Tutorial on Lab-8

What is Node-Red:

- Node-RED is a flow-based programming environment for wiring together hardware devices, APIs and online services as part of IoT.
    - Network of connected objects that are able to collect and exchange data in real time using embedded sensors. Thermostats, cars, lights, refrigerators, and more appliances can all be connected to the **IoT.**
- You can say, Node-RED as a graphical user interface (GUI) environment to Node.js for developing IoT flows.
- It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

Node-RED was developed by IBM Emerging Technology and the open-source community.

It works by allowing you to wire-up web services or custom nodes to each other, or to things, to do thinks like:

- Send an email on a rainy weather forecast
- Push sensor data to services like Twitter
- Perform complex analysis on data with ease

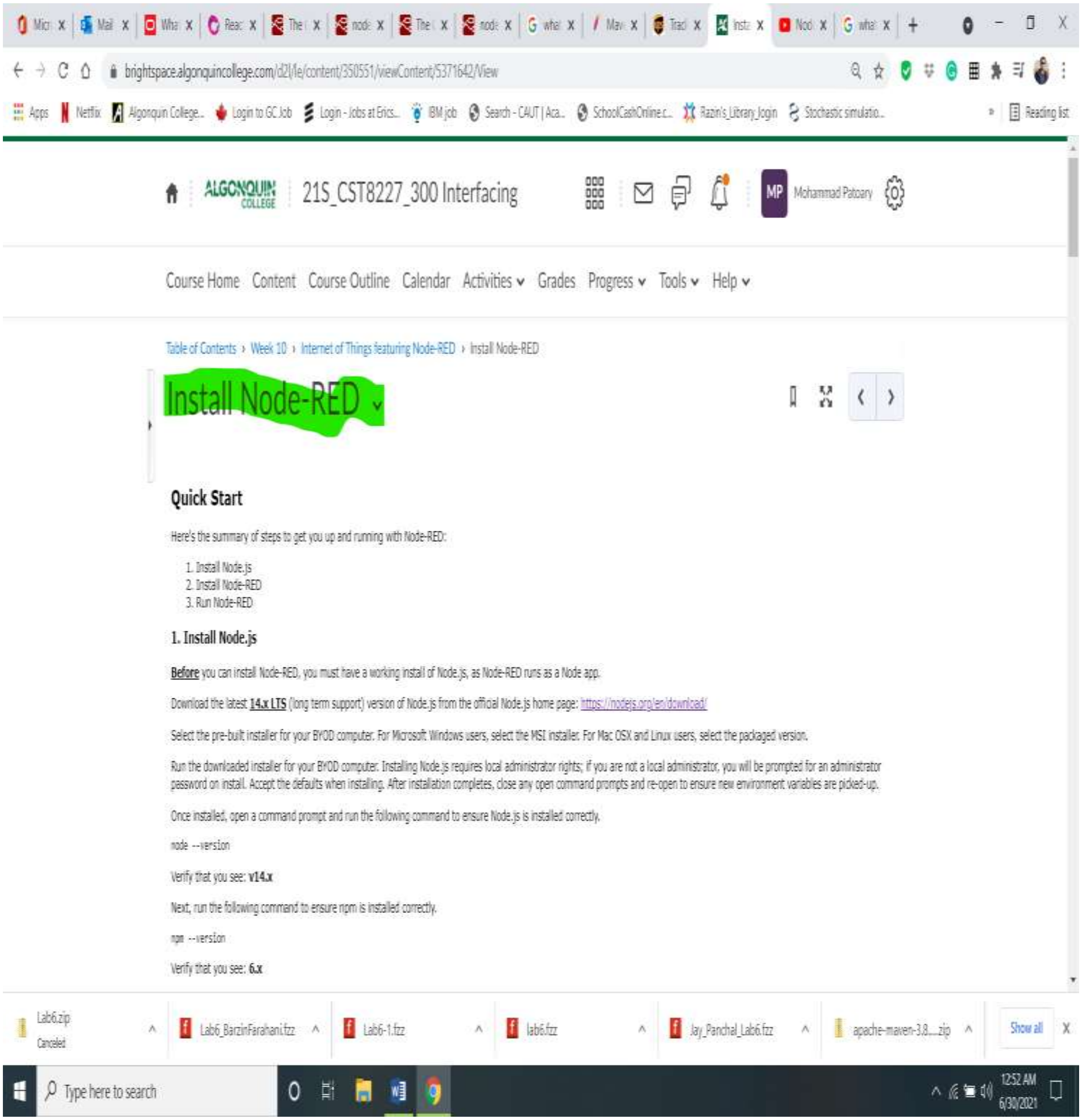
JavaScript functions can be created within the editor using a rich text editor.

A built-in library allows you to save useful functions, templates or flows for re-use.

The flows created in Node-RED arcs stored using JSON, (JavaScript Object Notation) which can be easily imported and exported for sharing.

An online flow library allows you to share your best flows with the world.

Install Node-RED: Please follow the BS post.



brightspace.algonquincollege.com/d2l/le/content/350551/viewContent/5371642/View
21S_CST8227_300 Interfacing
Course Home Content Course Outline Calendar Activities Grades Progress Tools Help
Table of Contents › Week 10 › Internet of Things featuring Node-RED › Install Node-RED

Install Node-RED

Quick Start

Here's the summary of steps to get you up and running with Node-RED:

1. Install Node.js
2. Install Node-RED
3. Run Node-RED

1. Install Node.js

Before you can install Node-RED, you must have a working install of Node.js, as Node-RED runs as a Node app.

Download the latest **14.x LTS** (long term support) version of Node.js from the official Node.js home page: https://nodejs.org/en/download/

Select the pre-built installer for your BYOD computer. For Microsoft Windows users, select the MSI installer. For Mac OSX and Linux users, select the packaged version.

Run the downloaded installer for your BYOD computer. Installing Node.js requires local administrator rights; if you are not a local administrator, you will be prompted for an administrator password on install. Accept the defaults when installing. After installation completes, close any open command prompts and re-open to ensure new environment variables are picked-up.

Once installed, open a command prompt and run the following command to ensure Node.js is installed correctly.

```
node --version
```

Verify that you see: **v14.x**

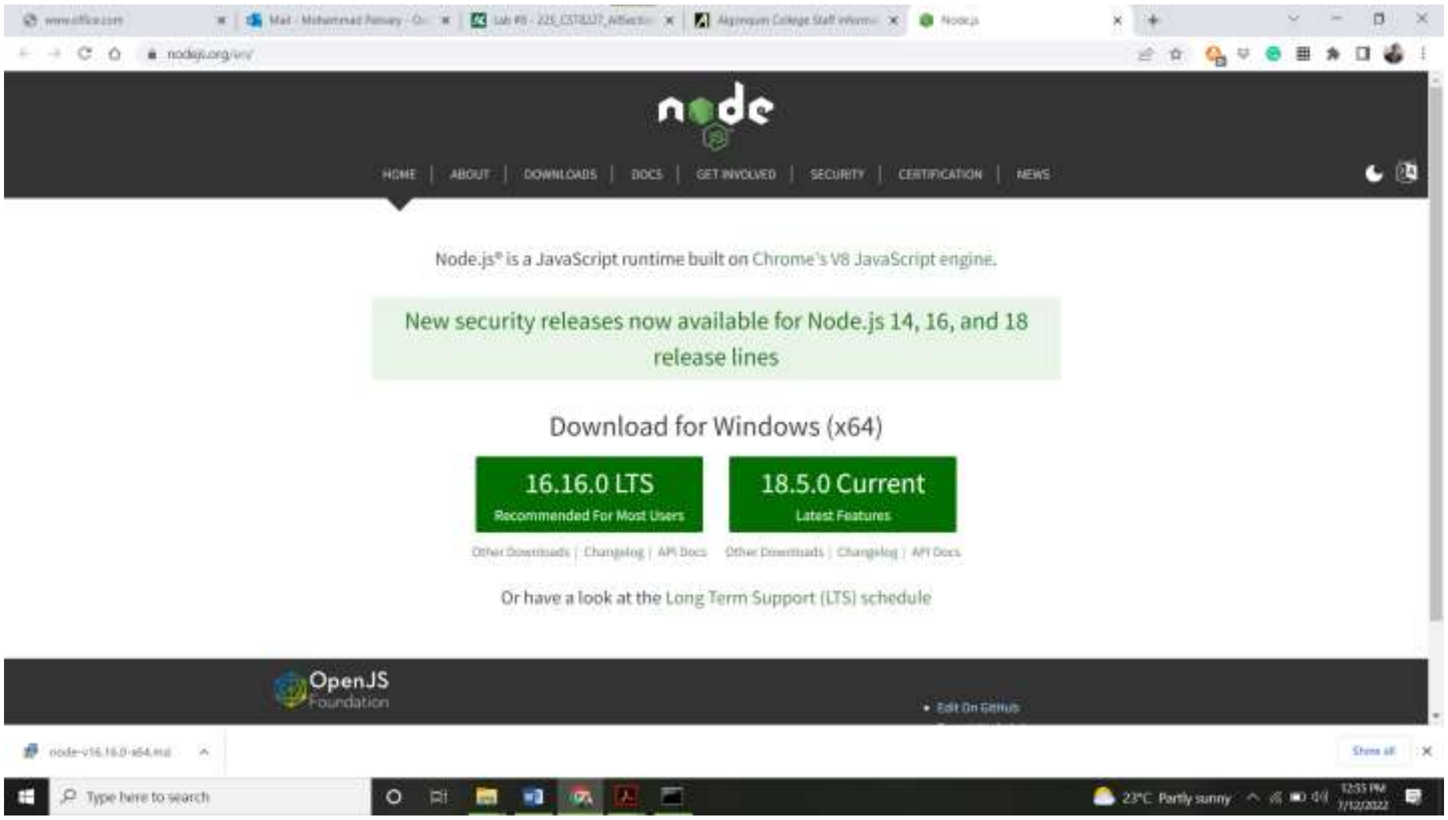
Next, run the following command to ensure npm is installed correctly.

```
npm --version
```

Verify that you see: **6.x**

Download and install Node-red:

Download version 14.17.3 LTS from here: https://nodejs.org/en/


Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

New security releases now available for Node.js 14, 16, and 18 release lines

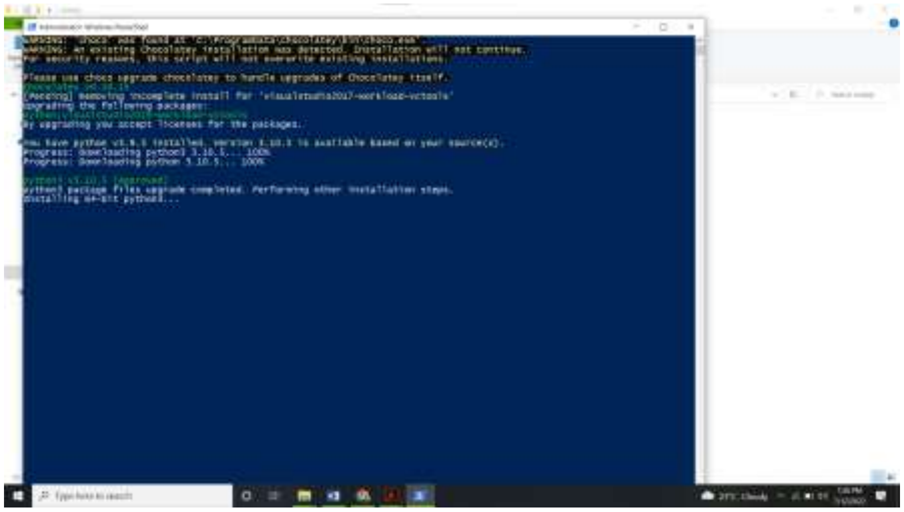
Download for Windows (x64)

