Serial Communication

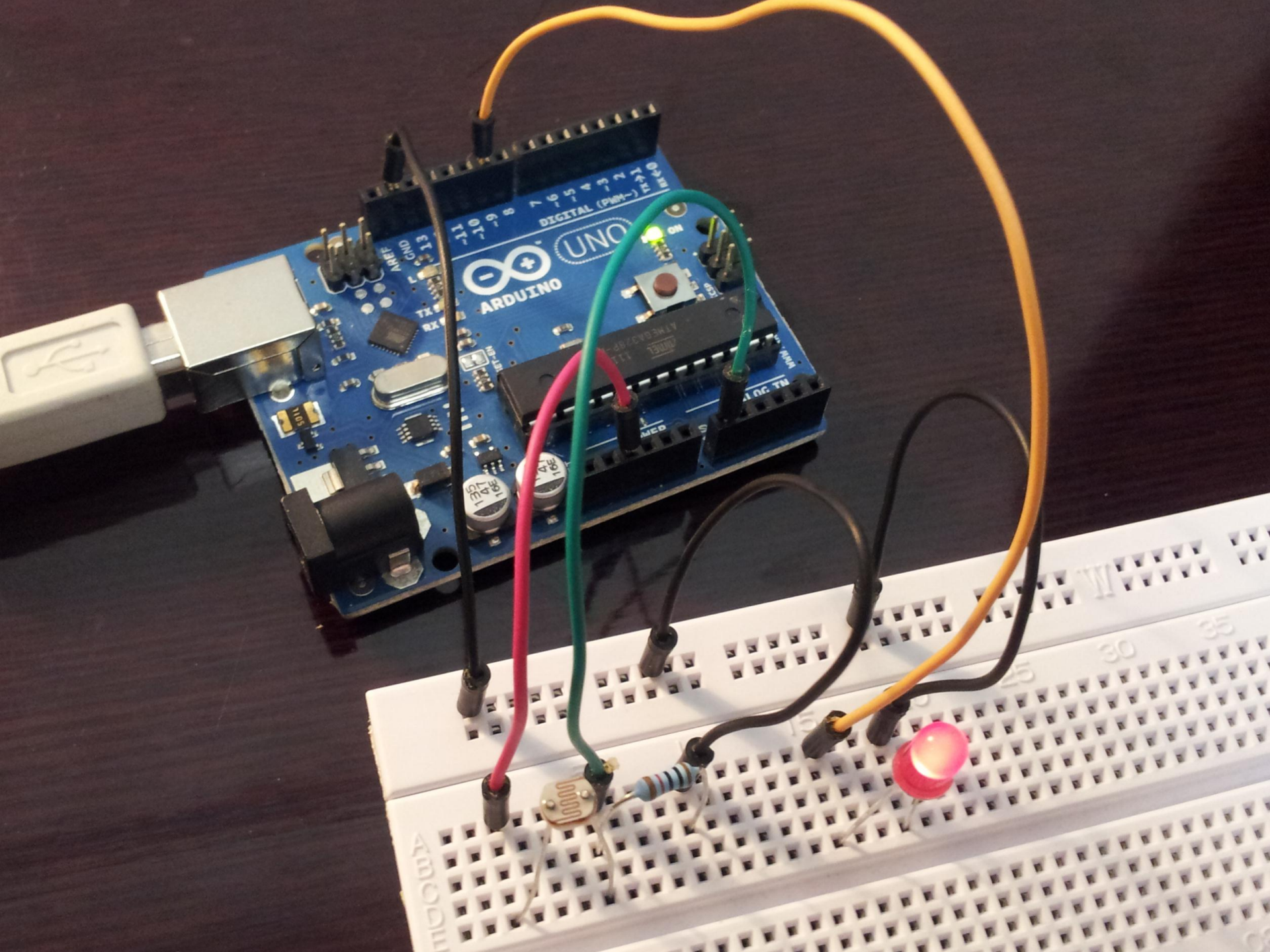
j.hu@tue.nl

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Analog Input Calibration

- **File>Examples>Analog>Calibration**
  - **Analog sensor (Light sensor will do) attached to analog input 0**
  - **LED attached from digital pin 9 to ground**


