

© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Learning Data Engineering: Creating IoT Apps using the Node-RED and the RPI Technologies

Zenon Chaczko¹, Robin Braun²

^{1,2} School of Computing & Communications, Faculty of Engineering & IT, University of Technology, Sydney – 2007, NSW, Australia
Emails: ¹ zenon.chaczko, ² robin.braun@uts.edu.au

Abstract – *This paper demonstrates the suitability and the practicality of using the advanced open source tools such as the Raspberry Pi and the Node-RED for teaching and learning in the Internet of Things (IOT) subject within a newly created major of Data Engineering in the Faculty of Engineering and IT at University of Technology, Sydney. Understanding and practicing of the Internet of Things largely depend on the high availability of tools, their low cost, and ease of use that can accelerate learning processes. This paper demonstrates relatively uncomplicated practical lab exercises involving the Raspberry Pi hardware, firmware and the Node-RED programming environment that students can execute to stimulate their learning, understanding of the Internet of Things technology and acquire fundamental data engineering skills.*

Keywords-*IOT Apps, RPI hardware, Node-RED IDE, Data Engineering*

I. INTRODUCTION

The goal of Node-RED programming environment [1] is to empower system developers and engineers to make programming as easy as possible by wiring together graphical blocks called nodes. In practice, this software tool allows even the beginners to rapidly start developing Internet of Things (IOT) applications [2]. The Node-RED Integrated Development Environment (IDE) was initially developed as an open source at IBM in 2013 [3]. The IDE allows to quickly connect a broad range of hardware devices (including Raspberry Pi microcomputers) and software to Web Services [2]. Relatively quickly, the Node-RED become a general-purpose Internet of Things (IOT) programming tool [4]. At present, the Node-RED has attracted a very active community of developers and supporters who are contributing to new software components (nodes). A growing list of opens source software components allows creating even very advanced applications very quickly, providing at the same time a very visual and stimulating learning environment.

A. Node-RED programming environment

The goal of Node-RED programming environment is to empower system developers and engineers to make programming as easy as possible by wiring together graphical blocks called nodes. In practice, this software tool allows even the beginners to rapidly start developing Internet of Things (IOT) applications. The Node-RED Integrated Development Environment (IDE) was initially developed as an open source at IBM in

2013. At present, the Node-RED has attracted a very active community of developers and supporters who are contributing to new software components (nodes). A growing list of opens source software components allows creating even very advanced applications very quickly, providing at the same time a very visual and stimulating learning environment.

And what user can do with? As mentioned earlier, the Node-RED IDE is a very effective tool for creating IoT-based applications. These applications could involve various scenarios of data acquisition, transportation, monitoring, and analytics. A group of individual students, professional developers or engineers may decide to develop applications (apps) that allow looking after their home or working environment remotely. Another group of developers may want to be aware of the new trends in their social networking applications (i.e. Twitter or Facebook). The Node-RED seems to be an ideal tool to create a range of similar applications and in an easy way. The Node-RED, at the very first instance, can be used by programming beginners. However, like many other similar tools that pretend to be novice oriented tools, rather quickly require for users to develop some more skills in programming. In our case, one will probably need to learn JavaScript language to create some more meaningful Node-RED applications. Being a visual and very powerful tool, the Node-RED aims to create a broad range of the Internet of Things applications that involve various hardware components and other types of applications in a flexible manner.

This paper elaborates on relatively simple applications targeting learning students. However, users can also rather quickly develop useful applications that might surpass all the examples shown in this work. Furthermore, another aim of this paper is to show how powerful and useful the Node-RED tool can be. An interesting feature of Node-RED is that each flow/program component can be shared with other users/developers by import/export functions which permit to copy and paste the generated code into the user workspace. At the end of each presented example, readers will find the code snippets so that these can be copied from a given workplace and added with as many modifications or improvements as required. Some of these customized Node-RED nodes cannot be found

among the node created by the community. In this paper, we demonstrate examples of what can be done by using the Node-RED, the Raspberry Pi embedded environment, and some other external tools.