16.16.0 LTS
Recommended For Most Users
18.5.0 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs

Or have a look at the Long Term Support (LTS) schedule

node-v14.17.1-x64 6/24/2021 1:44 PM Windows Installer ... 29,852 KB

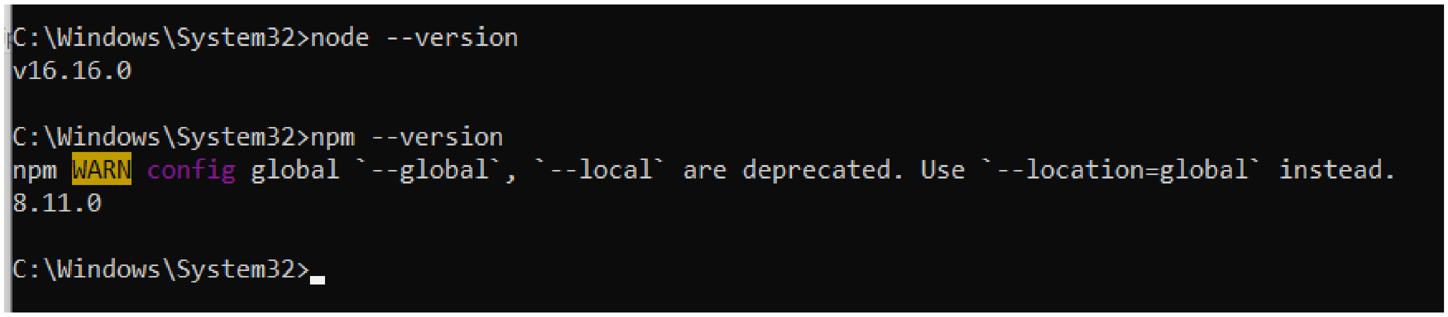
```
Run this exec file and install it.
```



```
WARNING: 'choco' was found at 'C:\ProgramData\chocolatey\bin\choco.exe'.
WARNING: An existing Chocolatey installation was detected. Installation will not continue.
For security reasons, this script will not overwrite existing installations.

Please use choco upgrade chocolatey to handle upgrades of Chocolatey itself.
Chocolatey v0.10.15
[Pending] Removing incomplete install for 'visualstudio2017-workload-vctools'
Upgrading the following packages:
python;visualstudio2017-workload-vctools
By upgrading you accept licenses for the packages.

You have python v3.9.5 installed. Version 3.10.5 is available based on your source(s).
Progress: Downloading python3 3.10.5... 100%
Progress: Downloading python 3.10.5... 100%

python3 v3.10.5 [Approved]
python3 package files upgrade completed. Performing other installation steps.
Installing 64-bit python3...
```

After installation:

Open another terminal to check its versions:



```
C:\Windows\System32>node --version
v16.16.0

C:\Windows\System32>npm --version
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
8.11.0

C:\Windows\System32>_
```

Alternate way to install Node-red i.e. cloning a version:

You need to install git from here first: https://phoenixnap.com/kb/how-to-install-git-windows

Then follow this way: https://github.com/node-red/node-red

git clone https://github.com/node-red/node-red.git

cd node-red

npm install

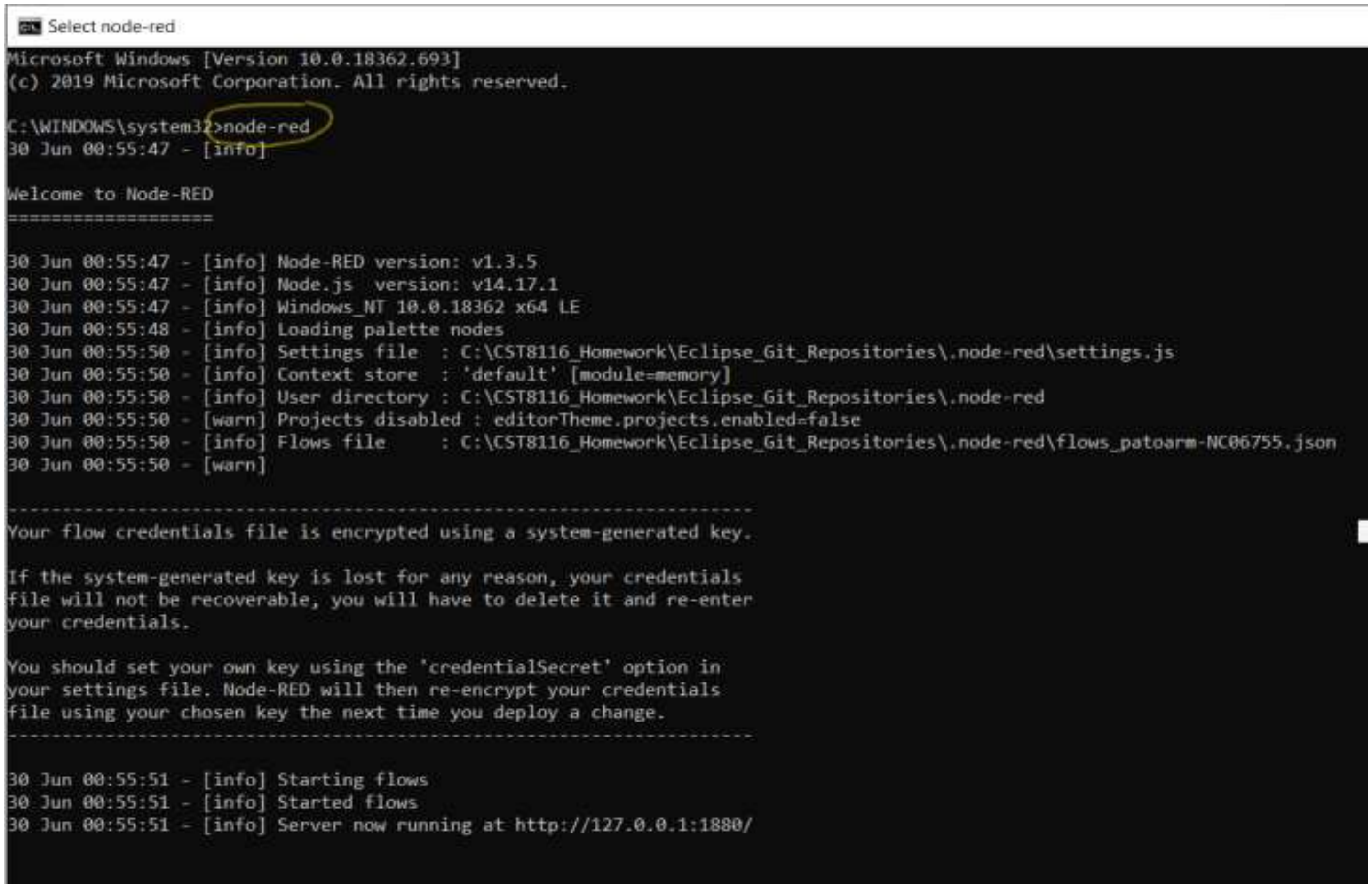
npm run build

npm start

Now install node-red globally here: **`npm install -g --unsafe-perm node-red`**

`C:\node-red`

Start node-red server:



Select node-red

```
Microsoft Windows [Version 10.0.18362.693]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>node-red
30 Jun 00:55:47 - [info]

Welcome to Node-RED
===================

30 Jun 00:55:47 - [info] Node-RED version: v1.3.5
30 Jun 00:55:47 - [info] Node.js  version: v14.17.1
30 Jun 00:55:47 - [info] Windows_NT 10.0.18362 x64 LE
30 Jun 00:55:48 - [info] Loading palette nodes
30 Jun 00:55:50 - [info] Settings file  : C:\CST8116_Homework\Eclipse_Git_Repositories\.node-red\settings.js
30 Jun 00:55:50 - [info] Context store  : 'default' [module=memory]
30 Jun 00:55:50 - [info] User directory : C:\CST8116_Homework\Eclipse_Git_Repositories\.node-red
30 Jun 00:55:50 - [warn] Projects disabled : editorTheme.projects.enabled=false
30 Jun 00:55:50 - [info] Flows file     : C:\CST8116_Homework\Eclipse_Git_Repositories\.node-red\flows_patoarm-NC06755.json
30 Jun 00:55:50 - [warn]

---------------------------------------------------------------------
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

30 Jun 00:55:51 - [info] Starting flows
30 Jun 00:55:51 - [info] Started flows
30 Jun 00:55:51 - [info] Server now running at http://127.0.0.1:1880/
```

Open http://localhost:1880

Interacting with Teensy: Please follow the BS post. Let’s go there….

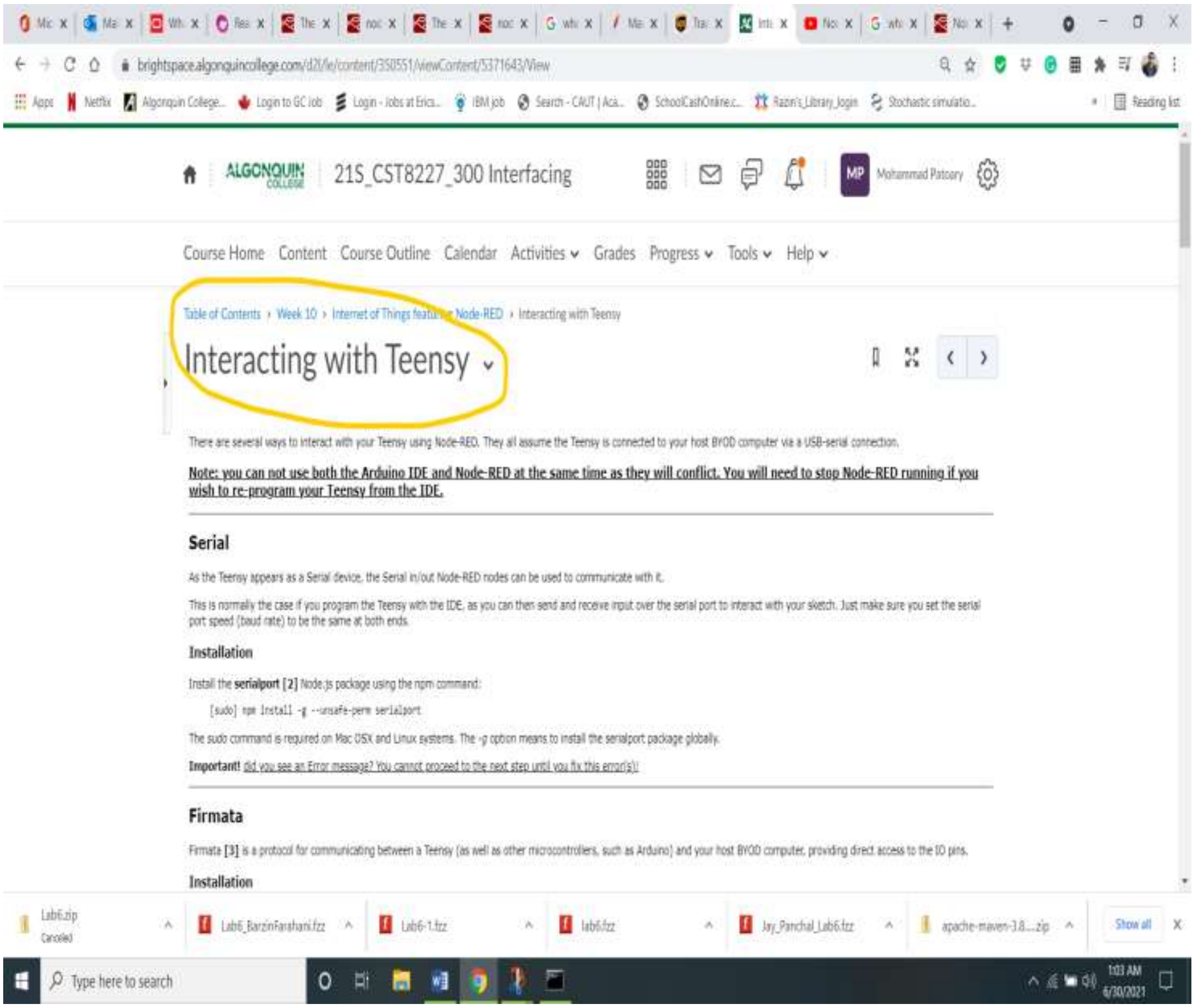

21S_CST8227_300 Interfacing

Table of Contents › Week 10 › Internet of Things featuring Node-RED › Interacting with Teensy

Interacting with Teensy

There are several ways to interact with your Teensy using Node-RED. They all assume the Teensy is connected to your host BYOD computer via a USB-serial connection.

Note: you can not use both the Arduino IDE and Node-RED at the same time as they will conflict. You will need to stop Node-RED running if you wish to re-program your Teensy from the IDE.

Serial

As the Teensy appears as a Serial device, the Serial in/out Node-RED nodes can be used to communicate with it.

This is normally the case if you program the Teensy with the IDE, as you can then send and receive input over the serial port to interact with your sketch. Just make sure you set the serial port speed (baud rate) to be the same at both ends.

Installation

Install the **serialport [2]** Node.js package using the npm command:

```
[sudo] npm install -g --unsafe-perm serialport
```

The sudo command is required on Mac OSX and Linux systems. The -g option means to install the serialport package globally.

Important! did you see an Error message? You cannot proceed to the next step until you fix this error(s)!

Firmata

Firmata [3] is a protocol for communicating between a Teensy (as well as other microcontrollers, such as Arduino) and your host BYOD computer, providing direct access to the IO pins.

Installation

First, Install serial:

As the Teensy appears as a Serial device, the Serial in/out Node-RED nodes can be used to communicate with it.

Install the **serialport from a terminal** using the npm command:

```
C:\npm install -g --unsafe-perm serialport
```

Second, Install Firmata:

Firmata is a protocol for communicating between a Teensy (as well as other microcontrollers, such as Arduino) and your host BYOD computer, providing direct access to the IO pins.

First you need to load the default Firmata sketch onto the Teensy. Launch the **Arduino IDE** and open the sketch:

File > Examples > Firmata > Standard Firmata Plus

Make sure the sketch is successfully uploaded to your Teensy before proceeding to the next step.

Modify the Standard Firmata Plus Sketch

By default the Standard Firmata Plus sketch does not blink the onboard LED upon start-up. You can change that behaviour if you like.

Edit the StandardFirmataPlus.ino sketch file, and comment-out **Line 794**: Just put //

```
//Firmata.disableBlinkVersion();
```

Then verify and upload it. You do not need to save it. **NB**: your teensy should be connected there. Here you are just loading the program i.e. protocol here… you will see a blinking when it is loaded. that's it.

Third, Install Node-RED Arduino Nodes to your user directory:

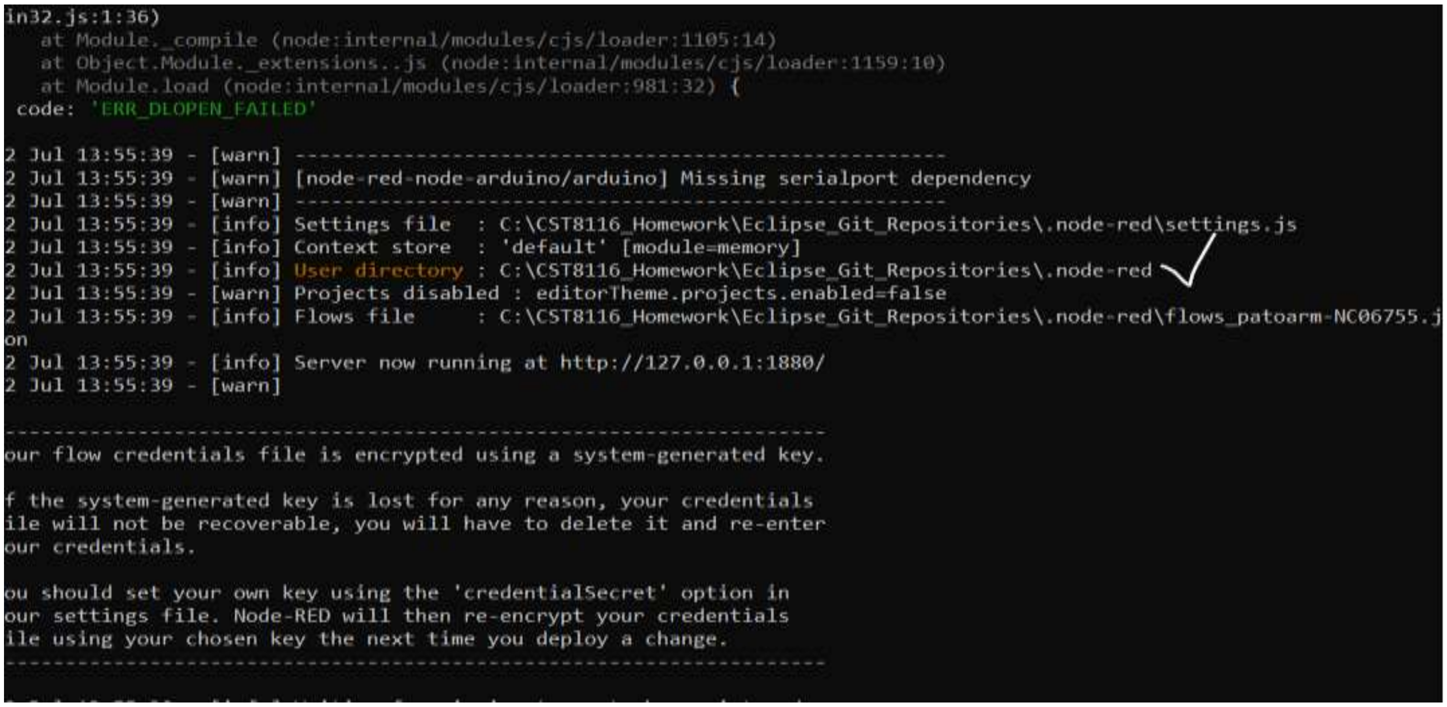
By default, Node-RED does not include any nodes to interact with a microcontroller, such as your Teensy.

You then need to install the Node-RED *Arduino* nodes into the palette.

On Windows PC, your command will be: >**`cd C:\Users\{yourUsername}\.node-red`**

`For me, the path was:`
`C:\CST8116_Homework\Eclipse_Git_Repositories\.node-red`

`You can see it here: after running your server…..`



```
in32.js:1:36)
   at Module._compile (node:internal/modules/cjs/loader:1105:14)
   at Object.Module._extensions..js (node:internal/modules/cjs/loader:1159:10)
   at Module.load (node:internal/modules/cjs/loader:981:32) {
 code: 'ERR_DLOPEN_FAILED'

2 Jul 13:55:39 - [warn] ------------------------------------------------------
2 Jul 13:55:39 - [warn] [node-red-node-arduino/arduino] Missing serialport dependency
2 Jul 13:55:39 - [warn] ------------------------------------------------------
2 Jul 13:55:39 - [info] Settings file  : C:\CST8116_Homework\Eclipse_Git_Repositories\.node-red\settings.js
2 Jul 13:55:39 - [info] Context store  : 'default' [module=memory]
2 Jul 13:55:39 - [info] User directory : C:\CST8116_Homework\Eclipse_Git_Repositories\.node-red ✓
2 Jul 13:55:39 - [warn] Projects disabled : editorTheme.projects.enabled=false
2 Jul 13:55:39 - [info] Flows file     : C:\CST8116_Homework\Eclipse_Git_Repositories\.node-red\flows_patoarm-NC06755.j
on
2 Jul 13:55:39 - [info] Server now running at http://127.0.0.1:1880/
2 Jul 13:55:39 - [warn]

---------------------------------------------------------------------
our flow credentials file is encrypted using a system-generated key.

f the system-generated key is lost for any reason, your credentials
ile will not be recoverable, you will have to delete it and re-enter
our credentials.

ou should set your own key using the 'credentialSecret' option in
our settings file. Node-RED will then re-encrypt your credentials
ile using your chosen key the next time you deploy a change.
---------------------------------------------------------------------
```

Then install the Arduino nodes:

`npm install node-red-node-arduino`

npm will install the package locally to this folder.

Finally, restart (or start) Node-RED, and reload the editor in the browser. There should now be two new Arduino nodes in the palette.

Yes - proceed to the next step, Blink Flow :)

No - you cannot proceed to the next step, and must trouble-shoot. The most frequent issue for an error is... the **serialport** node module does not work with your BYOD laptop. By default, the serialport module installs pre-built binaries for the popular operating systems (Windows, Mac, Linux). For what ever reason, the pre-built binaries for your BYOD laptop are incompatible and do not work. The solution is to build the serialport module from source. You'll need to consult the official documentation for details on how to build from source. Here are the links:

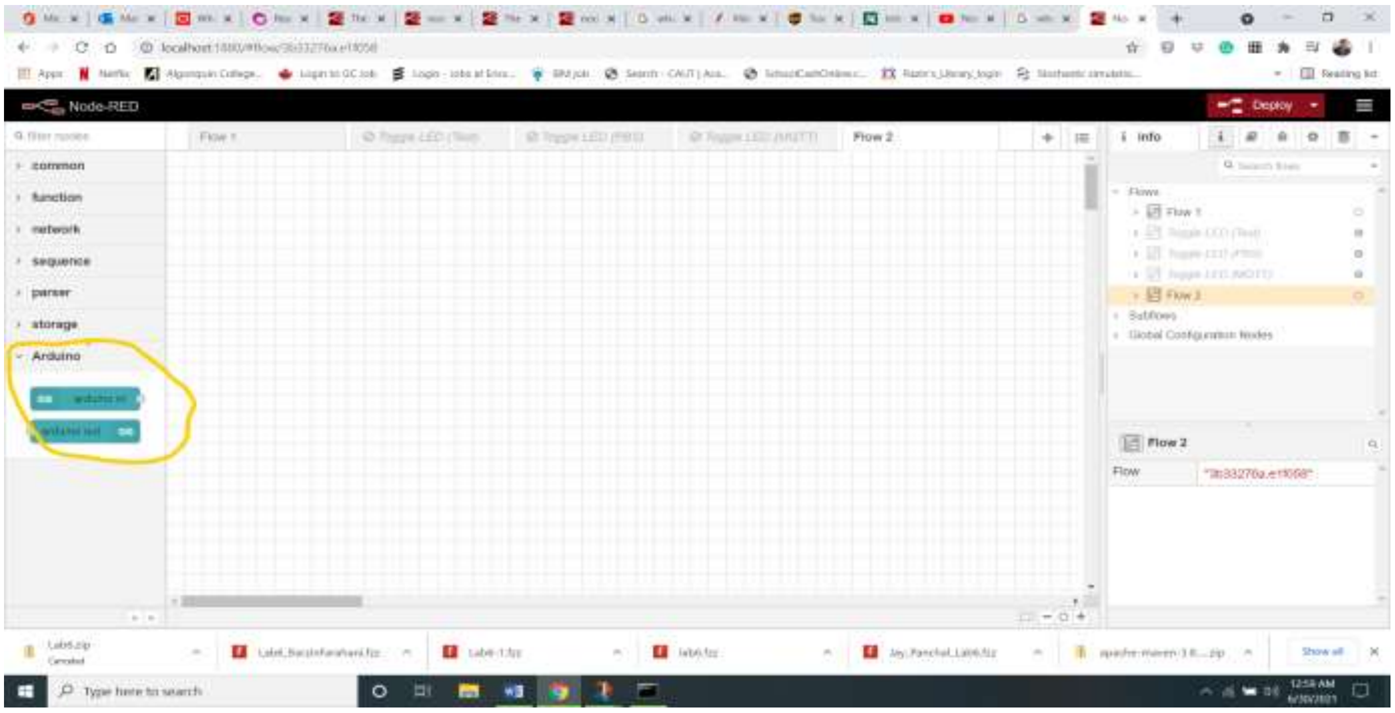
- Installing SerialPort - scroll to the appropriate section for your BYOD's operating system
- Installing node-gyp - node-gyp is a cross-platform command-line tool written in Node.js for compiling native addon modules for Node.js
  - Note: you may be required to install other software packages, such as python. You'll need to consult the documentation for your BYOD computer.

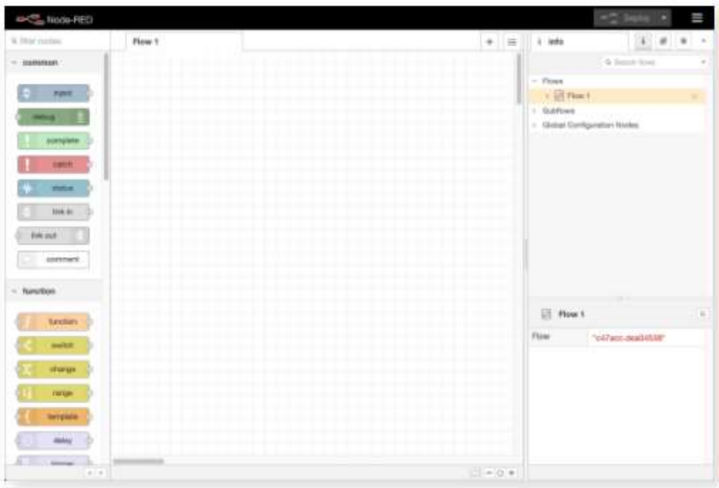
Special Note for Windows BYOD: follow these instructions to help install Node-RED https://nodered.org/docs/getting-started/windows

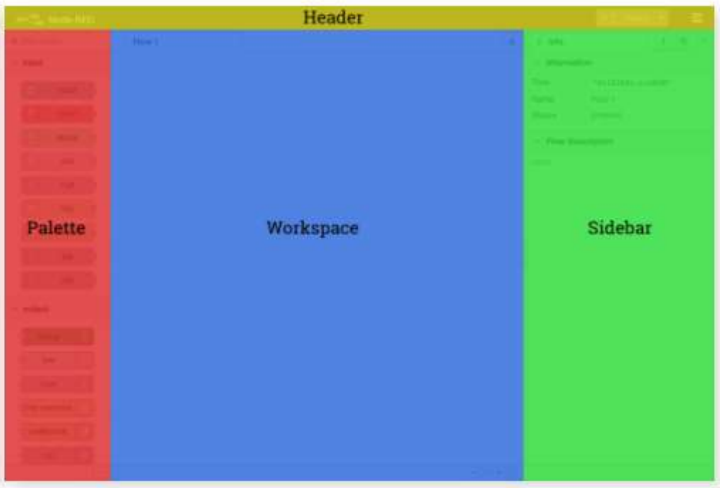
Try running following commands to build the serialport module from source:

```
sudo npm -g install node-gyp
sudo npm -g install serialport @serialport/list --build-from-source --unsafe-perm
sudo npm -g install node-red --build-from-source
node-red
```

Note: node-gyp requires Python **3.9** (note version) on you BYOD. You may need to edit your $PATH environment variable to include $PYTHON_HOME\bin

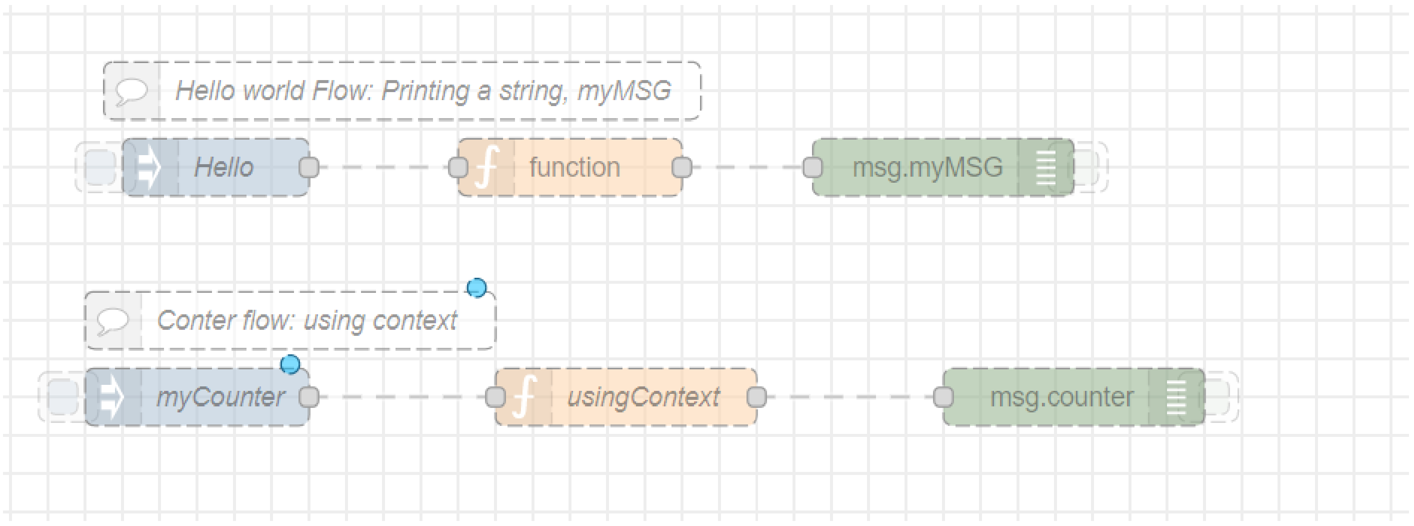



Editor window



Editor components

Basic flow programming:


Hello world Flow: Printing a string, myMSG
Hello
function
msg.myMSG
Conter flow: using context
myCounter
usingContext
msg.counter

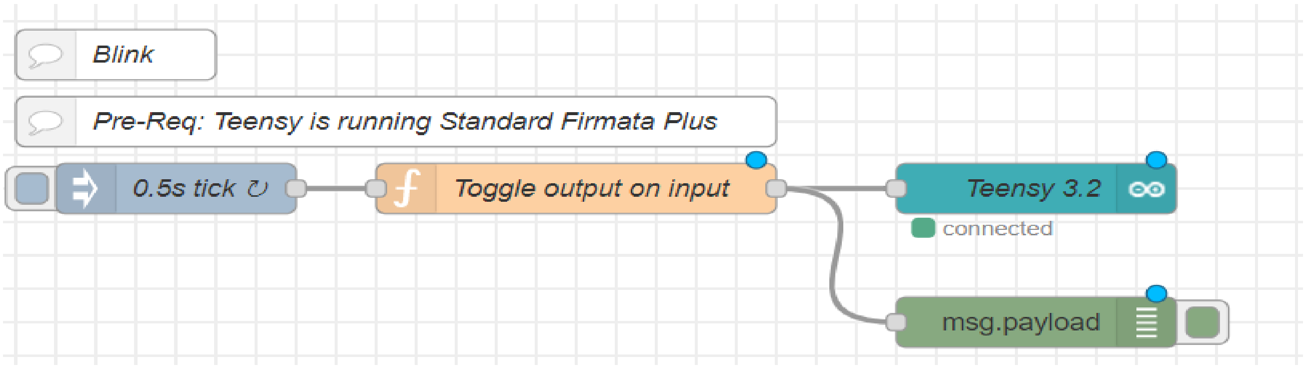
What is Context?

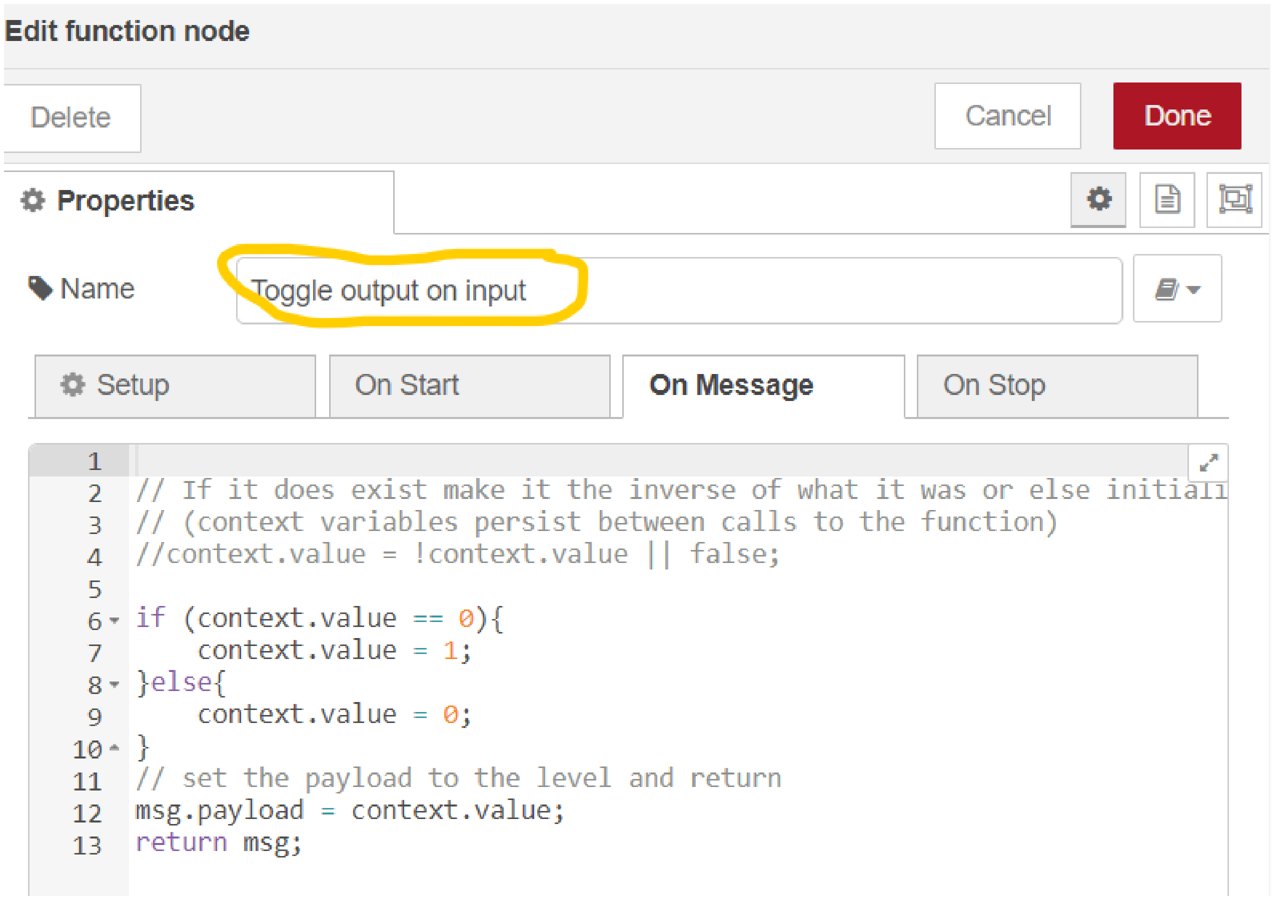
Node-RED provides a way to store information that can be shared between different nodes without using the messages that pass through a flow. This is called ‘context’.

There are 3 context scope:

- Node - only visible to the node that set the value.
- Flow - visible to all nodes on the same flow (or tab in the editor)
- Global - visible to all nodes

Blink Flow:

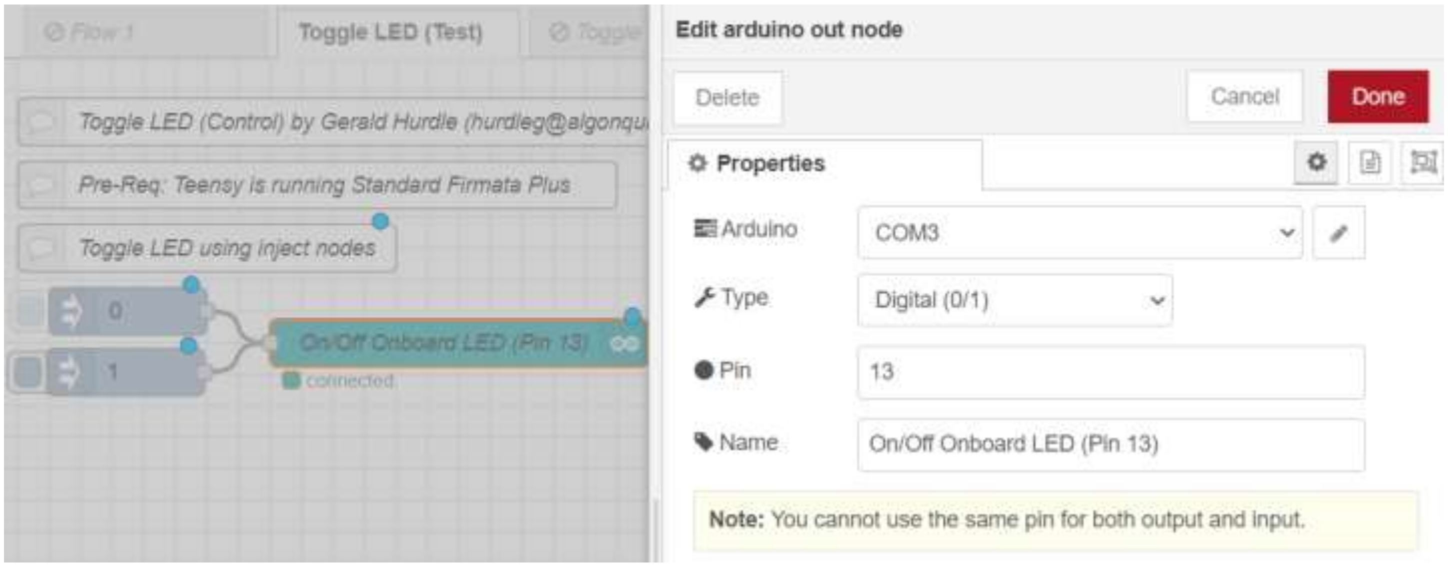




Name: Toggle output on input

```
// If it does exist make it the inverse of what it was or else initiali
// (context variables persist between calls to the function)
//context.value = !context.value || false;

if (context.value == 0){
    context.value = 1;
}else{
    context.value = 0;
}
// set the payload to the level and return
msg.payload = context.value;
return msg;
```

Toggle:


Toggle LED (Test)

Toggle LED (Control) by Gerald Hurdle (hurdleg@algonqu

Pre-Req: Teensy is running Standard Firmata Plus

Toggle LED using inject nodes

0

1

On/Off Onboard LED (Pin 13)

connected

Edit arduino out node

Delete Cancel Done

Properties

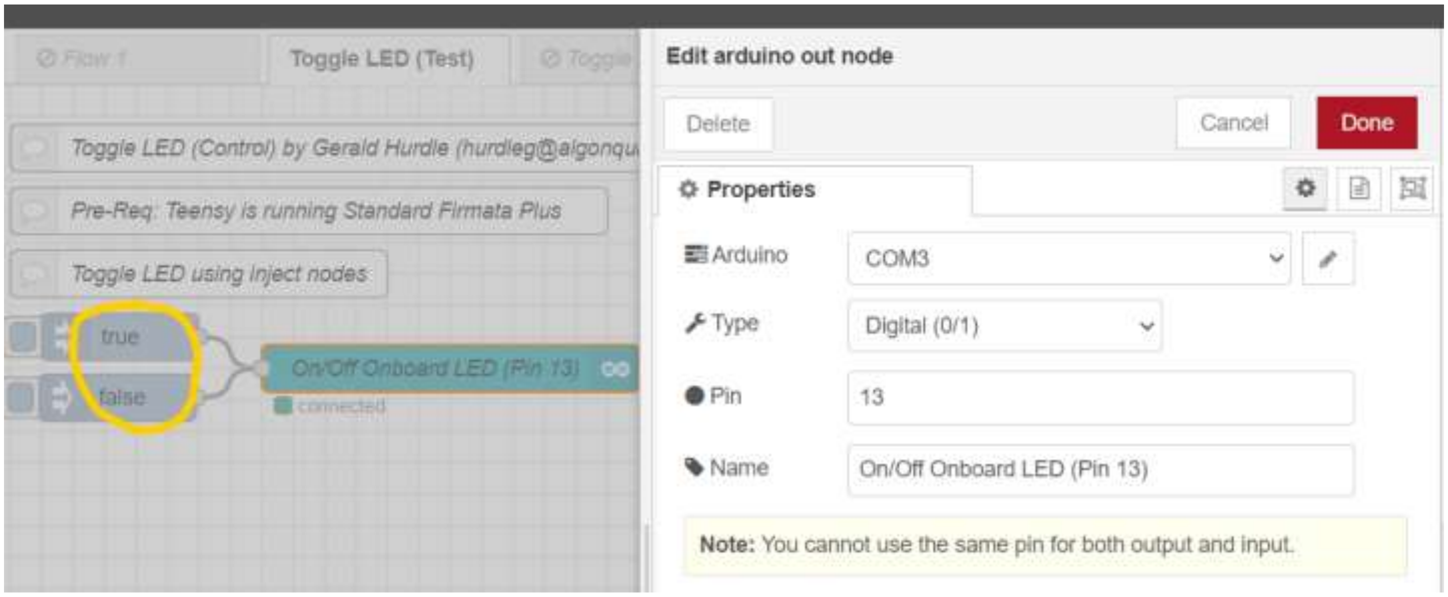
Arduino: COM3

Type: Digital (0/1)

Pin: 13

Name: On/Off Onboard LED (Pin 13)

Note: You cannot use the same pin for both output and input.


Toggle LED (Test)

Toggle LED (Control) by Gerald Hurdle (hurdleg@algonqu

Pre-Req: Teensy is running Standard Firmata Plus

Toggle LED using inject nodes

true

false

On/Off Onboard LED (Pin 13)

connected

Edit arduino out node

Delete Cancel Done

Properties

Arduino: COM3

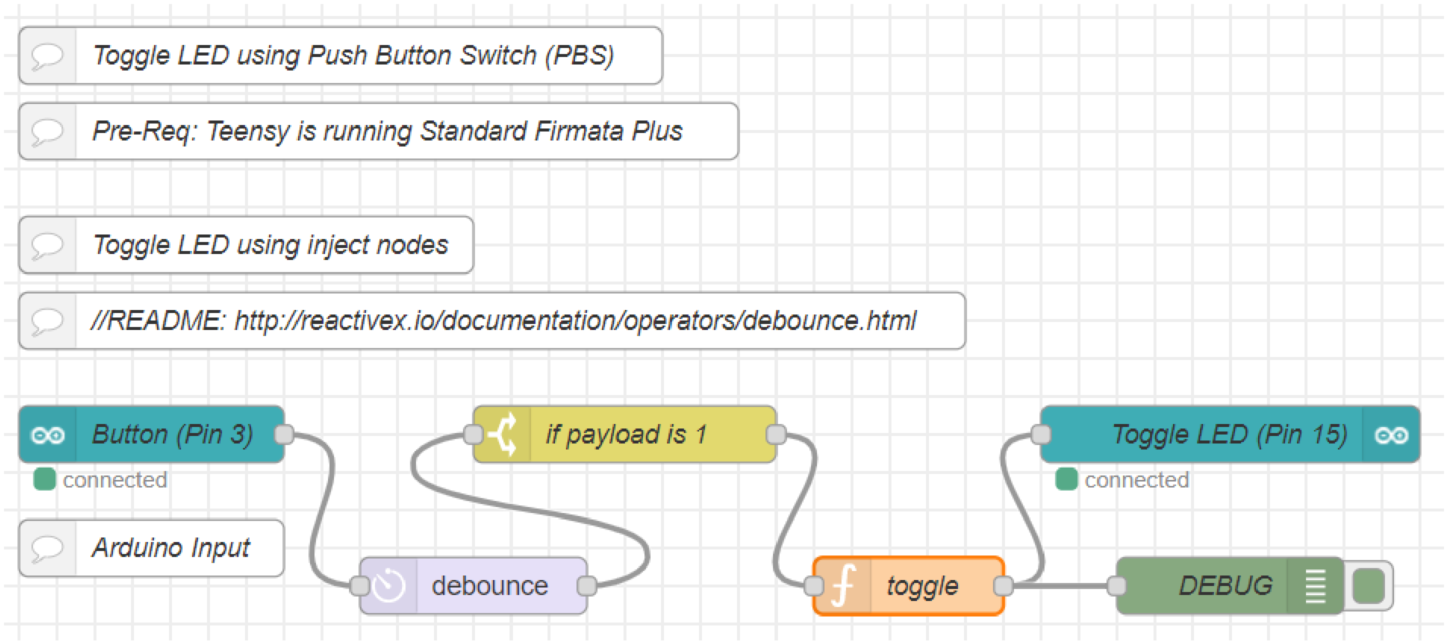
Type: Digital (0/1)

Pin: 13

Name: On/Off Onboard LED (Pin 13)

Note: You cannot use the same pin for both output and input.

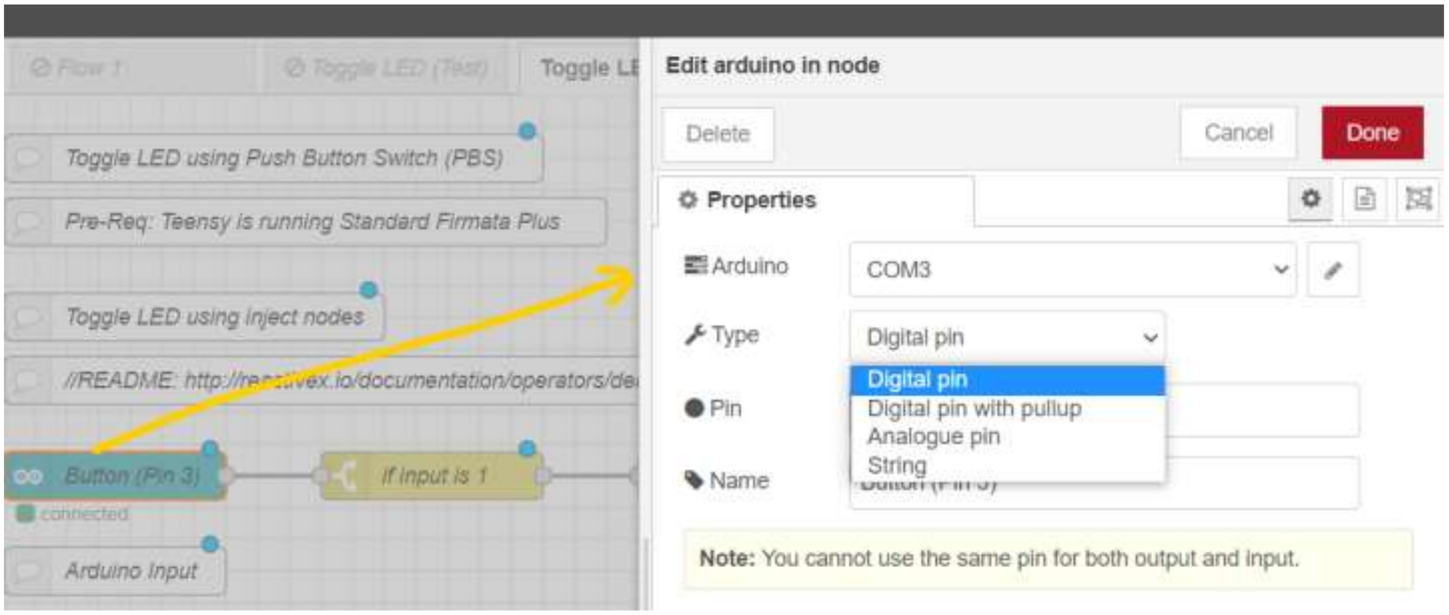
Toggle discrete LED using PBS: Lab-8_B


Toggle LED using Push Button Switch (PBS)

Pre-Req: Teensy is running Standard Firmata Plus

Toggle LED using inject nodes

//README: http://reactivex.io/documentation/operators/debounce.html


Edit arduino in node

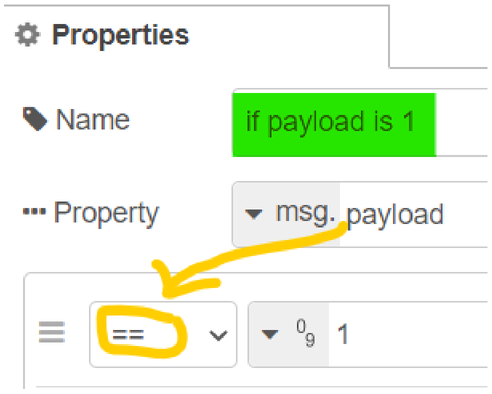
Arduino: COM3

Type: Digital pin

Name: Button (Pin 3)

Note: You cannot use the same pin for both output and input.

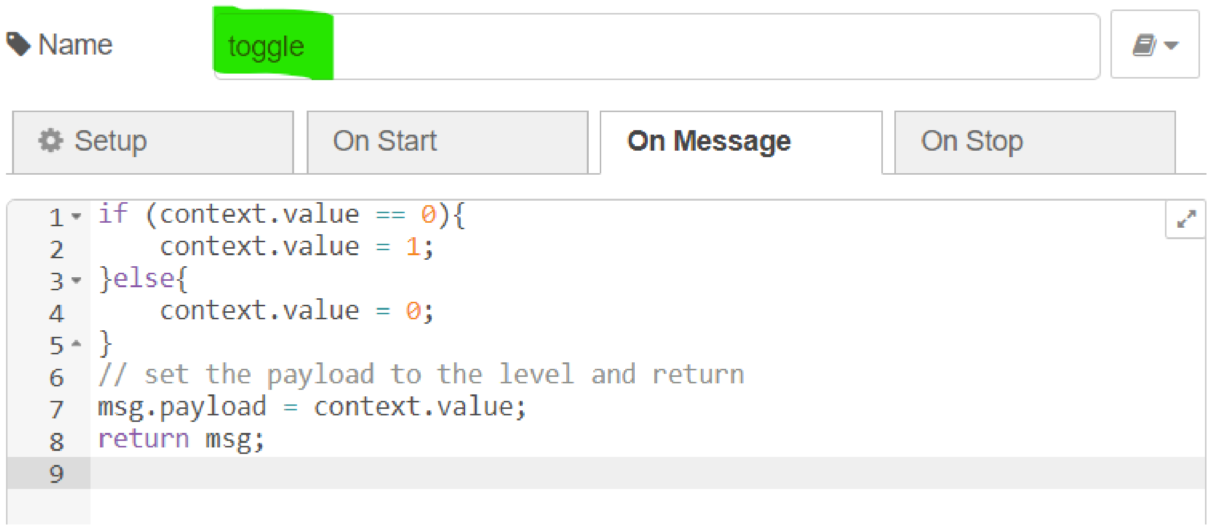
Flow 1
Toggle LED (Test)
Toggle LE
Toggle LED using Push Button Switch (PBS)
Pre-Req: Teensy is running Standard Firmata Plus
Toggle LED using inject nodes
//README: http://reactivex.io/documentation/operators/de
Button (Pin 3)
if input is 1
connected
Arduino Input
Delete
Cancel
Done
Properties
Pin
Digital pin
Digital pin with pullup
Analogue pin
String


Properties

Name: if payload is 1

Property: msg. payload

== 1


Name toggle

Setup | On Start | On Message | On Stop

```
if (context.value == 0){
    context.value = 1;
}else{
    context.value = 0;
}
// set the payload to the level and return
msg.payload = context.value;
return msg;
```

Common nodes:

functions/switches/rbe/etc

Edit -> Deploy -> Restart Node-Server **Cycle**:

When developing your own Node-RED projects, you need to be aware of the Edit -> Deploy -> restart node-red server cycle in 4 steps:

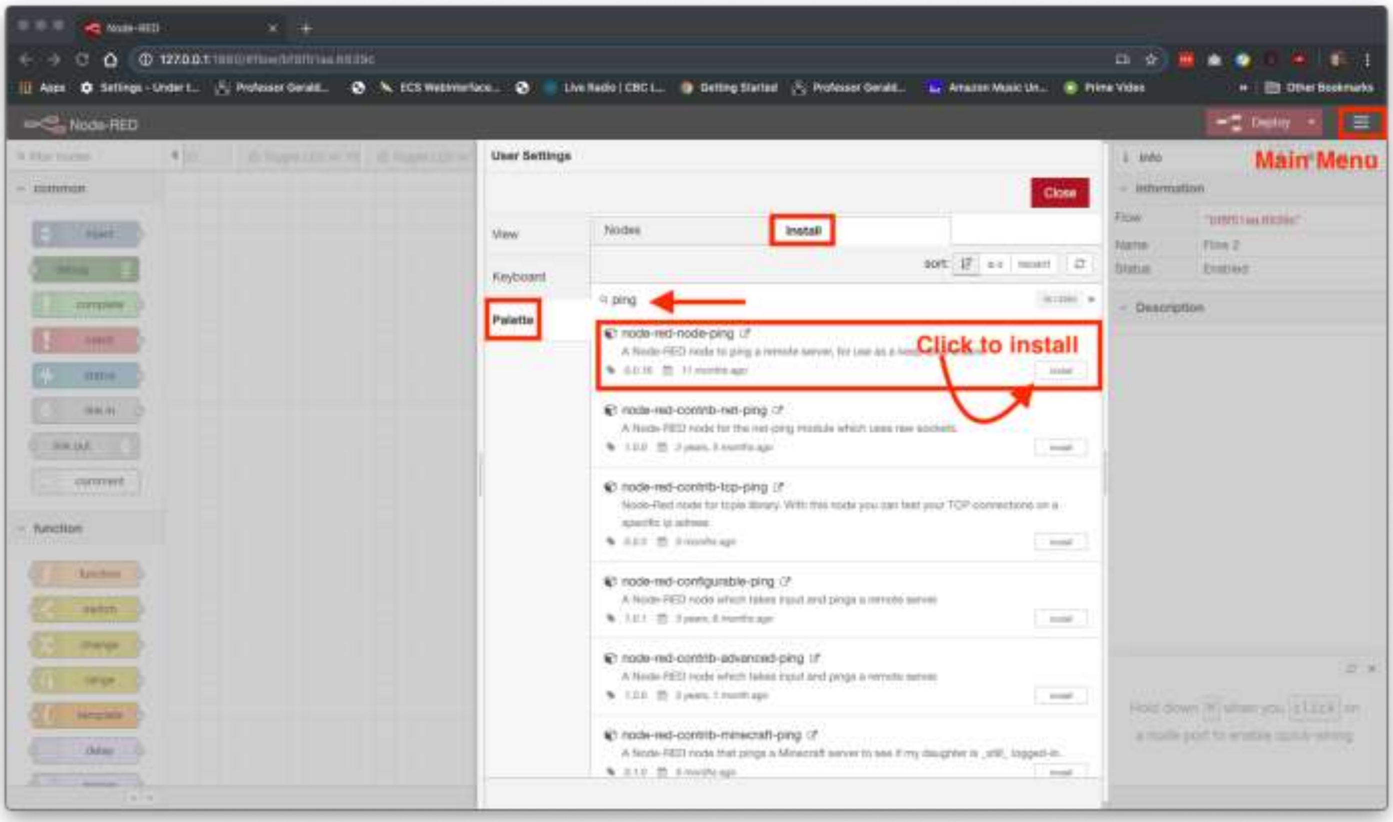
1. Edit your Node-RED flow
2. Click the Deploy button when you've completed your edits
3. Restart the node-red Server
4. Verify all Arduino nodes (in and out) have a status of: connected

Adding nodes to the palette:

Node-RED permits developers to add new nodes to the palette. You can install a node in two ways: using the GUI Editor, or using npm from the command-line.

Using GUI:

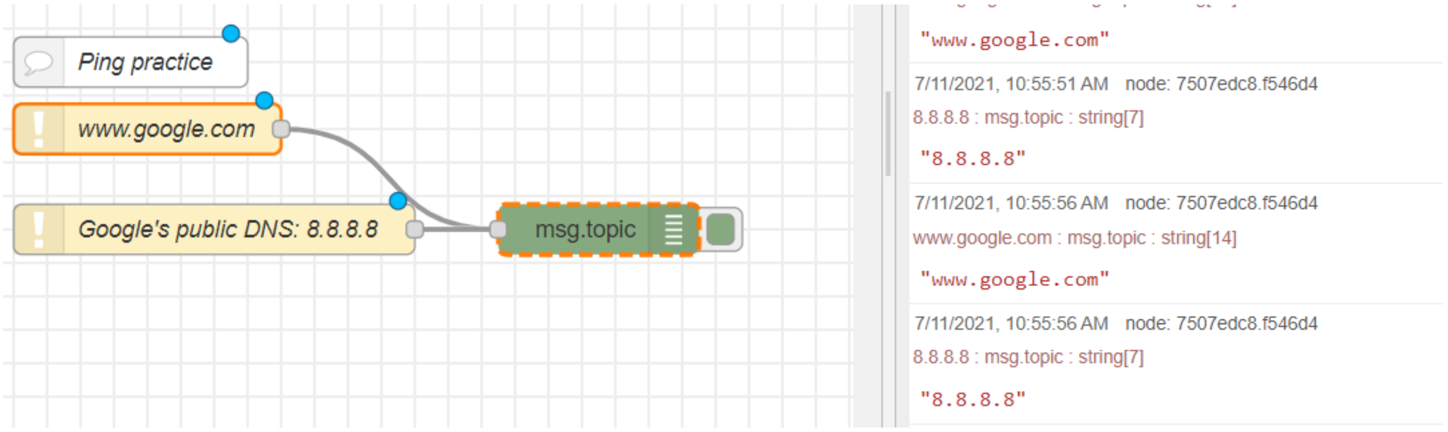
Let's install the **ping** node using the graphical user interface (GUI) Editor.