ARDUINO
UNO
DIGITAL (PWM~)
ON

Analog Input Calibration

```
// These constants won't change:
const int sensorPin = A0;    // pin that the sensor is attached to
const int ledPin = 9;        // pin that the LED is attached to

// variables:
int sensorValue = 0;         // the sensor value
int sensorMin = 1023;        // minimum sensor value
int sensorMax = 0;           // maximum sensor value
```

Analog Input Calibration

```
void setup() {
  // turn on LED to signal the start of the calibration period:
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);

  // calibrate during the first five seconds
  while (millis() < 5000) {
    sensorValue = analogRead(sensorPin);

    // record the maximum sensor value
    if (sensorValue > sensorMax) {
      sensorMax = sensorValue;
    }

    // record the minimum sensor value
    if (sensorValue < sensorMin) {
      sensorMin = sensorValue;
    }
  }

  // signal the end of the calibration period
  digitalWrite(13, LOW);
}
```

Analog Input Calibration

```
void loop() {
  // read the sensor:
  sensorValue = analogRead(sensorPin);

  // apply the calibration to the sensor reading
  sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);

  // in case the sensor value is outside the range seen during calibration
  sensorValue = constrain(sensorValue, 0, 255);

  // fade the LED using the calibrated value:
  analogWrite(ledPin, sensorValue);
}
```

Serial Library

- **We use the same example for Calibration**

Serial Library

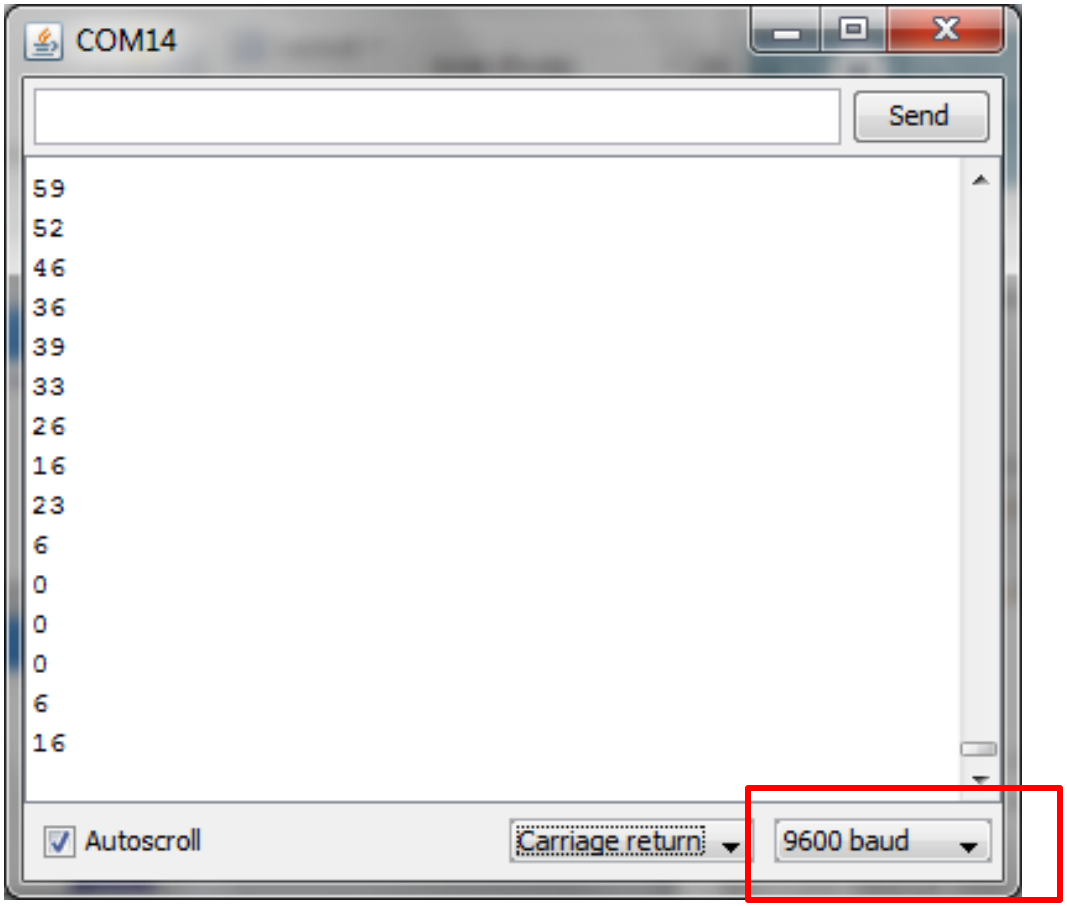
```
void setup() {
  // turn on LED to signal the start of the calibration period:

  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);

  // calibrate during the first five seconds
  while (millis() < 5000) {
    sensorValue = analogRead(sensorPin);

    // record the maximum sensor value
    if (sensorValue > sensorMax) {
      sensorMax = sensorValue;
    }

    // record the minimum sensor value
    if (sensorValue < sensorMin) {
      sensorMin = sensorValue;
    }
  }

  // signal the end of the calibration period
  digitalWrite(13, LOW);

  Serial.begin(9600);
}
```