B. The Raspberry Pi

The Raspberry Pi (RPI) board is an inexpensive, credit-card sized computer [5, 6] that can be interconnected with a computer monitor, a standard keyboard and a mouse. This small yet very capable device allows users of all ages to learn about computing, and to explore how to develop programs, browse the internet, play high-definition video and games, make spreadsheets, and do word-processing. Additionally, the Raspberry Pi can connect and interact with the outside world, and has been used in a wide range of electronic or digital maker projects, from music boxes and children trackers to weather stations and aquariums and birdhouses equipped with infra-red cameras. The original aim of the Raspberry Pi designers was to provide a technology for kids and adults all around the world to learn how to program and understand how computers work. The RPI device can do everything one would expect from a desktop computer to do. This may include a very diverse list of applications that can be created when using this tool. Once a user becomes familiar with using the device and the Node-RED environment, it won't take too long for him/her to develop first IoT applications. For teaching and learning purposes, we demonstrate some very basic examples of possible applications only. The main aim is to help potential adopters of the technology to understand how the tool works. The usage scenario presented in the next section show how we could send emails to ourselves and receive notifications using a mobile phone.

II. CREATING A FIRST IOT APPLICATION

The Node-Red is an easy out-of-the-box development environment, however, initially it can be a little bit too tedious to start, run and stop. It is recommended to use a simple start-up (init.d file) script that can do all the mandane work for the user. A sample script for starting, stopping and restarting services in Linux can be found at [7].

A. Creating The Sending an Email Application

For a first IoT application using the Node-RED the students may consider to develop a program that sends an automatic email using RPI Linux box. For this task, the easiest is to use a Google's Gmail account, although other email services can also be used, a give server can host a given user account. In this first example, the application is going to send an email each time the left button of the mouse connected to RPI's board is pressed by the user. This scenario might not be too useful; however, it allows students to learn how to use the Email Node component of the Node-RED.

1) Wiring the Flow

Once an instance of Node-Red IDE is running (in the RPI environment) the user needs to locate the required

nodes under input menu on the left of the editor panel (Fig 1.). The required nodes are in the first column under the filter nodes tag. If a user hover over any of the displayed nodes, a brief description will be shown that explains its functionality. For a simple " Sending an Email " application involves a design of a flow of interconnected nodes such as: an input node - *Pi Mouse Left*, *Email Sending* node and *Email Sending Test* node. These nodes need to generate an email send action when the left button of the mouse is clicked on Raspberry Pi. As soon as the left button is clicked, the Raspberry Pi will send a message where the body is *msg.payload* and the name of the message is *msg.topic*. The execution of the flow is shown in Fig 3.

In the Edit e-mail node, the developer needs to fill *userid* and *Password* with own Gmail account id and with an accurate detail of a person who is going to receive the email.

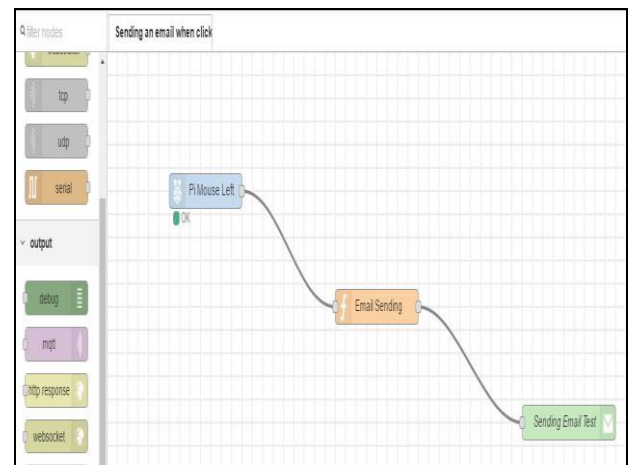


Figure 1: Node-RED editor. A wired flow of nodes.