Node-RED
User Settings
Close
View
Keyboard
Palette
Nodes
Install
ping
Main Menu
Click to install
node-red-node-ping
node-red-contrib-net-ping
node-red-contrib-tcp-ping
node-red-contrib-configurable-ping
node-red-contrib-advanced-ping
node-red-contrib-minecraft-ping

msg.topic contains the target host ip.

You can ping by either name or IPv4 address. Notice Google's DNS server at 8.8.8.8 is a publicly routed Class A address.

You can follow the posted JSN file…. To check the ping node.



```
"www.google.com"
7/11/2021, 10:55:51 AM  node: 7507edc8.f546d4
8.8.8.8 : msg.topic : string[7]
"8.8.8.8"
7/11/2021, 10:55:56 AM  node: 7507edc8.f546d4
www.google.com : msg.topic : string[14]
"www.google.com"
7/11/2021, 10:55:56 AM  node: 7507edc8.f546d4
8.8.8.8 : msg.topic : string[7]
"8.8.8.8"
```

Installing with npm:

To install a node module form the command-line, you can use the following command from within your user data directory.

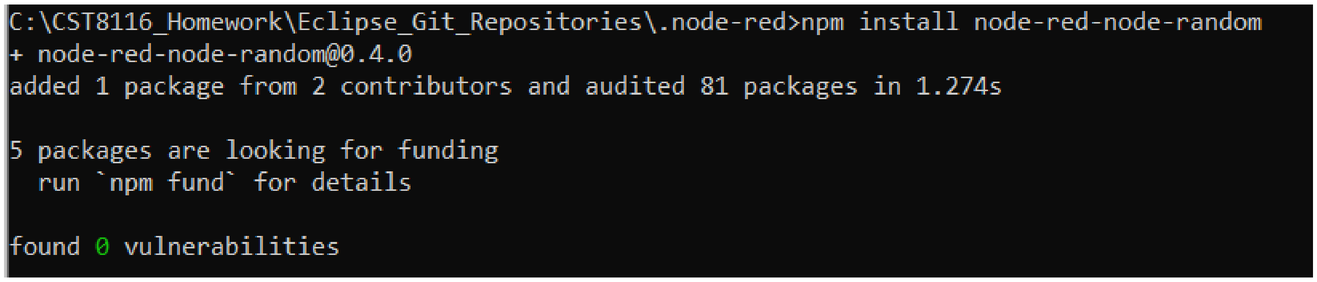
Enter the following commands from Command.app:

```
cd C:\Users\{yourUsername}\.node-red
npm install <npm-package-name>
```

You will then need to restart Node-RED for it to pick-up the new nodes.

Let's install the **random** node from the command line using npm.

For me: C:\CST8116_Homework\Eclipse_Git_Repositories\.node-red



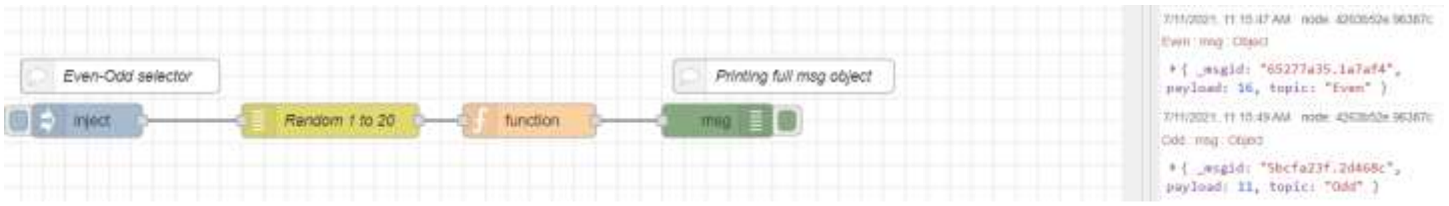
```
C:\CST8116_Homework\Eclipse_Git_Repositories\.node-red>npm install node-red-node-random
+ node-red-node-random@0.4.0
added 1 package from 2 contributors and audited 81 packages in 1.274s

5 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Random node:

The **random** node generates a random number between a low and high value.

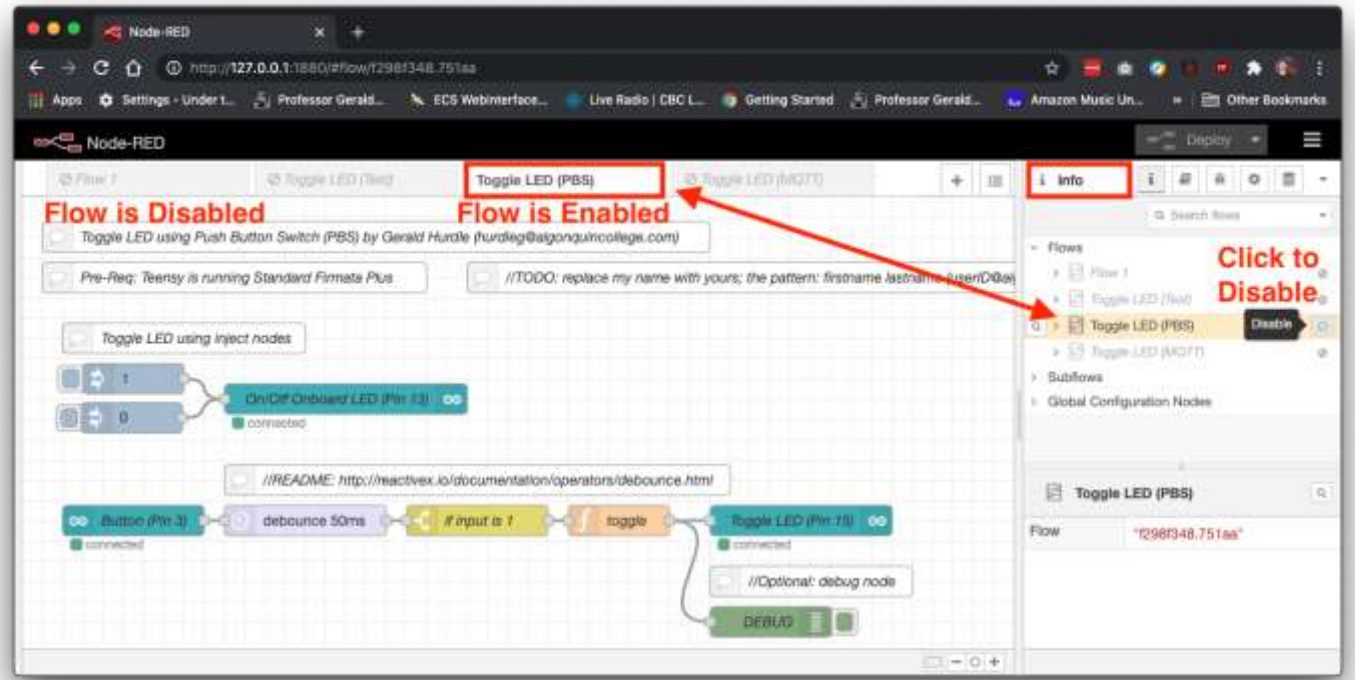
If set to return an integer it can include both the low and high values. **min <= n <= max**

I used a random node with default values. It generates whole integer numbers from 1 to 10 (inclusive).