Serial Library

```
void loop() {
  // read the sensor:
  sensorValue = analogRead(sensorPin);

  // apply the calibration to the sensor reading
  sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);

  // in case the sensor value is outside the range seen during calibration
  sensorValue = constrain(sensorValue, 0, 255);

  // fade the LED using the calibrated value:
  analogWrite(ledPin, sensorValue);

  Serial.println(sensorValue);
  delay(100);
}
```

Serial Library

- **Try it out.**



Serial Library

- **Now change it a bit. Try again the Serial Monitor**

```
void loop() {
  // read the sensor:
  sensorValue = analogRead(sensorPin);

  // apply the calibration to the sensor reading
  sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);

  // in case the sensor value is outside the range seen during calibration
  sensorValue = constrain(sensorValue, 0, 255);

  // fade the LED using the calibrated value:
  analogWrite(ledPin, sensorValue);

  Serial.write(sensorValue);
  delay(100);
}
```

Serial Library

- Now try to receive the sensor input from Processing
- In Processing
  - File>Examples>Books>Get Started>
  - Chapter 11>Ex_11_07

Serial Library

```
import processing.serial.*;

Serial port;  // Create object from Serial class
float val;    // Data received from the serial port

void setup() {
  size(440, 220);
  // IMPORTANT NOTE:
  // The first serial port retrieved by Serial.list()
  // should be your Arduino. If not, uncomment the next
  // line by deleting the // before it. Run the sketch
  // again to see a list of serial ports. Then, change
  // the 0 in between [ and ] to the number of the port
  // that your Arduino is connected to.
  println(Serial.list());
  String arduinoPort = Serial.list()[1];
  port = new Serial(this, arduinoPort, 9600);
}
```

Serial Library

```
void draw() {
  background(255);
  if (port.available() > 0) { // If data is available,
    val = port.read();         // read it and store it in val
    val = map(val, 0, 255, 0, height);  // Convert the value
  }
  rect(40, val-10, 360, 20);
}
```

Serial Library

- Now use the same hardware, try out
- In Processing
  - File>Examples>Books>Get Started>Chapter 11>Ex_11_08