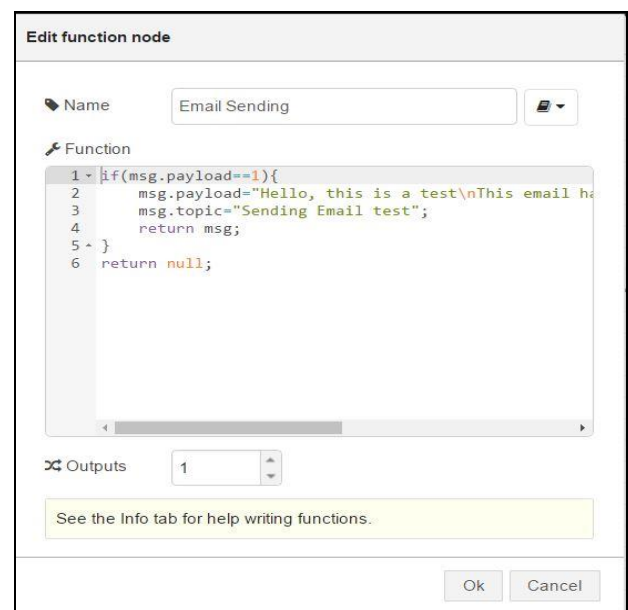


Figure 2: *Email Sending* node in Edit mode.

```

[{"id":"a755e00b.fcb9c","type":"rpi-
mouse","z":"3e4e2c63.192cc4","name":"","butt":"1","x":190,
y":120,"wires":[["87b713bf.e2b56"]]},{"id":"f432197a.617028"
type":"e-
mail","z":"3e4e2c63.192cc4","server":"smtp.gmail.com","port
":"465","name":"zenon.chaczko@uts.edu.au","dname":"Sendin
Email
Test","x":890,"y":320,"wires":[],"id":"87b713bf.e2b56","typ
":"function","z":"3e4e2c63.192cc4","name":"Email
Sending","func":"if(msg.payload==1){\n msg.payload=\\\"Hello
this is a test\\nThis email has been sent automatically from
Raspberry Pi\\nstudentAB\\\";\n msg.topic=\\\"Sending Email
test\\\";\n return msg;\n}\nreturn
null;","outputs":1,"noerr":0,"x":520,"y":220,"wires":[["f4321
7a.617028"]}]}

```

Figure 3: Flow Description.

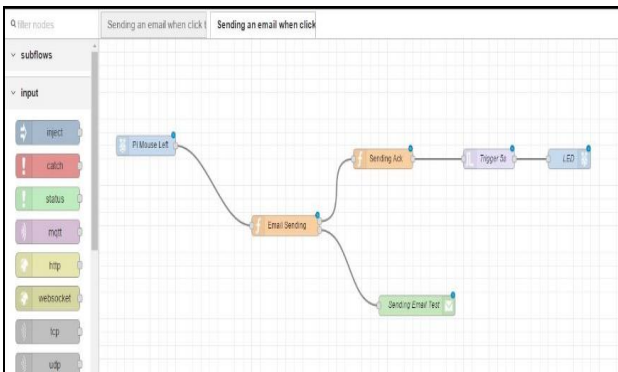


Figure 4: Flow of Sending an Email and Blinking LED.

2) Wiring the Flow From Sending an Email and Blinking a LED when Sending

As an extension, one may want to consider adding a visual confirmation feature that activate a blinking LED once the email is correctly transmitted. In this version, of the application a LED turns on while the e-mail is sending. The application requires, a couple of new functions to be added (developed) to the flow that confirm that an email has been sent (Fig 4). If the function receives a non-null message from the corresponding node, then it returns a high level which make the LED turns on for 5 seconds. This application requires three additional modules: the *Sending Ack* node (Fig. 5), the *Trigger 5s* node (Fig. 6) and the *LED* node.

