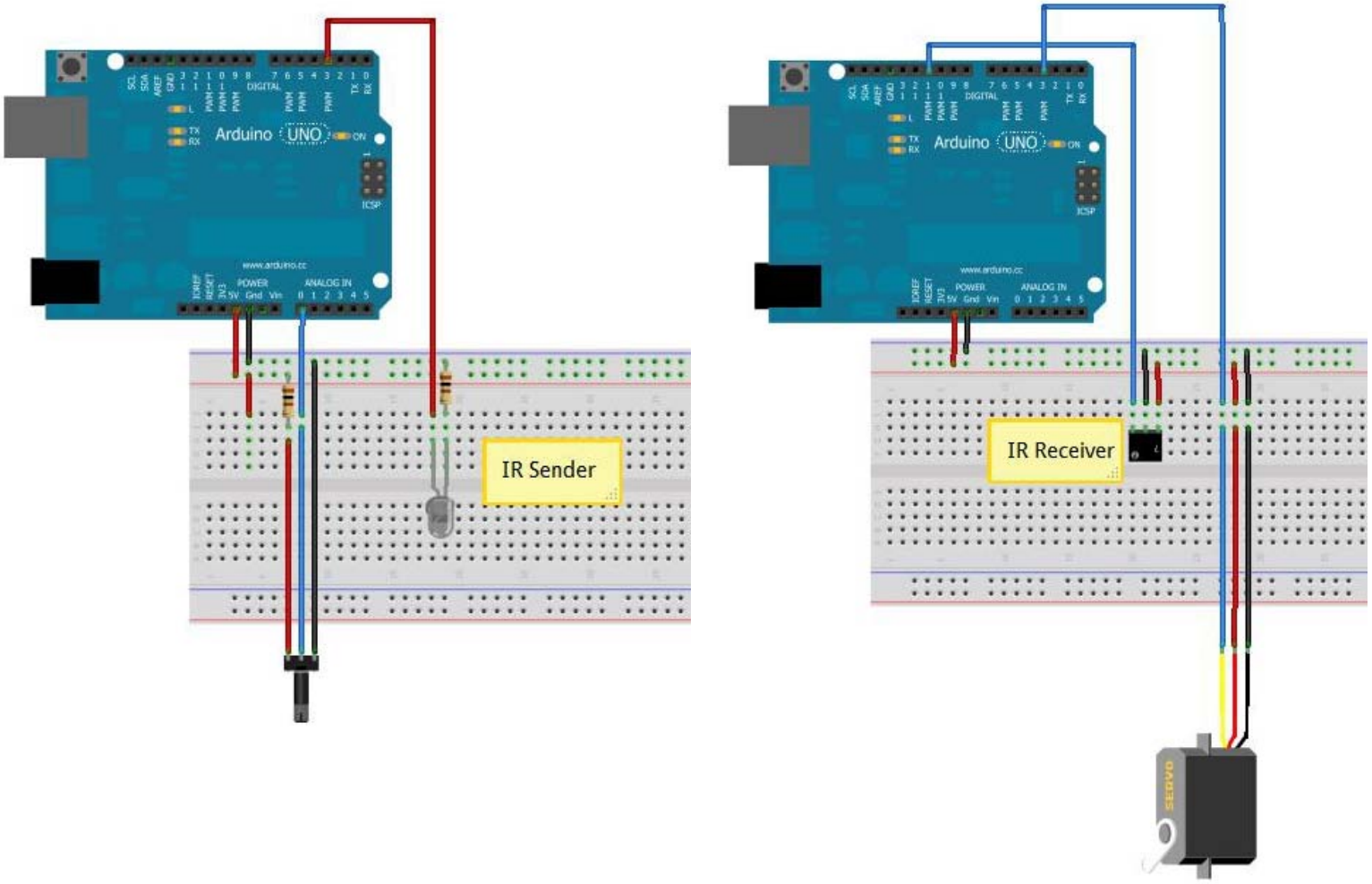
02_Arduino-Arduino\2InfraRed_Remote\01_simpleSendRecv





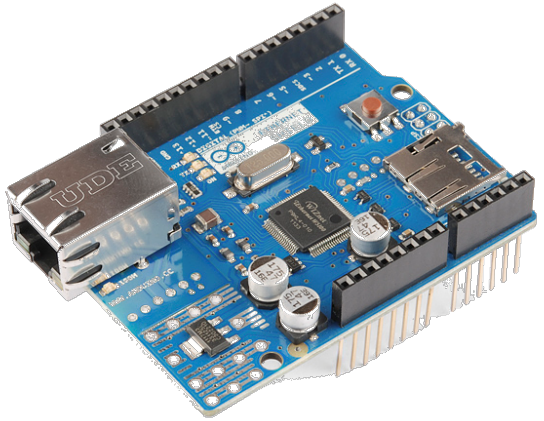
Ex04 : IR + VR + Servo

02_Arduino-Arduino\2InfraRed_Remote\02_VR_ServoControl

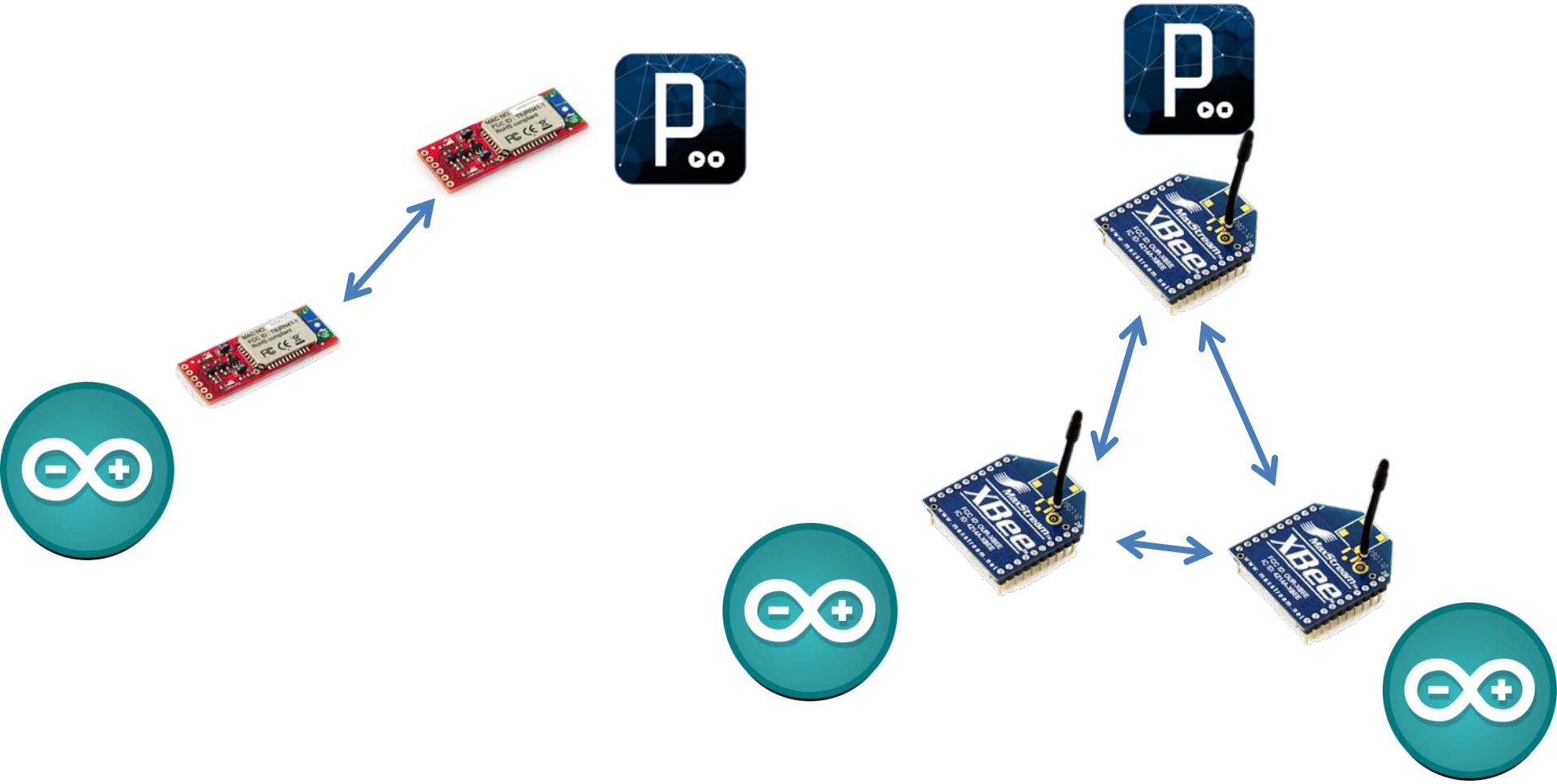


4-2-3 Other Network Devices

- Bluetooth
- XBee
- WiFi
- Ethernet Shield



4-2-3 Other Network Devices



Circuit Bending Exercise

1. Identify the operating voltage of the device. (battery/ AC adaptor)
   - Find out Power/GND lines, how they are routed.
   - **Solder+Extend them into the power rows of BreadBoard.**
2. List up INPUTs/OUTPUTs and their functions of the device.
   - INPUT: Buttons, Switches, microphone etc…
   - OUTPUT: Displays, LED, Speaker, motor, etc…
   - Function: alarm, warning, trigger gun shot etc…
3. Solder + Extend INPUT/OUTPUT signals to the bread board.
4. Try if those INPUT/OUTPUT can be triggered from other signals (POWER/GND etc…) .
5. Check if the machine can be operatable by arduino 5v power out.

   IF yes -> use arduino input/output to control those signal.

   IF no -> use “FET module” to control
   (avoid plugging into arduino input.)

DON’T BE AFRAID OF BRAKING THE DEVICE!