```
7/11/2021, 11:15:47 AM  node: 4263b52e.9636fc
Even : msg : Object
▸ { _msgid: "65277a35.1a7af4", payload: 16, topic: "Even" }
7/11/2021, 11:15:49 AM  node: 4263b52e.9636fc
Odd : msg : Object
▸ { _msgid: "5bcfa23f.2d468c", payload: 11, topic: "Odd" }
```

Enable / Disable Flows and Nodes: plz follow BS



Toggle LED using Push Button Switch (PBS) by Gerald Hurdle (hurdleg@algonquincollege.com)

Pre-Req: Teensy is running Standard Firmata Plus

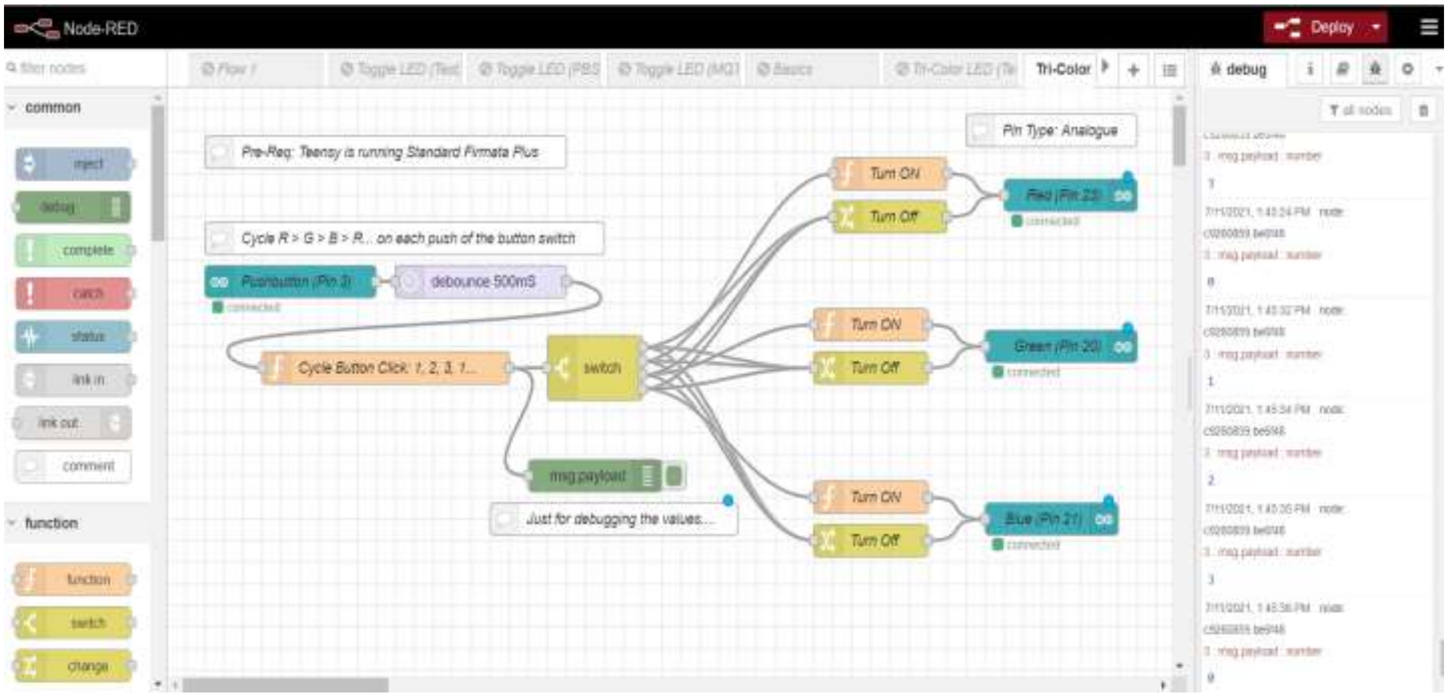
//TODO: replace my name with yours; the pattern: firstname lastname (userID@al

Toggle LED using inject nodes

//README: http://reactivex.io/documentation/operators/debounce.html

//Optional: debug node

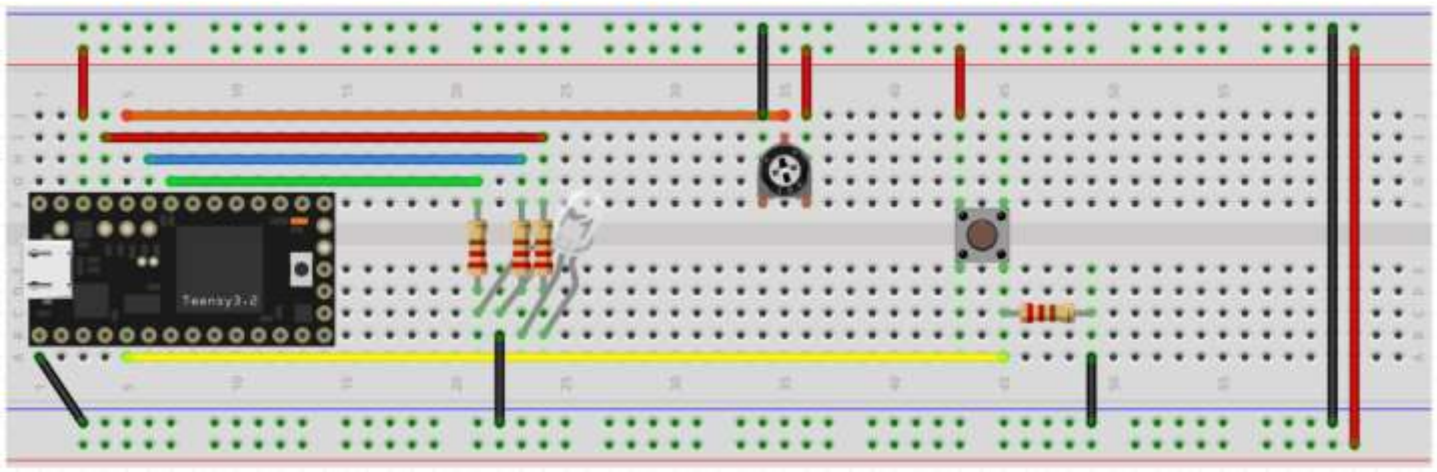
Tri color Led: Cycle R → G → B: (Lab-8 A- Level)

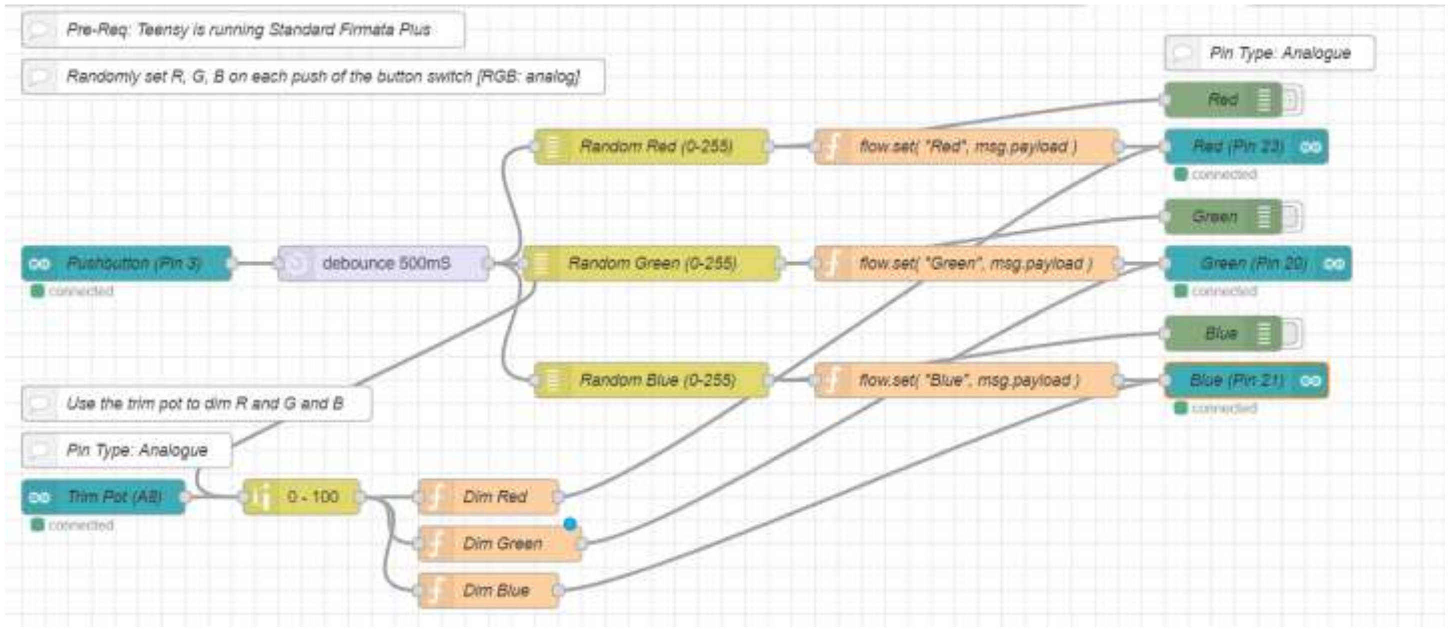


Cycle button click:

```
if (!context.value){
context.value = 0;
}
context.value +=1;
msg.payload = context.value%4;
return msg;
```

Adding potentiometer as dimmer to the tri-color LED: (Lab-8_A+ Level)




Pre-Req: Teensy is running Standard Firmata Plus
Randomly set R, G, B on each push of the button switch [RGB: analog]
Pin Type: Analogue
Red
Random Red (0-255)
flow.set("Red", msg.payload)
Red (Pin 23)
connected
Green
Pushbutton (Pin 3)
debounce 500mS
Random Green (0-255)
flow.set("Green", msg.payload)
Green (Pin 20)
Blue
Random Blue (0-255)
flow.set("Blue", msg.payload)
Blue (Pin 21)
Use the trim pot to dim R and G and B
Pin Type: Analogue
Trim Pot (A8)
0 - 100
Dim Red
Dim Green
Dim Blue

Setting random value to the Red → flow variable.

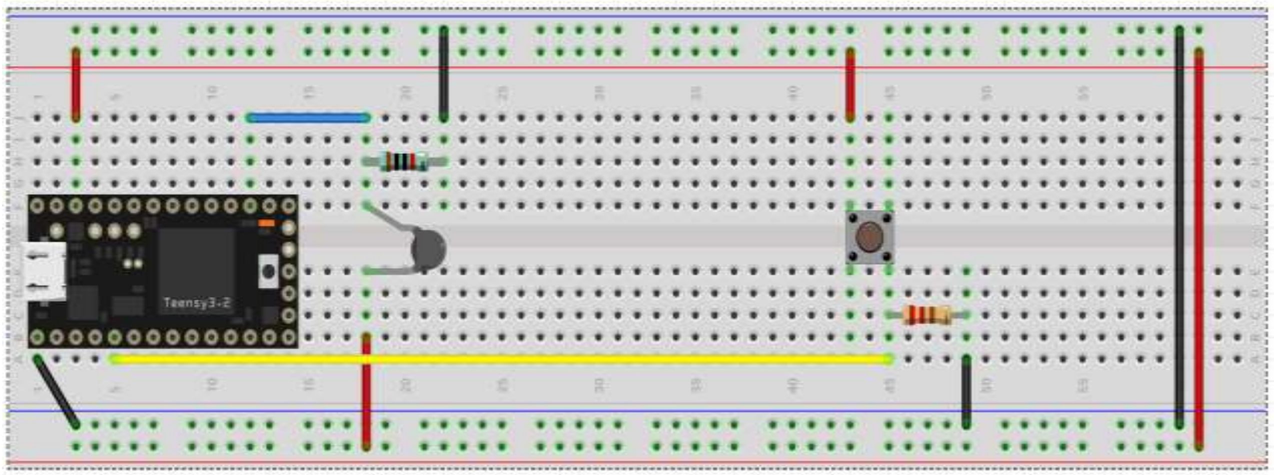
```
flow.set( "Red", msg.payload );
return msg;
```

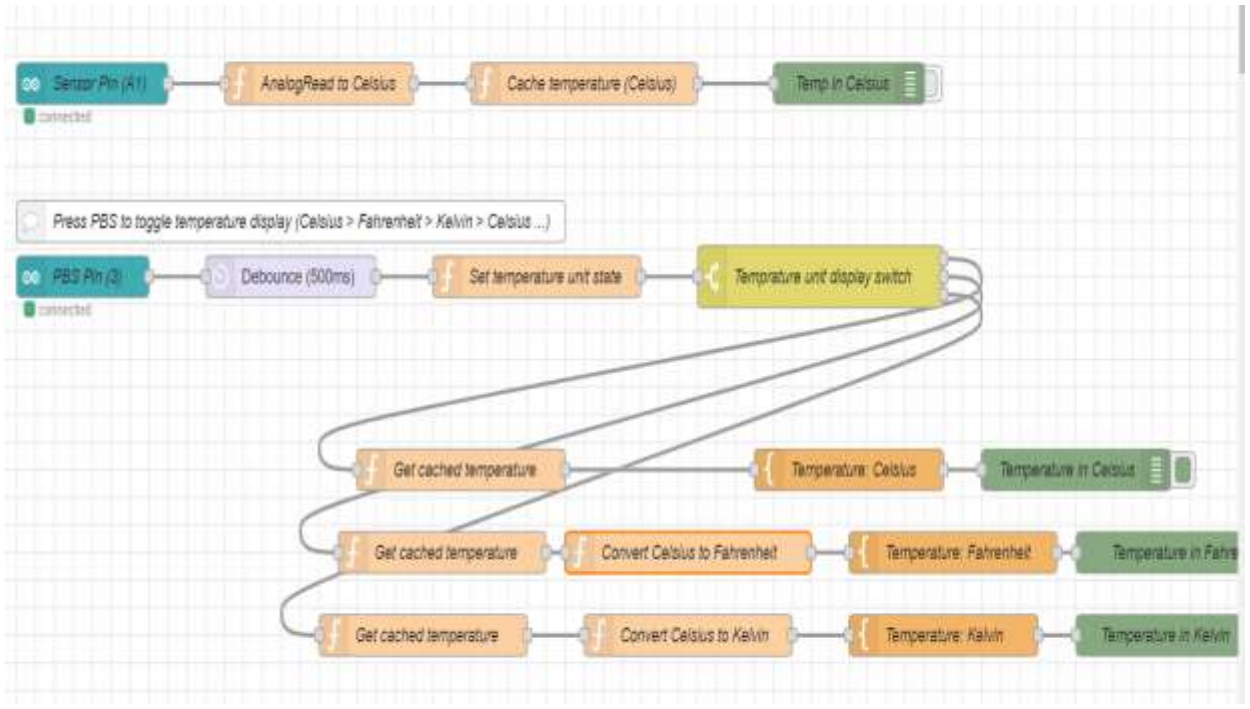
For *Dim Red*:

```
var redValue = flow.get('Red') || 255;
msg.payload = parseInt(Number(redValue) * (msg.payload / 100));
return msg;
```

Hybrid activity – 2:

In this hybrid activity you'll create a Node-RED flow that continuously reads a temperature sensor (i.e. a thermistor) and reports the temperature in Celsius, Fahrenheit and Kelvin with each press of a push-button switch (PBS).





Trouble shooting for not getting serial port for the teensy:
I had serial port issue: I resolved as follows…
https://serialport.io/docs/guide-installation

1. as I have already updated my vs so I did set it: npm config set msvs_version 2019
2. within .node-red folder, I just rebuild: npm rebuild.