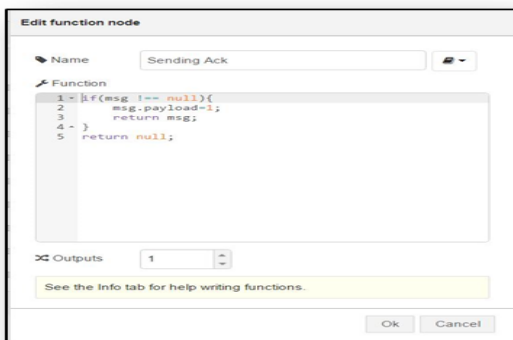
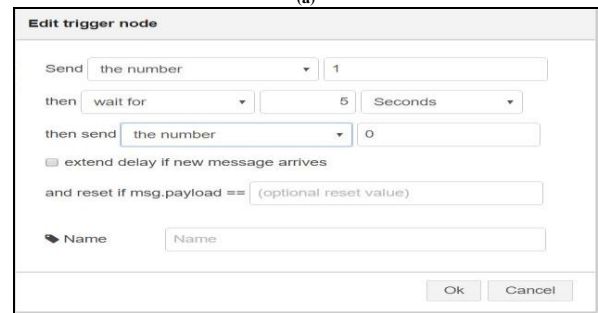


Figure 5: Node “Sending Ack”.



(a)



(b)

Figure 6: Trigger 5s Node, an icon (a) and Edit trigger node panel (b).

The *Trigger* node can be useful, in this case, to blink the LED. In fact, to let the user to watch the LED flashing, the LED need to be set high for a few seconds (5 seconds). To allow *Trigger* node to work properly there are the following few things that need to be set up:

- What will a reply when a given node receives a message. One can select a string, a number, the existing `msg.payload` or do nothing. In this example a number ‘1’ is sent
- Then, the action must be defined. A choice is to wait for a configured moment or wait until the node receives a reset message.
- There are other parameters available. These can be ignored in this example.

The *LED* node is not the original name of the node as it cannot be found in the standard menu of the Raspberry Pi dedicated nodes (Fig 7). For this example, *RPI GPIO* is selected as the developer needs to use the RPI pins as output using the *Edit rpi-gpio out* node (Fig. 8). In *GPIO Pin*, user should select which pin needs is to be used. Refer to the *GPIO Header* (Fig. 9). Some pins of Raspberry Pi are reserved for specific action (see Fig. 9). The pins that are free to use are marked in green. One needs to choose two of them. One to connect the Anode (+) of the LED and the ground to connect the Cathode (-) of the LED. It is up to the user to choose which one is preferred. Then, the user has to decide how to use this pin. If the *Digital Input* pin is chosen then only switch *on* and off the LED function can be implemented, if however, the *PWM output* is selected then the brightness of the LED can be defined by giving a number between 0 and 100, where 100 corresponds to the highest brightness. Once everything is set up as shown (Fig 9), a user need to deploy the flow on his device and verify that all is working and that a new email is receive in the mailbox.

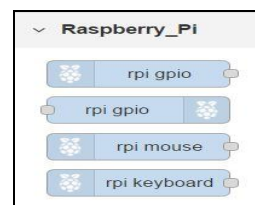


Figure 7: The Raspberry Pi menu of available nodes.

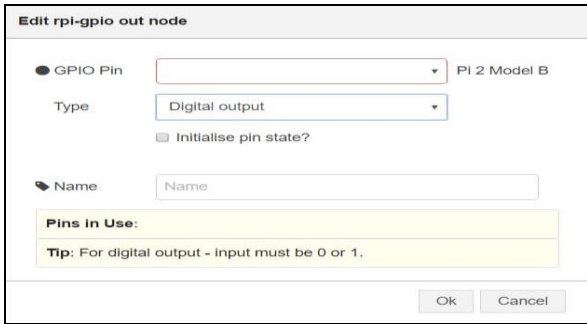


Figure 8: RPI GPIO node panel.



Figure 9: The GPIO Header

```
[{"id":"a755e00b.fcb9c","type":"rpi-
mouse","z":"3e4e2c63.192cc4","name":"","butt":"1
","x":110,"y":140,"wires":[["87b713bf.e2b56"]]},{"id
":"f432197a.617028","type":"e-
mail","z":"3e4e2c63.192cc4","server":"smtp.gmail.co
m","port":465,"name":"011197@student.ust.edu.a
u","dname":"Sending Email
Test","x":750,"y":380,"wires":[]},{"id":"87b713bf.e2
b56","type":"function","z":"3e4e2c63.192cc4","nam
e":"Email Sending","func":"if(msg.payload==1){\n
msg.payload=\\Hello, this is a test\\nThis email has
been sent automatically from Raspberri
Pi\\nstudentAB\\n msg.topic=\\Sending Email
test\\n return [msg, msg];\n}\nreturn [ null, null
];","outputs":2,"noerr":0,"x":440,"y":260,"wires"
:[["d18c0bd3.ec14e8"],["f432197a.617028"]]},{"id":"d
18c0bd3.ec14e8","type":"function","z":"3e4e2c63.19
2cc4","name":"Sending Accused","func":"if(msg !=
null){\n msg.payload=1;\n return msg;\n}\nreturn
null;","outputs":1,"noerr":0,"x":690,"y":160,"wires"
:[["ff55c568.4fac18"]]},{"id":"3cc62380.57b82c","typ
e":"rpi-gpio
out","z":"3e4e2c63.192cc4","name":"LED
","pin":15,"set":true,"level":0,"out":"out","x":
110,"y":160,"wires":[]},{"id":"ff55c568.4fac18","typ
e":"trigger","z":"3e4e2c63.192cc4","op1":1,"op2"
":0,"op1type":"val","op2type":"val","duration":5
,"extend":false,"units":"","reset":"","name":"Tri
gger
5s","x":920,"y":160,"wires":["3cc62380.57b82c"]}]]
```

Figure 10: Flow execution of sending an email and blinking a LED when sending

III. IOT APPLICATION WITH A BLUETOOTH SENSOR

A. A Set-up Stage

There are a few Bluetooth devices that allow to obtain data such as: temperature, humidity, movement (accelerometer), etc. In our experiment, CC2650SensorTag is used from TI device to demonstrate how it can work within the node-RED environment. There are a few preparation steps to execute prior to the application development. First, on the RPI, in a command terminal, one should install a Bluetooth Driver by tapping in prompt the following commands:

```
sudo apt-get install libbluetooth-dev libudev-dev pi-
bluetoothsudo setcap cap_net_raw+eip $(eval readlink
-f `which node`)
```

The above commands download the files to install the Bluetooth Driver needed to communicate with the Bluetooth Device. After this step, the user must go to the Node-RED user repository: ~/node-red and run the following command:

```
npm i node-red-node-sensortag
```

This command inserts a new node called sensorTag in the editor (in the node bar) when working with the Node-RED on a computer.

B. Using the new inserted node

When you insert the new node is inserted in the flow, a click on the item brings an Edit display panel [12] that allows to select categories of sensing data (i.e. temperature, pressure, luminosity, etc.) provided by the device.

C. Obtaining the data from the SensorTag

The easiest way to see what the node returns is to wire a debug node to our sensor Tag node (Fig. 13). In this example, only information concerning "Temperature/Humidity" and "Button pressed" have been selected. The debug window displays the content of the SensorTag message (Fig. 14).



Figure 11: The SensorTag node icon.