Firmata

Firmata

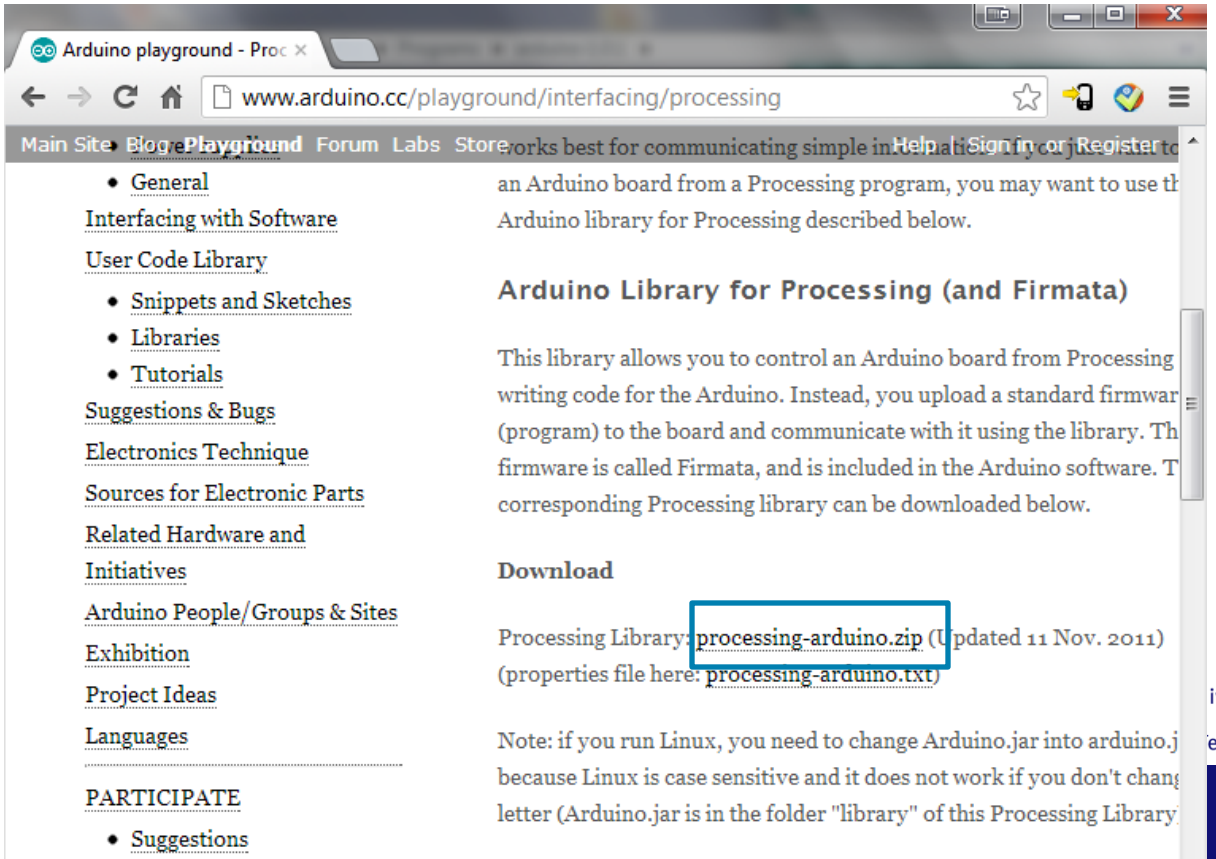
- **A generic protocol for communicating with Microcontrollers like the Arduino from software on a host computer.**
- **See also**
  - **http://firmata.org**
  - **http://www.arduino.cc/en/Reference/Firmata**
  - **http://www.arduino.cc/playground/interfacing/processing**

Firmata

- **Arduino Library for Processing (and Firmata)**
  - This library allows you to control an Arduino board from Processing *without writing code for the Arduino*. Instead, you upload a *standard firmware* (program) to the board and communicate with it using the library.

