TECHNOLOGY CAMP

DAY 4 : INTERNET-OF-THINGS

Controlling IoT Devices

Session 3

YELLOW CIRCLE INC
PO Box 2383
Elk Grove, CA 95759-2383

Teacher Lesson Plan

Session Name:

Controlling IoT Devices

Summary:

The Internet-of-Things allows for control of devices both from the cloud and through physical controls. In this lesson, we will learn how to control an LED through software and by building a manual button.

Time Allotment:

65 minutes

Learning Objectives:

- *Learn 3 methods to copy and reuse Node-RED flows*
- *Describe common nodes and their uses*
- *Build a manual control button using Node-RED*

Supplies:

- *Scrap paper / notepad to take notes*
- *Laptop / computer with Internet access*
- *Raspberry Pi and connected breadboard with pre-built circuits*

Learning Activities:

- **(3 minutes) - Session overview**

  The Internet-of-Things allows for control of devices both from the cloud and through physical controls. In this lesson, we will learn how to control an LED through software and by building a manual button.

  Students learn 3 methods to copy and reuse Node-RED flows, common nodes and their uses, and how to build a manual control button using Node-RED

- **(5 minutes) – Node Review**
    - *The inject node inserts (injects) a message, known as a payload into the flow. When first dropped on the workspace, the node defaults to timestamp, but edited to inject other payloads as selected from the drop down menu*
    - *The debug node displays either the message payload or the complete message object. The message can be output in to several different locations.*
    - *The rpi gpio output node (note the logo on the right) allows for message payloads (values) to control the state of GPIO pins to control IoT devices.*
- **(4 minutes) - Video : Node-Red switch node for beginners**

  https://www.youtube.com/watch?v=oRTb6bgGN_I
- **(3 minutes) – New Nodes**
    - *The function node allows for the addition of JavaScript code. Messages received by the function node can be processed by the JavaScript code and then return a message, message object(s), or nothing (ending the flow).*
    - *The switch node evaluates a received message based on a hierarchy of pre-defined rules. The switch can be set to stop at the first match or run through all the rules in the switch. Once the evaluation is complete, the switch routes message on based on the flow design.*

  Remind students that by dropping any node in the workspace area, clicking on it, and opening the info tab on the right, they can get additional help and information about the node and the node properties.
- **(6 minutes) - Video : How to Copy Nodes and Flows in Node-Red**

  https://www.youtube.com/watch?v=oRTb6bgGN_I

  Stop at 5:25
- **(3 minutes) - Lab: Raspberry Pi Set up**

(Confirm that all students have both VNC and Node-RED running and open on their laptops before proceeding. Follow these instructions only if any students shut down their Raspberry Pi at the end of the last lesson)

- *Please **do not** unplug or turn off the PI without shutting it down from the App Menu ( in the upper left corner). Doing so could corrupt the SD card.*
- *Open your laptop and double click on the VNC Viewer icon.* 
- *Login using the default credentials:*
- *UserID: pi*
- *Password: raspberry*
- *When the Raspberry Pi desktop appears, click on the VNC icon*  *in the upper right and note your IP address listed under Connectivity*

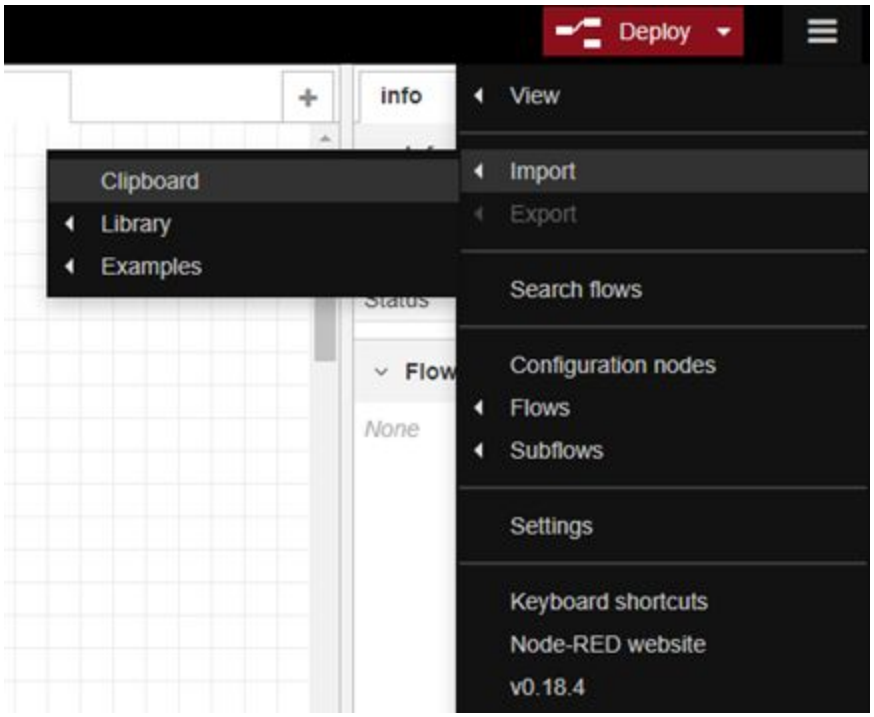
- *Open a new browser tab and enter **http://{Your IP Address Here}:1880** to start Node-RED*

