

Lab #1

CET4811 – Spring 2019

Arduino ↔ Webserver/Browser Communication

Introduction

Lab Purpose: Use serial port to connect to visualization tools

- Use Node.js Serial Port Module to make Arduino and Web Server/Browser (Node.js) work together
- This lab was updated from an online tutorial. Check back on this site for more instructions on getting Node.js to communicate with Arduino:
- <http://www.barryvandam.com/node-js-communicating-with-arduino/#comment-4834>
- <https://youtu.be/m-3XvNQko4s>
- Arduino compatible visualization tools
 - Processing – Java
 - Line Charts: https://forum.processing.org/two/discussion/3141#Item_3
 - Node.js – Javascript/JQuery
 - Line Charts: <https://canvasjs.com/html5-javascript-line-chart/>
 - OpenFrameworks – C++
 - <https://openframeworks.cc/documentation/graphics/ofPolyline/>

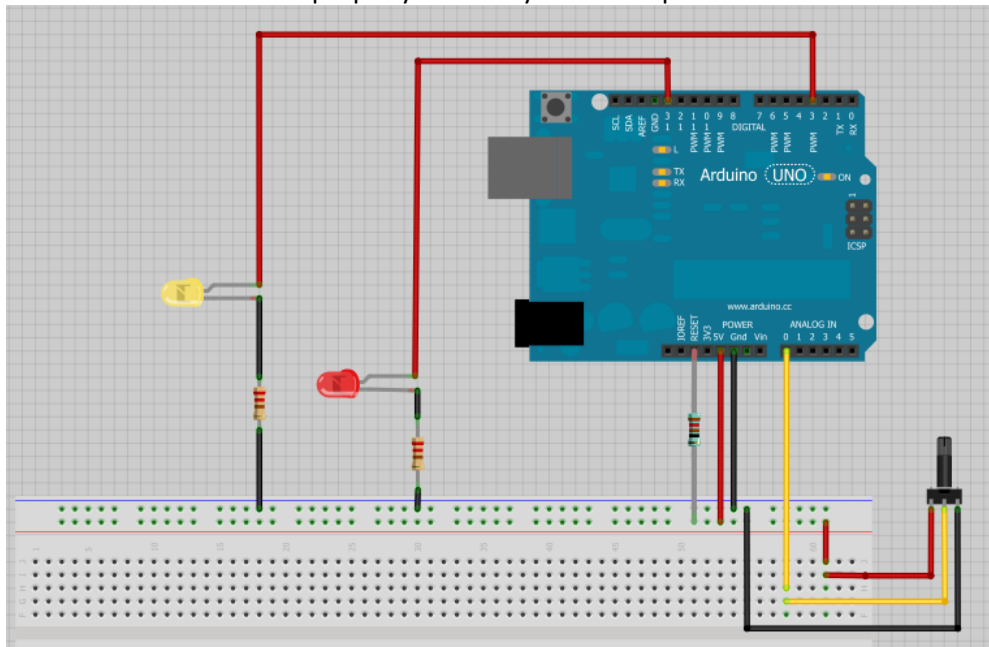
Part I. Installation

- 1) Download the updated software tools from blackboard after reading the setup at the online tutorial.
- 2) Install Node.js & NPM (Node Package Manager)
 - a. see this installation setup for Windows systems: <https://blog.teamtreehouse.com/install-node-js-npm-windows>
 - b. pick the most up-to-date version for your computer at <http://nodejs.org/>
 - c. run the installer and accept the license agreement
 - d. Verify Node.js is installed
 - i. Open a command prompt (type 'cmd' on the search bar for Windows Users)
 - ii. type 'node -v' (no quotes!) => this should give you the Node version number
 - iii. now type 'npm -v' => this should give you the NPM version
 - iv. If both commands are working proceed to the next step
- 3) Add the appropriate modules in the code working directory:
 - a. change directories inside the command prompt:
 - b. for example: 'cd c:\Documents\NewDir\4811\Lab1'
- 4) install the software tools from blackboard into this directory
 - a. this should make another directory SCOM4Class2
- 5) From the command prompt cd into SCOM4Class2 and run the following commands
 - a. 'cd SCOM4Class2'
 - b. 'npm install socket.io'
 - c. 'npm install -g express'
 - d. 'npm install serialport' => this command may give you an error, see Step 6 if needed

- 6) You most likely will need to also install windows build tools to get the serialport module installed properly:
 - a. 'npm install -g --production windows-build-tools'
 - b. then re-run the serialport install command
- 7) Node.js installation is complete!