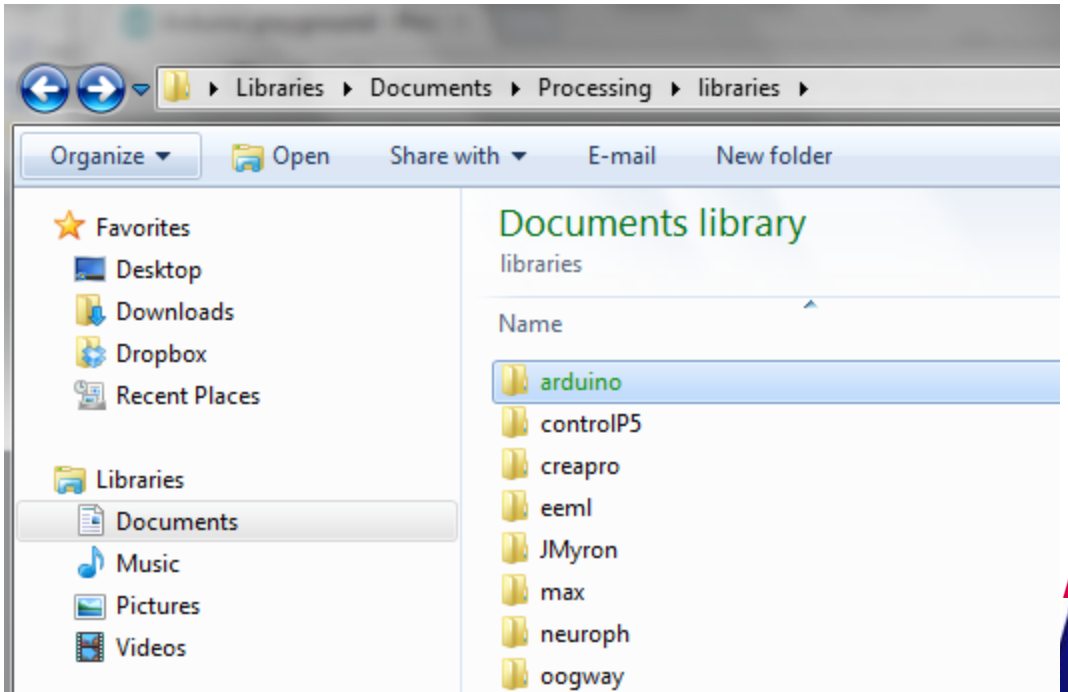
Firmata

- **Install Arduino.jar for Processing**
  - http://www.arduino.cc/playground/interfacing/processing
  - **Download** processing-arduino.zip



Firmata

- **Install Arduino.jar for Processing**
  - **http://www.arduino.cc/playground/interfacing/processing**
  - **Download processing-arduino.zip**
  - **Unzip the library and copy the "arduino" folder into the "libraries" sub-folder of your Processing Sketchbook.**



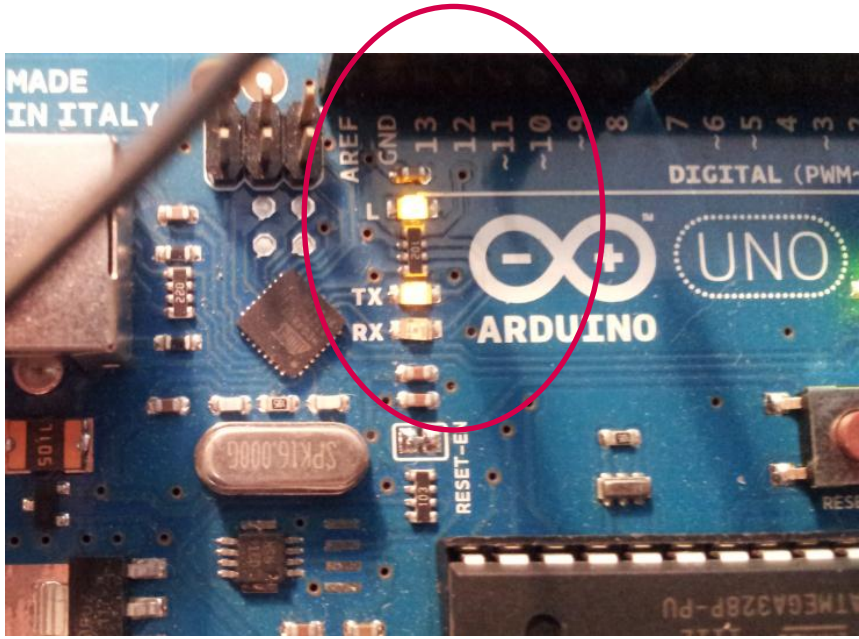
Firmata

- **Load the standard firmware to Arduino**
  - **Run Arduino, open**
    - **Examples > Firmata > StandardFirmata,**
  - **and upload it to the Arduino board.**

Firmata

- **In Processing, try the example from**
  - **http://www.arduino.cc/playground/interfacing/processing**

```
import processing.serial.*;
import cc.arduino.*;

Arduino arduino;
int ledPin = 13;

void setup()
{
  //println(Arduino.list());
  arduino = new Arduino(this, Arduino.list()[1], 57600);
  arduino.pinMode(ledPin, Arduino.OUTPUT);
}

void draw()
{
  arduino.digitalWrite(ledPin, Arduino.HIGH);
  delay(1000);
  arduino.digitalWrite(ledPin, Arduino.LOW);
  delay(1000);
}
```