- **(12 minutes) – Activity: Import, Review, Modify, and Test Blinking LED Flow**

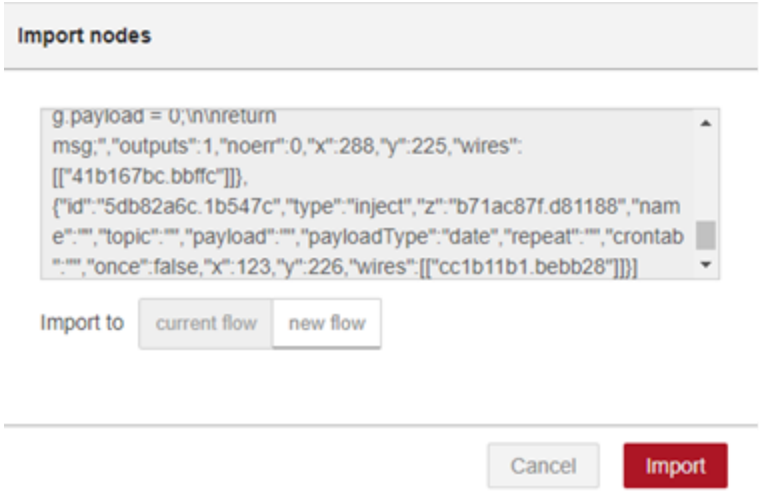
  So far we have built our flows from scratch, but in this activity, we will be importing a flow built by a developer named IgnacioRubio. The flow is written in JSON (JavaScript Object Notation). The original can be found at:

  https://flows.nodered.org/flow/3dbd62dfa8f6db78a598d0c7263f0f97

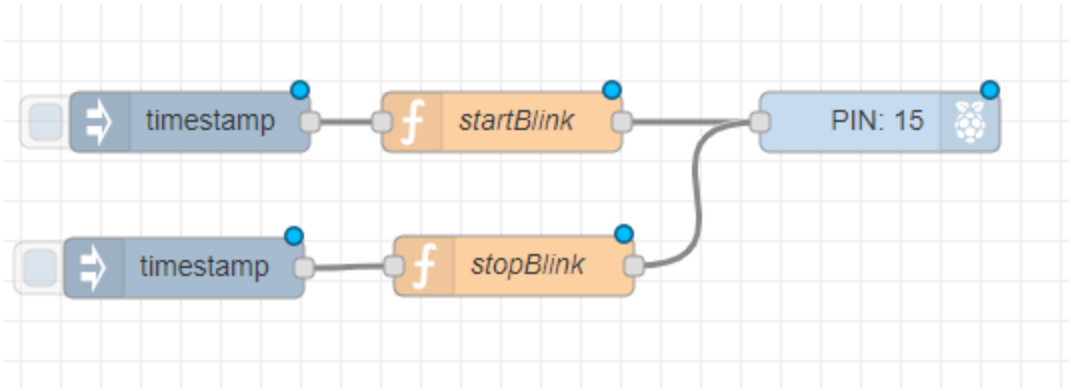
1. *Open the resources folder on your laptop, and then open the BlinkLEDFlow.txt file in a text editor.*
2. *Ctrl+A and Ctrl+C to copy the entire JSON to the clipboard*
3. *Open the Node-RED tab in your browser*
4. *Select the hamburger menu in the upper right>Import>Clipboard*



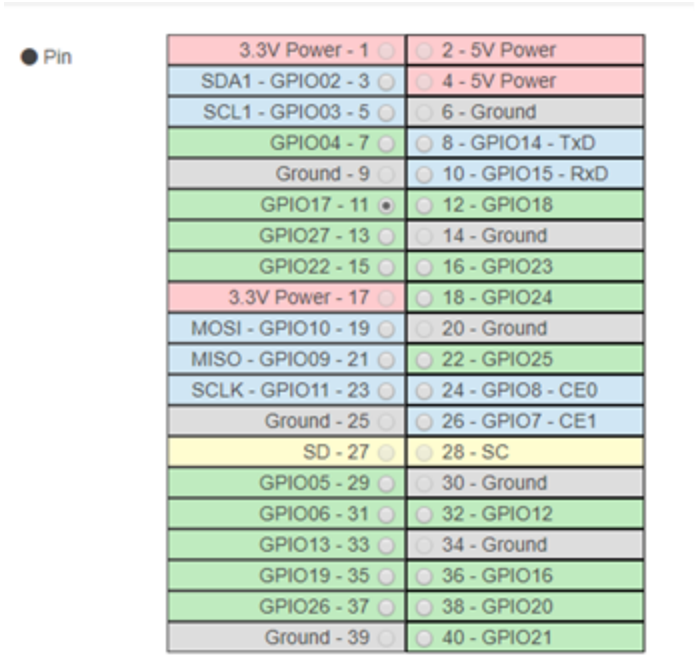
5. *Select the Import to "new flow" button, then Ctrl+V in the Paste nodes here area and select the Import button*



6. *The entire flow will appear in the workspace, select a location for placement and left click to place.*



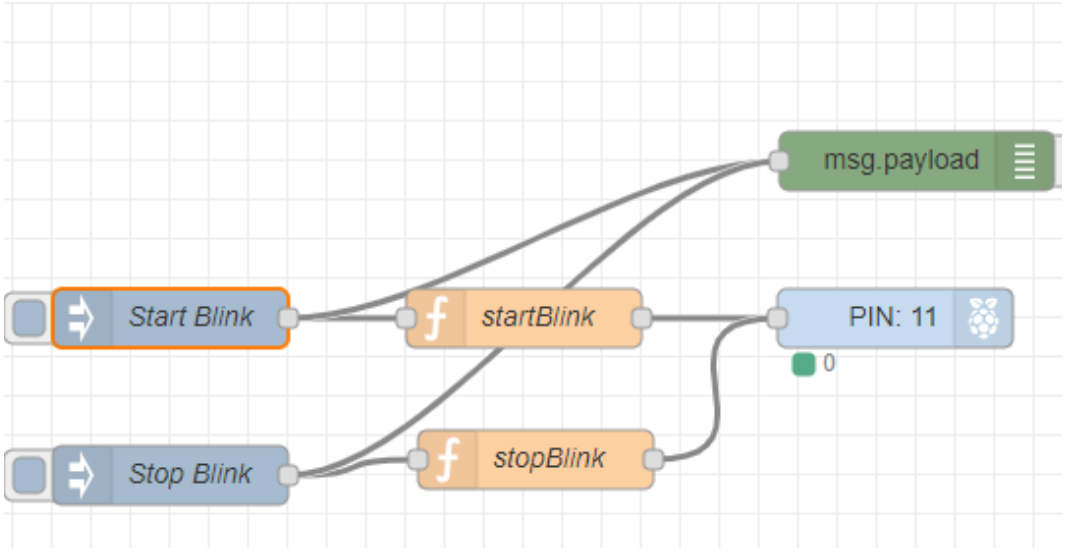
7. *Note that this flow is written for PIN:15 (this is GPIO22), but our LED is on GPIO17 (PIN:11), so let's double click on the PIN:15 node and correct this.*



8. *Just to be sure, let's double check there aren't any references that need to be updated in any of the other nodes by double clicking on each one. If everything is OK, just click cancel and move to the next one. Note: All of the other nodes are OK. You may want to take a moment to point out the code in the startBlink function node, and note how it uses alternating if, else statements with 1 and 0 payloads to achieve the blink.*

9. *Now we are ready to test. Click on Deploy our update to the GPIO pin, and then select the timestamp inject node connected to the startBlink function node. Confirm that the LED blinks, then select the timestamp inject node connected to the stopBlink function node to confirm that the blinking stops.*

10. *Now that we have confirmed that it works, let's make some modifications. Double click on the two inject nodes and give them more descriptive names (e.g. Start Blink and Stop Blink). Let's also add a debug node and connect it two the inject nodes we just renamed. Deploy, open the debug tab and note the results.*

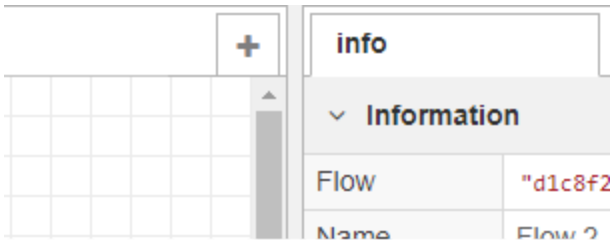


11. *Note the initial payload is a 1 and the LED is on, but not blinking. Now click Start Blink and note that the debug provides a timestamp number. Click on the number and it will scroll through several formats and filly display as a human readable timestamp.*

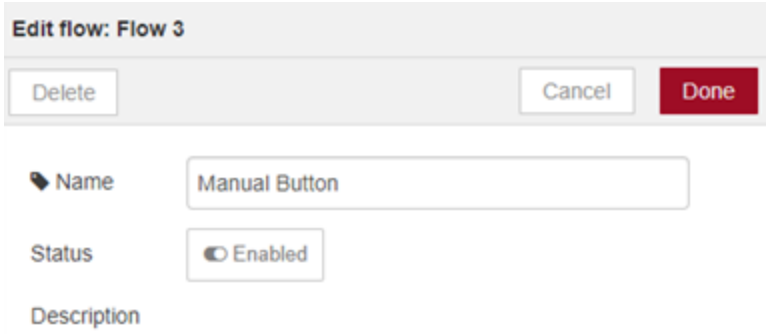
- **(12 minutes) – Activity : Import, Review, Modify, and Test Blinking LED Flow**

  We have been using software to control our Raspberry Pi devices in all of our activities so far, but in the IoT, many devices have manual controls. In this activity, we will use Node-RED to write code to use a manual button to control our LED.

  1. *Open the Node-Red tab in your browser, then click on the + symbol on the top near the info tab to open a new workspace.*



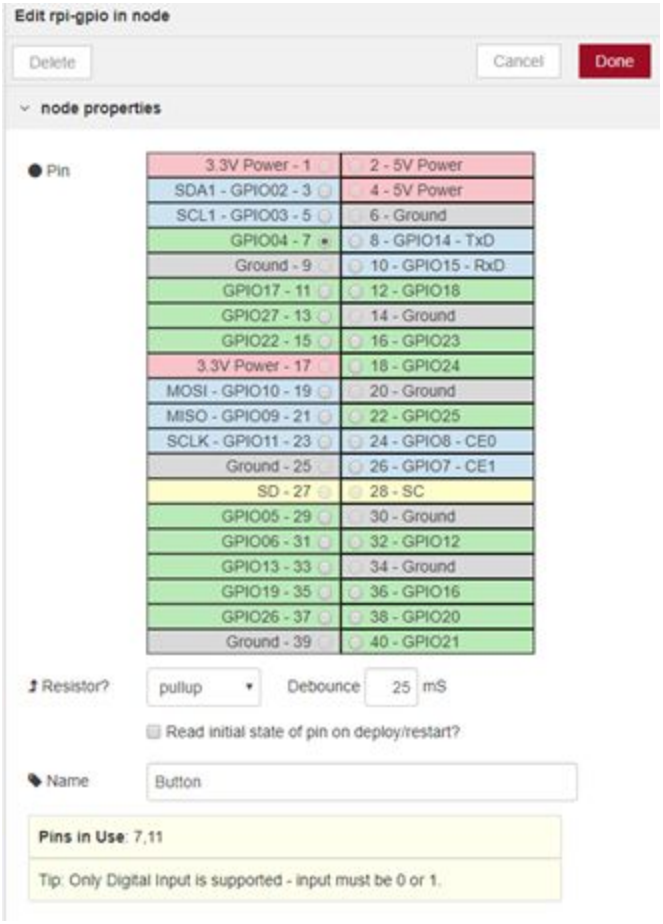
2. *Double click on the new workspace tab and give it a descriptive name (e.g. manual button) and select Done*



3. *Navigate to the tab from the prior lesson where you created the software buttons to control the LED. Select the rpi gpio node (in the example, Red LED) and Ctrl+C to copy. Then navigate back to your new tab (e.g. Manual Button) and Ctrl+V to paste. Select the location for the node and left click to set it on the workspace.*

   Note: Have the students double click on the node and confirm the settings are the same as in the prior exercise. (Click Cancel when done to return to the workspace.)

4. *Select an rpi gpio input node (note the logo on the left) and set it on the workspace. We will be using GPIO 4 (pin 7) for the button for this node. Set the properties as follows:*



5. *Explain that because the button pin is set to pullup, sending a 1 (ON) message to the LED as the default, and sending an 0 (Off message) only when pressed, a flow with the current configuration would operate in the reverse of a normal button.*

   Optional: If you have time, you may wire up the flow at this point, to demonstrate that the LED will be on at deployment and that pressing the button will turn it off. Remove the connections between the nodes by selecting each one and pressing delete and then proceed to the next step.

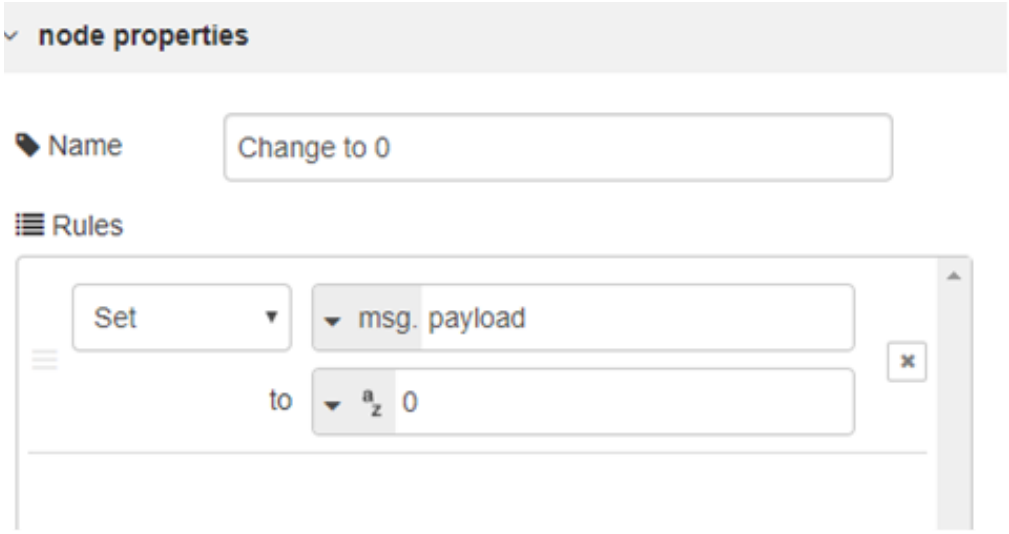
6. *We need to insert some switches to correctly process the messages so that the LED is off by default and only turns on when the button is pressed.*

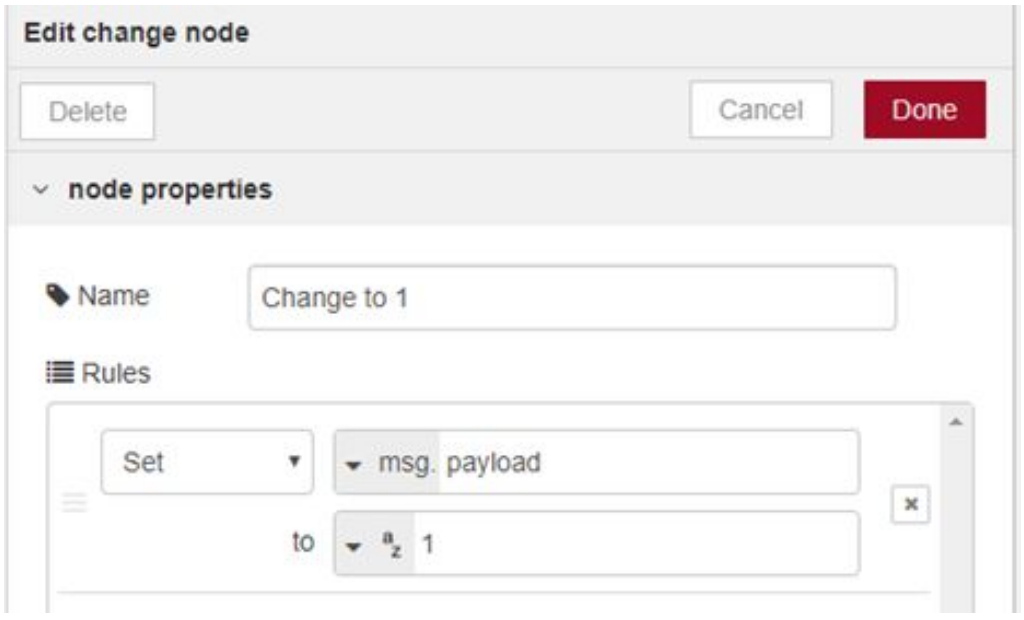
Select a switch node and drop it on the workspace. Configure the node properties as below and select Done.

Note: To add the second rule, select the +Add button to the bottom left of the rules area.

7. *You will see that the switch now has 2 paths, one if the payload is 1 and another for all other messages. We now will add 2 change nodes to modify the messages depending on the path from the switch node (Button Start)*

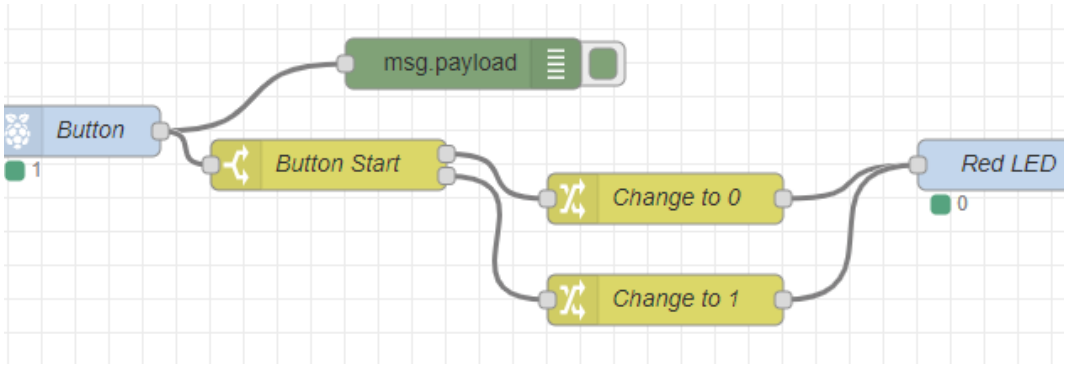
8. *Select and drop 2 change nodes on the workspace and configure as follows:*





9. *Now wire the flow so that If the payload is 1, it will change to 0 (connection), otherwise it will change to 1 (bottom connection). Connect*

the rpi gpio input node (Button) to the switch node (Button Start) and the 2 change nodes to the rpi gpio output node (Red LED) and deploy.



10. *The LED will now light when you press the button and turn off when the button is released.*
11. *At this point, we have several open tabs in our Node-RED, but we don't really need most of them. To demonstrate how to delete a tab, select the tab for the LED Blink flow and select the Delete button to remove it.*

- **(15 minutes) Try It!**
  - *Modify the flows you built.*
  - *Build your own flows.*
  - *Experiment with using different options and not the results.*
- **(2 minutes) Closing / Wrap-up**
  - *Learn 3 methods to copy and reuse Node-RED flows*
  - *Describe common nodes and their uses*
  - *Build a manual control button using Node-RED*
- **(1 minutes) - What's next?**

Inform students to head back to the cafeteria for snacks / break, and remind them to use the restroom before next session starts.