Part II. Work with Node.js & Arduino

- 8) Important files to understand/edit
 - a. Arduino Sketch file: / _ Arduino/ArduinoNodeJS/ArduinoNodeJS.ino
 - b. Node webserver & SCOM server file: /bin/www
 - c. Main webpage: /public/index.ejs
 - d. Main javascript file to communicate between browser & Arduino via SCOM port: /public/javascripts/arduinoData.js
- 9) Build your Circuit!
 - a. make sure to properly connect your RESET pin!



- 10) Start the Arduino software FIRST!
 - a. run Arduino code in the Arduino directory with your favorite Arduino IDE
 - b. connect your Arduino board to your computer and send the Arduino code to the board
 - c. Arduino has to have write control permission of the SCOM (serial com) port 1ST otherwise it will NOT run simultaneously with Node
- 11) Edit the COM port in your webserver
 - a. edit the /bin/www file
 - b. change the portName variable to the appropriate COM port

```
var portName = 'COM3';
```

- 12) Start the Node webserver
 - a. to run the webserver that will load the browser page run this command from your code working directory:

- b. 'npm start'

```
C:\Users\gotta\Documents\Tools\SCOM\SCOM4class2>npm start
> SCOM_4811@0.0.0 start C:\Users\gotta\Documents\Tools\SCOM\SCOM4class2
> node ./bin/www

HTTP Server is up
Listening at: http://localhost:3001
open serial communication
_3652
_3651
_3652
```

- c. **If you want to STOP the node server just type Ctl-C from the command prompt to interrupt the program.**

- 13) You should now be able to view the webpage at: <http://localhost:3001>
- 14) Make a short video or show me in class of your system interacting with the browser
- Show the potentiometer readings update the browser
 - Show the LED values turned on digitally
 - Show the LED values can be activated with a dimmer switch using analogwrite()
- 15) Edit #sandbox section of Webpage based on SCOM input & user interaction
- Write your name, the potmeter value, and a date stamp everytime you Toggle the button
 - Turn the background color of the #sandbox section to GREEN if potmeter > 600, and RED if potmeter is < 600;
 - If the slider is at it's max value, make an "alert" call that says: "WARNING: Slider is at MAXIMUM value"

Part III. Questions

- 16) Looking at the Serial output from the command line and reading the comments in /public/javascripts/arduinoData.js -> function parseSerialData()
- What are the parts of the format string for the serial output (value) sent by the Arduino Sketch?
- 17) Open the Arduino Sketch file / _ Arduino/ArduinoNodeJS/ArduinoNodeJS.ino and find where the data is being sent to the serial port.
- What is the 'index' value used that is sent along with the data?
 - How can you change this index value to another number between 0-9 (write the code line)
- 18) Most Node.js webserver default to run on http port 3xxx. Change the Node.js webserver to a different port in the 3000s. (ie. 3023)
- inside /bin/www, look at where httpServer is created. Edit the port number
 - Stop the Node webserver (Ctl-c from the command line)
 - Re-run the Node webserver and check in the browser that it is running on the new port
 - screenshot your browser address line with the updated port value

Bonus

try adding the a Canvas.js chart to the #sandbox section of your webpage:

<https://canvasjs.com/html5-javascript-line-chart/>

<https://webdesign.tutsplus.com/tutorials/how-to-create-a-simple-line-chart-with-chartjs--cms-28129>
<https://canvasjs.com/docs/charts/basics-of-creating-html5-chart/>
<https://canvasjs.com/docs/charts/basics-of-creating-html5-chart/updating-chart-options/>
<https://canvasjs.com/javascript-charts/dynamic-live-line-chart/>