AD32600 Physical Computing
Prof. Fabian Winkler
Fall 2014

Arduino Workshop 04 – Arduino ←→ Processing Integration

This workshop is about the integration of Arduino and Processing, building on examples of the previous Arduino and Processing workshops. In the first half of the workshop, we use the Firmata Arduino firmware together with the Arduino library in Processing to remotely control the input and output pins of an Arduino board. Firmata and the Arduino library are both based on the serial communication protocol that we have used so far for Arduino debugging in Arduino's serial monitor and for communication between the Arduino board and the iRobot Create. Then we take a quick look at wireless serial communication with XBee modules and conclude with examples connecting the iRobot Create via Arduino to Processing using our own serial communication protocol.

This workshop is organized into the following sections:

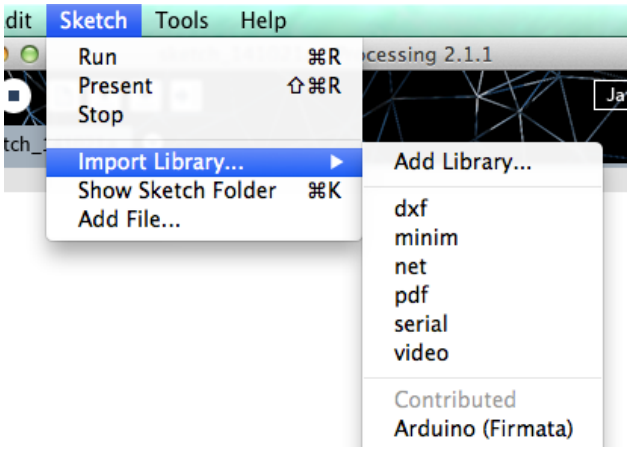
1) Processing → Arduino: RGB color mixer
2) Arduino → Processing: drawing lines based on LDR values
3) XBee introduction (separate workshop document)
4) iRobot → Processing: drawing lines based on position
5) Processing → iRobot: send navigation commands

(1) Processing → Arduino: RGB color mixer

In this example we would like to control the individual red, green and blue components of the RGB LED with three sliders in Processing creating a simple color mixer.
We need to prepare both, the Arduino board and Processing with firmware/libraries though before we can establish a successful communication between the two.

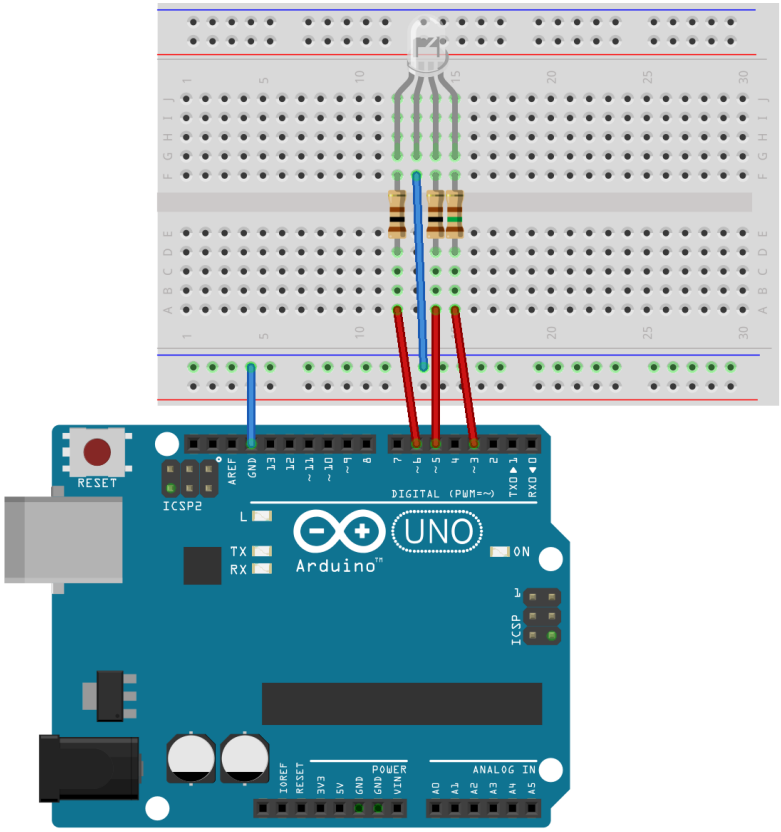
Do the following:

- Upload the Arduino Firmata firmware onto your Arduino board, in the Arduino software go to: `File > Examples > Firmata > Standard Firmata`
  For more information about Firmata see: http://firmata.org
- Download the Arduino (Firmata) library for Processing here: http://playground.arduino.cc/Interfacing/Processing and follow the installation instructions in the document. The next time you open Processing (close it first if you had it open during installation) you should see the Arduino (Firmata) library in the Contributed section of your libraries. In Processing go to `Sketch > Import Library` to see a list of all installed libraries.



- While you are installing libraries in Processing, download and install the controlP5 library as well: http://www.sojamo.de/libraries/controlP5, it will help us building user interfaces, such as sliders for the LED control.

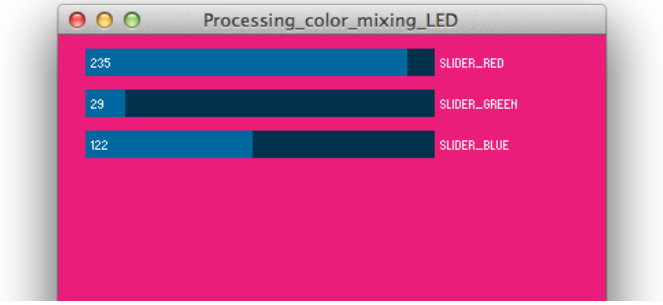
Now build the Arduino RGB LED circuit from the end of Arduino workshop 01:



With our RGB LED (Jameco part no. 2125181) use the following resistors: R1, R2 = 100Ω, R3 = 150Ω or if you don't have these values R1, R2 = 100Ω, R3 = 220Ω.

You do not need to write a program in Arduino, it is enough to upload the firmata firmware. All the Arduino control will be performed from within your Processing sketch.

First however, make sure you understand how sliders work in the Processing controlP5 library:
http://web.ics.purdue.edu/~fwinkler/AD32600_F14/Processing_color_mixing.zip



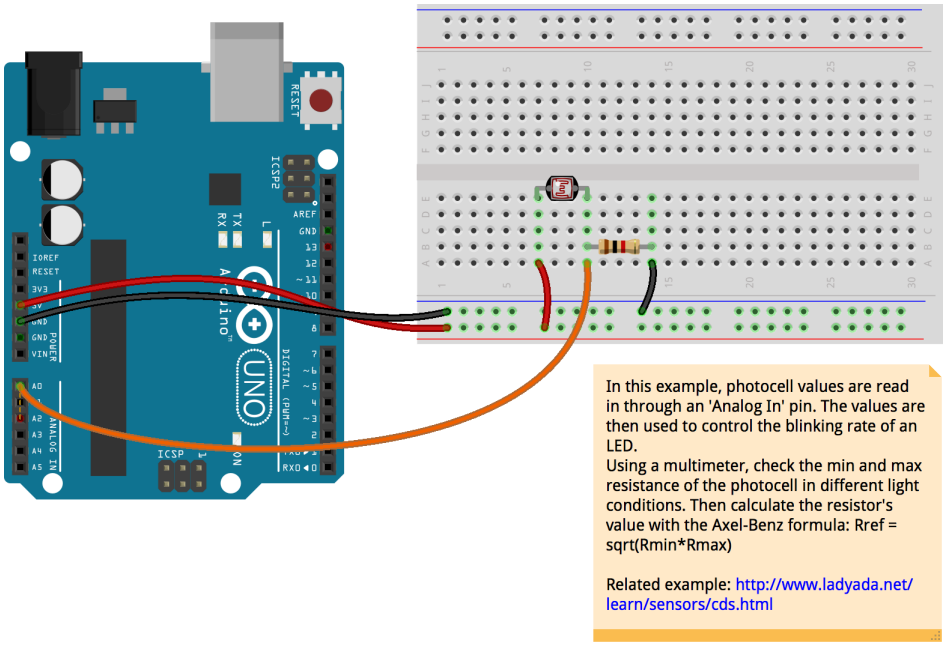
Then add the functionality to send out values from your Processing sketch that control the brightness of each color component in the RGB LED:
http://web.ics.purdue.edu/~fwinkler/AD32600_F14/Processing_color_mixing_LED.zip

If you would like to get some ideas of how to control motors from within Processing take a look at this workshop I prepared a while back:
http://web.ics.purdue.edu/~fwinkler/AD32600_F14/Processing_motor_control.pdf

(2) Arduino → Processing: drawing lines based on LDR

In this example we would like to experiment with Arduino and Processing in the opposite direction: sensor input from the Arduino board will generate visual content in Processing. Specifically, a photocell connected to the analog input pin A0 on the Arduino board will allow us to draw animated circles in Processing. It's quite hypnotic!

Use a modified version of the circuit from Fritzing: `File > Open Example > Arduino > Analog > Input > Photocell`



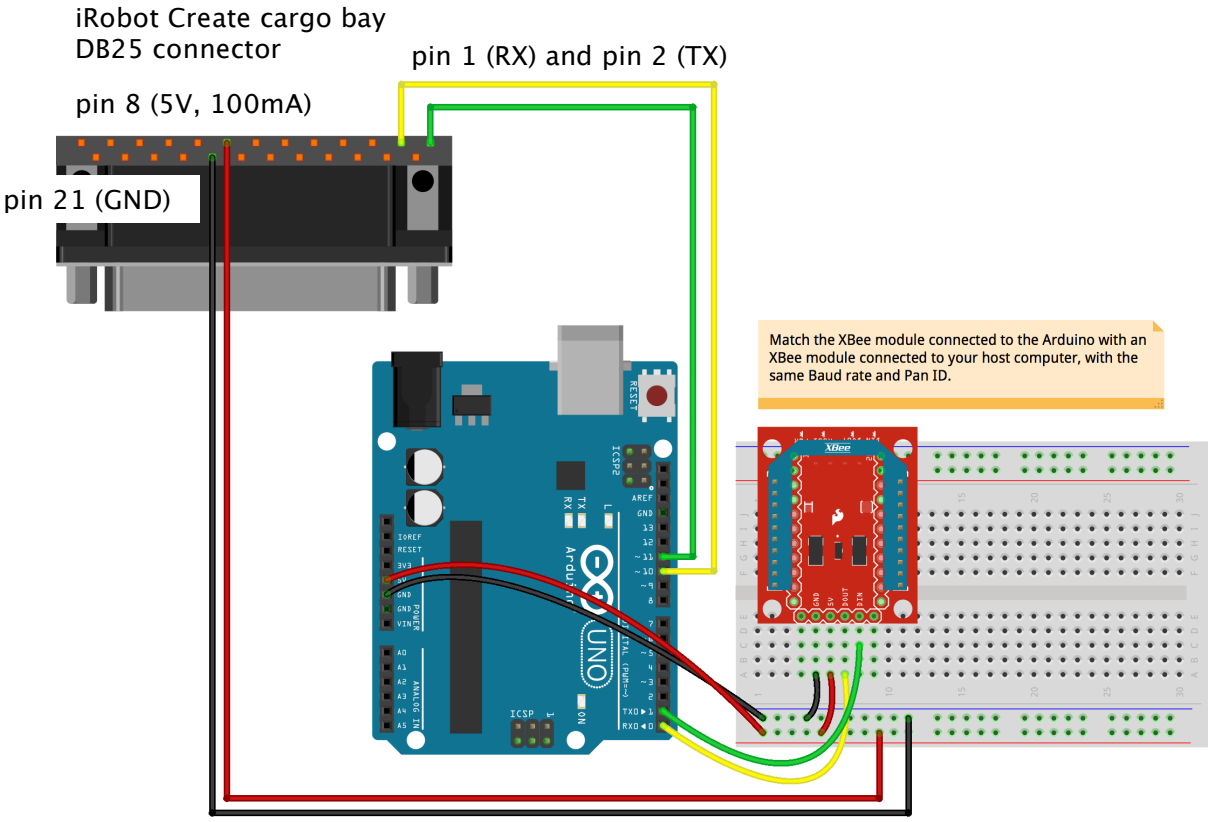
Here is the Processing code that goes with the example (you don't need to upload anything onto the Arduino board as long as it still has the Firmata firmware from the last example):
http://web.ics.purdue.edu/~fwinkler/AD32600_F14/photocell_hypnosis.zip

XBee introduction

XBee technology is very helpful if you would like to control (or receive data from) your Arduino board in Processing wirelessly. I made a separate workshop document if you are interested in how to configure XBee modules so you can use them with Arduino and Processing: http://web.ics.purdue.edu/~fwinkler/AD32600_F14/XBee_Arduino.pdf

iRobot → Processing: drawing lines based on position

If you don't remember how to connect the Arduino board to the iRobot Create, here is the breadboard view of the circuit including the connections for the XBee module:



This example was inspired by Tod E. Kurt's iRobot Create to Processing example in "Hacking Roomba" chapter 7 (pp.131 – 150). It uses the Create's sensor readings for the distance traveled and turn angles directly from the robot's wheels and – using some math – turns them into line drawings in Processing. This technique of determining a robot's position based on its wheel sensors (rotary encoders) is called dead reckoning. It is not perfect and a more complex sensor system such as a scanning laser rangefinder couple with SLAM techniques (Simultaneous Localization and Mapping) would result in a much more precise localization. However this strategy is much more expensive in terms

of your finances and computational calculations. It requires a sensor such as this one: http://www.robotshop.com/en/rplidar-360-laser-scanner.html?gclid=CJD1jfaqwMECFQemaQodJooAiA and a microprocessor such as the Raspberry PI that can run ROS (robot operating system, http://www.ros.org/). This would be something for a more advanced robotics class. We'll keep it simple and cheap for now and stick to Arduino and the iRobot Create's onboard sensors.

- **Arduino code** (takes care of the driving behavior – bump and go and sends sensor values to Processing), unzip and upload this code onto your Arduino board: http://web.ics.purdue.edu/~fwinkler/AD32600_F14/bump_and_run_serial_send_position.zip
- **Processing code** (reads sensor values from Arduino and translates them into a drawing that tracks the robot's position), unzip and run this code in Processing: http://web.ics.purdue.edu/~fwinkler/AD32600_F14/Processing_receive_position_data.zip

I think these two pieces of code can be improved still, but they are a good starting point if you are interested in drawing things in Processing with the iRobot Create. Also take a look at Tod E. Kurt's RoombaComm library (http://hackingroomba.com/code/roombacomm/) this is great if you don't want to use the Arduino and connect an XBee module directly to the iRobot Create DB25 connector. However, eliminating the Arduino will make connecting external sensors (such as photocells, potentiometers, line reading sensors, etc…) more complicated.

Processing → iRobot: send navigation commands

the last example creates an easy programming interface in Processing that lets users send sequences of serial commands to the iRobot Create for execution. With this example, code that makes the robot move, spin and turn can be programmed in Processing and sent to the iRobot Create wirelessly. In this regard, it is very similar to the Tod E. Kurt's RoombaComm Processing library: http://hackingroomba.com/code/roombacomm/

Arduino code (interprets the serial commands from Processing and routes them to iRobot Create), unzip and upload onto the Arduino board: http://web.ics.purdue.edu/~fwinkler/AD32600_F14/Arduino_receive_from_Processing.zip

Processing code (translates simple commands and numerical values for speed, turn angles, etc. into serial commands that the iRobot Create understands), unzip and run in Processing: http://web.ics.purdue.edu/~fwinkler/AD32600_F14/Processing_sending_to_Arduino_iRobot.zip

Again, the code examples could be improved, but they give you a good starting point for more complex Processing to iRobot Create interactions, such as making the robot follow the mouse, develop some drawing algorithms that the robot can execute, etc…