Firmata

- **Arduino.list()**: returns a list of the available serial devices. If your Arduino board is connected to the computer when you call this function, its device will be in the list.
- **Arduino(parent, name, rate)**: create an Arduino object. Parent should be "this" (without the quotes); name is the name of the serial device (i.e. one of the names returned by Arduino.list()); rate is the speed of the connection (115200 for the v2 version of the firmware, 57600 for v1). Note that in the v2 library, the rate parameter is optional.
- **pinMode(pin, mode)**: set a digital pin to input or output mode (Arduino.INPUT or Arduino.OUTPUT).
- **digitalRead(pin)**: returns the value of a digital pin, either Arduino.LOW or Arduino.HIGH (the pin must be set as an input).
- **digitalWrite(pin, value)**: writes Arduino.LOW or Arduino.HIGH to a digital pin.
- **analogRead(pin)**: returns the value of an analog input (from 0 to 1023).
- **analogWrite(pin, value)**: writes an analog value (PWM wave) to a digital pin that supports it (pins 3, 5, 6, 9, 10, and 11); value should be from 0 (always off) to 255 (always on).

Firmata

- **Input example. In Processing**
  - **Examples>Contributed Libraries>arduino>arduino_input**

Firmata

- **Input example.**

```
import processing.serial.*;

import cc.arduino.*;

Arduino arduino;

color off = color(4, 79, 111);
color on = color(84, 145, 158);

void setup() {
  size(470, 280);
  arduino = new Arduino(this, Arduino.list()[1], 57600);

  for (int i = 0; i <= 13; i++)
    arduino.pinMode(i, Arduino.INPUT);
}
```

Firmata

- **Input example.**

```
void draw() {
  background(off);
  stroke(on);

  for (int i = 0; i <= 13; i++) {
    if (arduino.digitalRead(i) == Arduino.HIGH)
      fill(on);
    else
      fill(off);

    rect(420 - i * 30, 30, 20, 20);
  }

  for (int i = 0; i <= 5; i++) {
    ellipse(280 + i * 30, 240, arduino.analogRead(i) / 16, arduino.analogRead(i) / 16);
  }
}
```

Firmata

- **Output example. In Processing**
  - **Examples>Contributed Libraries>arduino>arduino_output**

Firmata

- **Output example.**

```
import processing.serial.*;

import cc.arduino.*;

Arduino arduino;

color off = color(4, 79, 111);
color on = color(84, 145, 158);

int[] values = { Arduino.LOW, Arduino.LOW, Arduino.LOW, Arduino.LOW,
 Arduino.LOW, Arduino.LOW, Arduino.LOW, Arduino.LOW, Arduino.LOW,
 Arduino.LOW, Arduino.LOW, Arduino.LOW, Arduino.LOW, Arduino.LOW };

void setup() {
  size(470, 200);

  println(Arduino.list());
  arduino = new Arduino(this, Arduino.list()[1], 57600);

  for (int i = 0; i <= 13; i++)
    arduino.pinMode(i, Arduino.OUTPUT);
}
```

Firmata

- **Output example.**

```
void draw() {
  background(off);
  stroke(on);

  for (int i = 0; i <= 13; i++) {
    if (values[i] == Arduino.HIGH)
      fill(on);
    else
      fill(off);

    rect(420 - i * 30, 30, 20, 20);
  }
}
```

Firmata

- **Output example.**

```
void mousePressed()
{
  int pin = (450 - mouseX) / 30;

  if (values[pin] == Arduino.LOW) {
    arduino.digitalWrite(pin, Arduino.HIGH);
    values[pin] = Arduino.HIGH;
  } else {
    arduino.digitalWrite(pin, Arduino.LOW);
    values[pin] = Arduino.LOW;
  }
}
```

Firmata

- **PWM example. In Processing**
  - **Examples>Contributed Libraries>arduino>arduino_pwm**

Firmata

- **PWM example.**

```
import processing.serial.*;

import cc.arduino.*;

Arduino arduino;

void setup() {
  size(512, 200);
  arduino = new Arduino(this, Arduino.list()[1], 57600);
}

void draw() {
  background(constrain(mouseX / 2, 0, 255));
  arduino.analogWrite(9, constrain(mouseX / 2, 0, 255));
  arduino.analogWrite(11, constrain(255 - mouseX / 2, 0, 255));
}
```

Serial Communication

j.hu@tue.nl

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts