KENNESAW STATE UNIVERSITY

CPE 4040: Data Collection and Analysis, Spring 2024

Laboratory Report #5

Ultrasonic Sensor Application with Node-RED

Team Members: Anindita Deb and Damisi Kayode

Electrical and Computer Engineering

Kennesaw State University

Faculty: Dr. Jeffrey L Yiin

Date of Lab Session: March 3, 2024 and March 6, 2024

I. Objective

1. Learn how to interface and control an ultrasonic sensor.

2. Learn how to run Node-RED development tool on Raspberry Pi.

3. Learn how to design a user interface in Node-RED to read distance measurement.

II. Material List

Hardware:

- Ultrasonic sensor (SR-04)
- Red and Green LED
- Breadboard and resistors

Software:

- Remote Desktop Connection
- Node-RED Development Platform

III. Lab Procedures and Results

1. We plug the power adapter to Raspberry Pi and power up the system; open Remote Desktop Connection (or SSH connection) on our laptop and connect to Raspberry Pi.

Section 1: Setting Up Node-RED Development Platform

2. We open a terminal window and type node -v to see if Node-RED is already installed. It was not installed so we followed the link here to install the latest Node-RED: https://nodered.org/docs/getting-started/raspberrypi

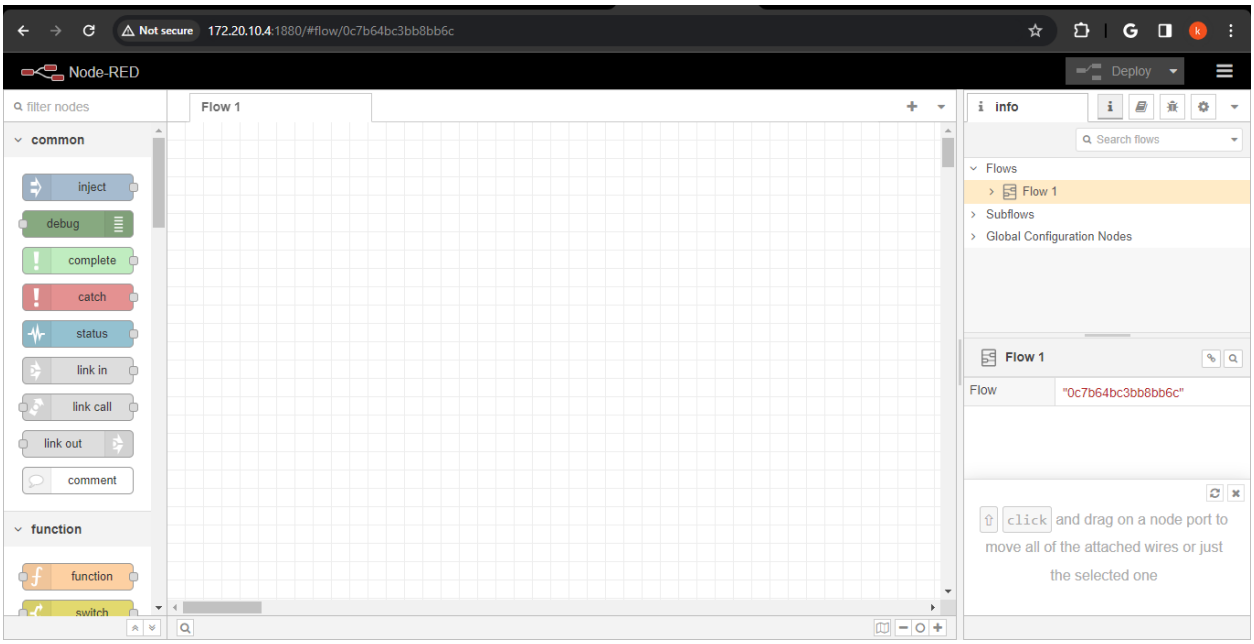
3. We start Node-RED by typing “node-red-start”.

```
damisi@dkayode: Node-RED console
File  Edit  Tabs  Help
On Pi Node-RED works better with the Firefox or Chrome browser

Use   node-red-stop                         to stop Node-RED
Use   node-red-start                        to start Node-RED again
Use   node-red-log                          to view the recent log output
Use   sudo systemctl enable nodered.service  to autostart Node-RED at every boot
Use   sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
4 Mar 10:30:22 - [info]
Welcome to Node-RED
===================
4 Mar 10:30:22 - [info] Node-RED version: v3.1.6
4 Mar 10:30:22 - [info] Node.js  version: v18.19.1
4 Mar 10:30:22 - [info] Linux 6.1.0-rpi7-rpi-v8 arm64 LE
4 Mar 10:30:23 - [info] Loading palette nodes
4 Mar 10:30:25 - [info] Settings file  : /home/damisi/.node-red/settings.js
4 Mar 10:30:25 - [info] Context store  : 'default' [module=memory]
4 Mar 10:30:25 - [info] User directory : /home/damisi/.node-red
4 Mar 10:30:25 - [warn] Projects disabled : editorTheme.projects.enabled=false
4 Mar 10:30:25 - [info] Flows file     : /home/damisi/.node-red/flows.json
4 Mar 10:30:25 - [info] Creating new flow file
4 Mar 10:30:25 - [warn] Encrypted credentials not found
4 Mar 10:30:25 - [info] Server now running at http://127.0.0.1:1880/
4 Mar 10:30:25 - [info] Starting flows
4 Mar 10:30:25 - [info] Started flows
```

4. Opening a web browser window on our laptop and enter the following address: http://your RPi's IP address: 1880 (e.g., http://10.100.125.75:1880)

Note: Port number 1880 is a registered TCP/UDP port by Node-RED

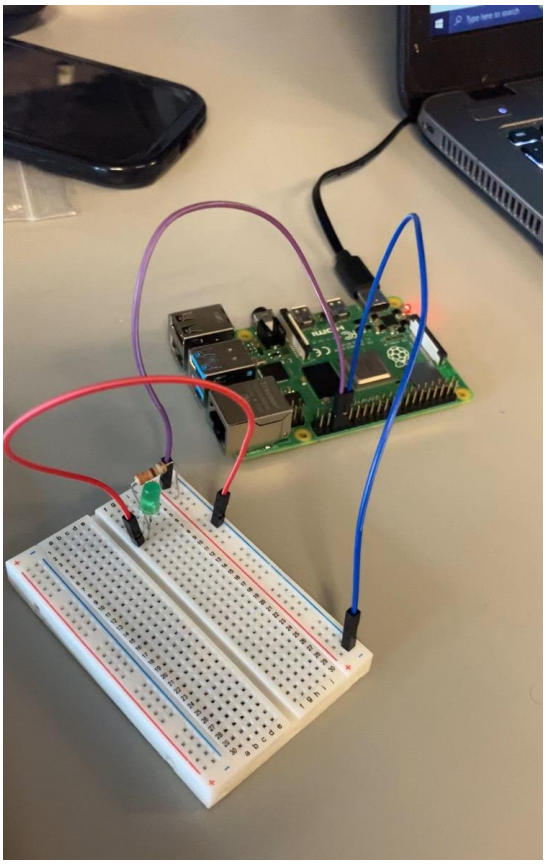
We should see a Node-RED development environment showing up in our browser.



5. From the menu button, we click on ‘Manage palette’. This allows us to modify the node palette menu on the left of our layout.

6. From this new menu we can see what nodes we currently installed and can install new nodes. Clicking on the ‘Install’ tab, we search and install the following packages: Node-RED-node-pisrf (This is for the ultrasonic sensor) Node-RED-dashboard (This installs a library of UI modules)

7. We connect a green LED with a serial (330Ω) resistor to GPIO16 (Pin No. 36) on the extension connector of your Raspberry Pi. We don’t forget to connect GND here.



8. In the node palette, we scroll down to find the rpi - gpio in and rpi - gpis out nodes. The first 2 node on the list, with the raspberry icon on the left, is for input. The second node, with the raspberry icon on the right, is for output. Turning on an LED is an example of output. We drag an output node to the working area in the middle.

9. Double clicking on the output node, a configuration box will pop up. We change the GPIO pin to GPIO16 and click Initialize pin state. Setting the initial pin state/level on low. We assign a name to the node (testLED). We click Done when finished.

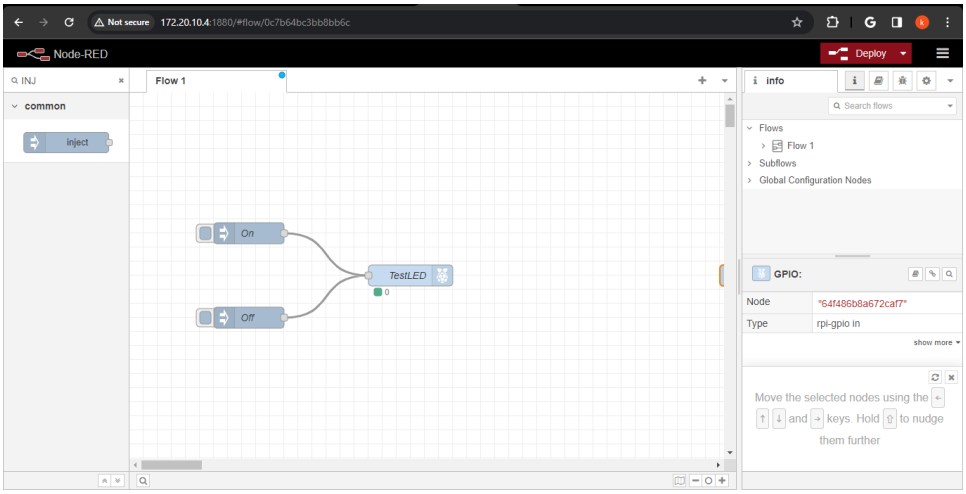
10. To turn the LED on and off, we need an input node. In Node-RED, we can inject messages into the flow that causes things to happen as a result. We drag an inject node onto the flow.

11. Double clicking on the inject node. We use the drop down next to Payload to change the data type to ‘string’ and type ‘1’ in the Payload box. We type ‘On’ in the Name box. Press Done when we’re finished.

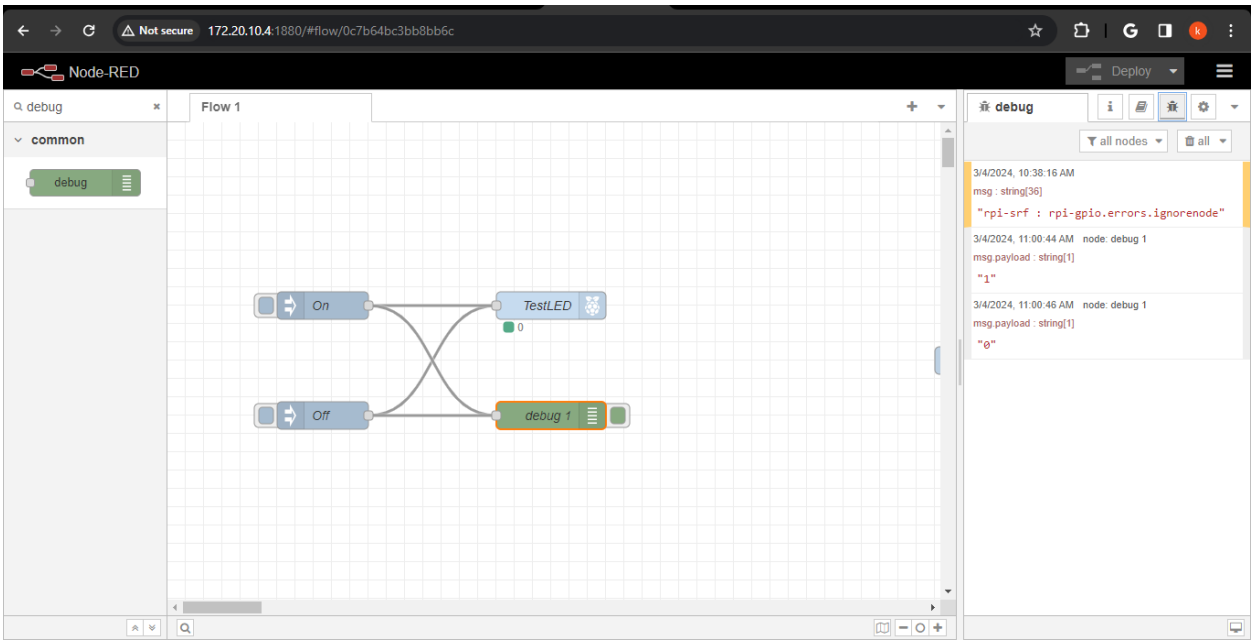
12. We repeat the previous step to create another inject node, however, typing ‘0’ as the Payload message, and named this node ‘Off'.

13. We click and drag the grey dot on the right side of the On node to the grey dot on the testLED node to connect them. Connecting the Off node as well. The flow should look like this now: Click on the red “Deploy” button on the upper righthand corner of the screen. A message, “Successfully deployed”, should pop up.

14. Clicking on the blue-grayish square on the left side of the On node to inject the message ‘1’. The testLED node receives the message and the LED will light up. We are able to turn the LED off by clicking on the blue-grayish square of the Off node, which injects the message ‘0’.

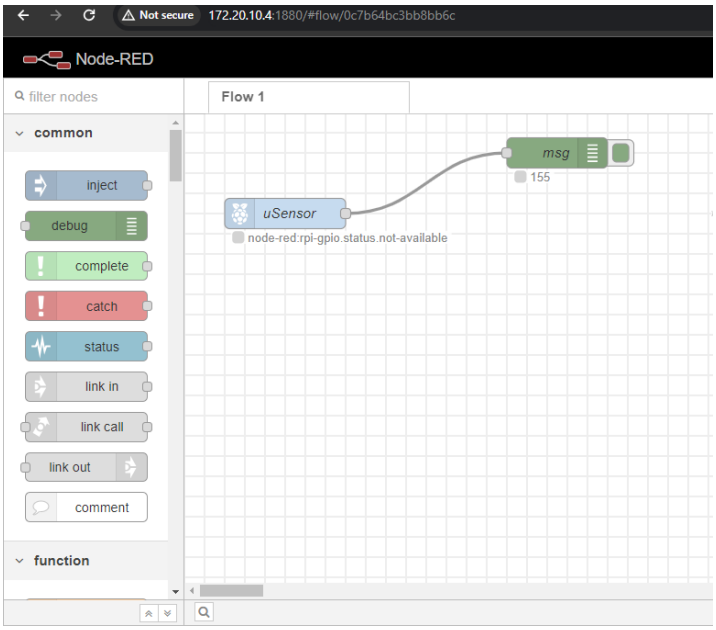


15. Node-RED can also display debugging information by wiring up the nodes to a debug node, which can be found in the node palette. We drag in a debug node and wire our two inject nodes to it, then click Deploy. When we click the blue buttons to inject the message, Node-RED will show what was injected in the Debug tab (one with a bug symbol) on the right side of the screen. We click the tab to display the messages.



16. We insert the Ultrasonic Sensor to the bread board. Connecting 5V, GND, Trigger and Echo pins to the Raspberry Pi connector. We start with GND and 5V. We connect Trigger input to GPIO23 (Pin 16) and connect Echo output to GPIO24 (Pin 18).

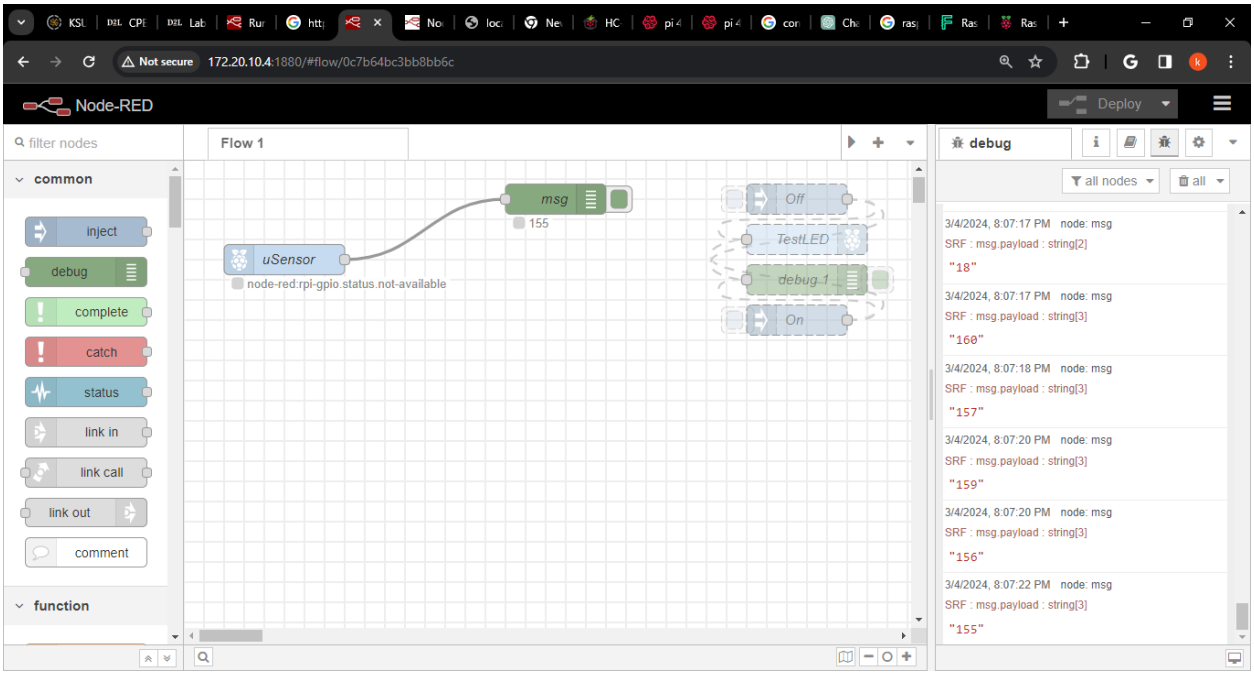
17. We scroll down the node palette menu and drag the rpi srf node and a debug node into the flow area. Then connect the nodes together.



18. Double clicking on the rpi srf node to open its configuration settings, we set Pins to “16, 18” (which correspond to GPIO 23 and 24). Next, we open the debug node configuration settings, checking the debug window and node status options. This will display the last sent message.

19. We deploy the flow and start to see numbers appearing below the debug node. These are distance measurements in centimeters. We can change the frequency of the measurements

within the rpi srf node. We can also see the entire measurement history in the debug tab.



Section 2: Design a User Interface for the Ultrasonic Parking Sensor

20. From the Node-RED menu, we scroll down to the dashboard section and drag a gauge node and a chart node. Double clicking on the gauge node and editing the Group name, as shown in the figure below. We double click on the chart node and make similar edits. We make sure to use gauge and chart from “dashboard", not “dashboard2".

21. We go back to the flow window, make node connections and deploy. To visualize the graphical user interface, we open another browser window and enter the following address:

http://your RPi’s IP address: 1880/ui

We see two graphs showing the distances obtained from the ultrasonic sensor. We try to put an object in front of the sensor, then move it back and forth to observe changes in the graphs.

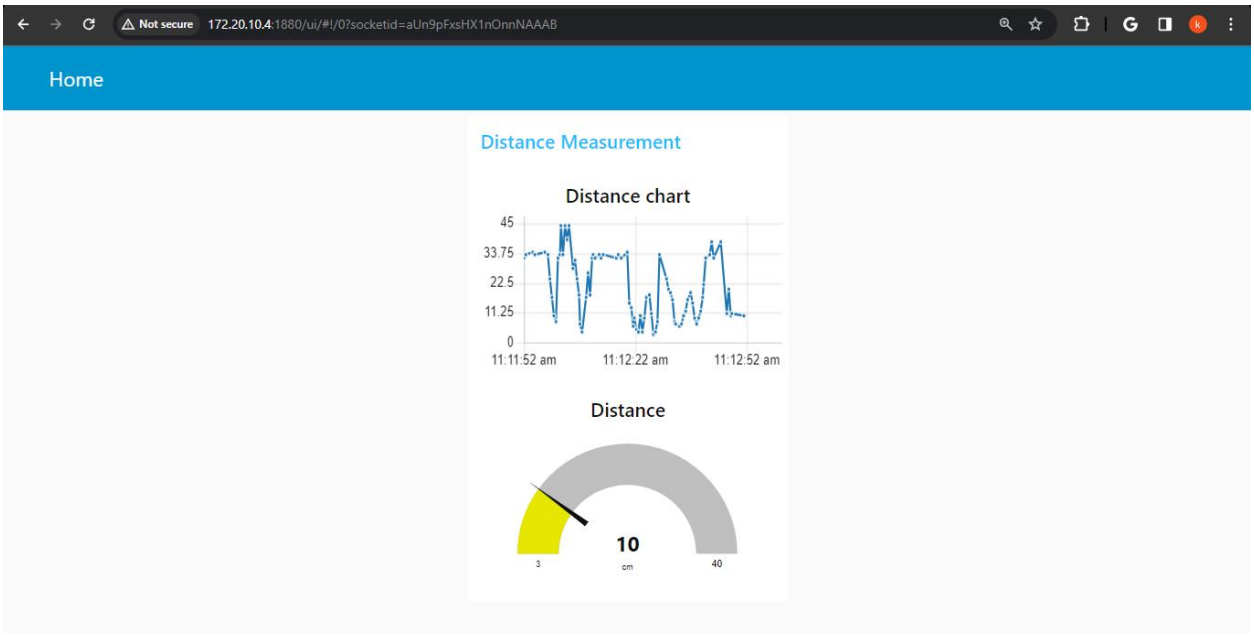
22. Next, we add a function node, two GPIO output nodes and two debug nodes to make the following connection flow:

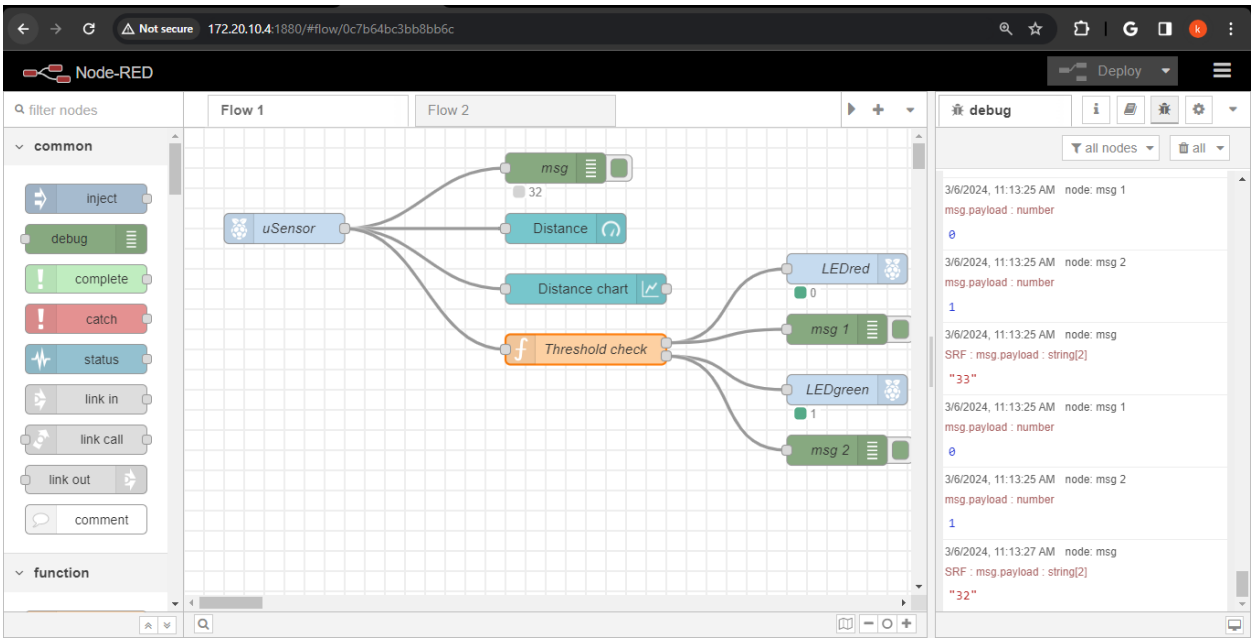
23. Double clicking on the function node, we configure the settings as follow:

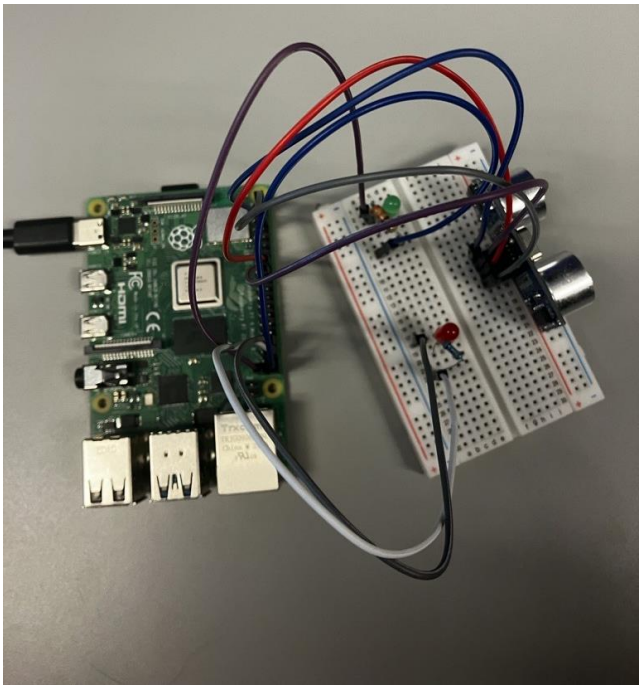
Enter the following codes in the Function block:

```
var msg1 = { payload: msg.payload.length };
var msg2 = {payload: msg.payload.length };
if (msg.payload<9) {
msg1.payload = 1;
msg2.payload = 0;
return [msg1, msg2];
} else {
msg1.payload = 0;
msg2.payload = 1;
return [msg1, msg2];
}
```

24. We enter the settings of GPIO nodes and change the names as LEDRed and LEDGreen with pin settings of GPIO16 and GPIO21, respectively. Also editing the debug nodes to change their names to msg1 and msg2. The code that we enter in the function node is to read distance value at the input and turn on the green LED if the value is greater than 15 cm and turn on red LED if the value is less than 15 cm. Deploy the flow and test it.


Not secure 172.20.10.4:1880/ui/#!/0?socketid=aUn9pFxsHX1nOnnNAAAB
Home
Distance Measurement
Distance chart
45
33.75
22.5
11.25
0
11:11:52 am
11:12:22 am
11:12:52 am
Distance
10
cm
3
40


Not secure 172.20.10.4:1880/#flow/0c7b64bc3bb8bb6c
Node-RED
Deploy
filter nodes
Flow 1
Flow 2
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
uSensor
msg
32
Distance
Distance chart
Threshold check
LEDred
0
msg 1
LEDgreen
1
msg 2
debug
all nodes
all
3/6/2024, 11:13:25 AM node: msg 1
msg.payload : number
0
3/6/2024, 11:13:25 AM node: msg 2
msg.payload : number
1
3/6/2024, 11:13:25 AM node: msg
SRF : msg.payload : string[2]
"33"
3/6/2024, 11:13:25 AM node: msg 1
msg.payload : number
0
3/6/2024, 11:13:25 AM node: msg 2
msg.payload : number
1
3/6/2024, 11:13:27 AM node: msg
SRF : msg.payload : string[2]
"32"



Conclusion

The outputs from that specific node in step 23 were unclear, and we couldn't understand why both LEDs were lighting up under the same conditions. As a recommendation for the next lab, it should be suggested to explicitly write in “change the output to 2” because we had difficulty finding that specific step. We also struggled to finding the correct dashboard for a while as well and fixed it by installing the correct dashboard instead of dashboard 2 after researching it a couple of times since it did not show in the top suggestion. Then lastly after changing conditions on the LEDs we ultimately “broke Node-Red”. It would load into the environment briefly before turning back off when typing in “node-red-start”. We had to load it from “safe mode” to readjust the values of the LEDs preventing the crashing.