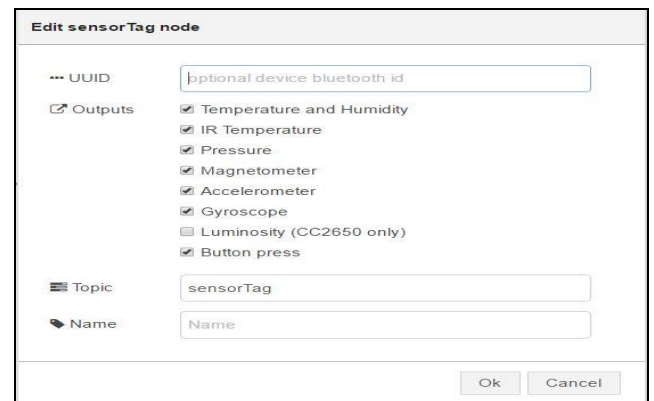


Figure 12: SensorTag Edit panel.

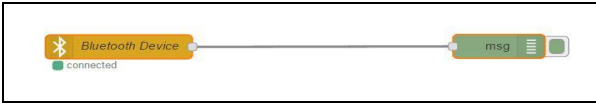


Figure 13: SensorTag wiring.

```

6/6/2016 12:15:11 c8a0a8aa-d45b058
sensorTag/humidity : msg : Object
{ "topic": "sensorTag/humidity", "payload": {
  "temperature": 29, "humidity": 21.1 },
  "_msgid": "73179232.8ce86c" }
6/6/2016 12:15:12 c8a0a8aa-d45b058
sensorTag/humidity : msg : Object
{ "topic": "sensorTag/humidity", "payload": {
  "temperature": 29, "humidity": 21.1 },
  "_msgid": "32b9a5d5.cd465a" }
6/6/2016 12:15:13 c8a0a8aa-d45b058
sensorTag/keys : msg : Object
{ "topic": "sensorTag/keys", "payload": {
  "left": true, "right": false, "magnet": false },
  "_msgid": "a73026d5.58cfd8" }
6/6/2016 12:15:13 c8a0a8aa-d45b058
sensorTag/keys : msg : Object
{ "topic": "sensorTag/keys", "payload": {
  "left": false, "right": false, "magnet": false },
  "_msgid": "b7541ae9.48abe8" }
6/6/2016 12:15:13 c8a0a8aa-d45b058
sensorTag/humidity : msg : Object
{ "topic": "sensorTag/humidity", "payload": {
  "temperature": 29, "humidity": 21.2 },
  "_msgid": "f1712f79.0e8ed" }

```

Figure 14: The SensorTag debug message.

```

[{"id":"ae776bd4.56a7f8","type":"sensorTag","z":"
2227dad1.821e26","name":"","topic":"sensorTag","
uuid":"","temperature":false,"humidity":true,"pres
sure":false,"magnetometer":false,"accelerometer":fal
se,"gyroscope":false,"keys":true,"luxometer":false,"
x":180,"y":300,"wires":[["a19bc146.6f1c5"]]},{"id":
"a19bc146.6f1c5","type":"debug","z":"2227dad1.82
1e26","name":"","active":true,"console":"false","co
mplete":"false","x":680,"y":300,"wires":[]}

```

Figure 15: Flow Description of the SensorTag wiring

For each sensor, the node “Bluetooth Device” sends an object message which contains such fields as: topic (which sensor the data relates to), payload (includes records of several sensor measures and specifies which number is related to which measure).

D. Using several Bluetooth devices

For using several Bluetooth devices in the same area, users will need to get the UUID. This is, in fact, the MAC address of the device in use. Users need so see what are the devices connected to the RPI. To verify this, in the command terminal, at the current folder, one needs to write the following command:

```
hcitool lescan
```

On the user prompt, it should be indicated that RPI is searching for connected devices. As soon as the SensorTag devices is in this list one needs to stop running the command. Then one needs to copy the MAC address which in our case is: 80:b4:48:C8:DC:86. Note, CC2650 SensorTag is the type of the device that was in use.

IV. CREATING THE IOT ENABLED ENVIRONMENTAL MONITORING SYSTEM WITH A GRAPHICAL UI

Node-RED has been created for people who have no skills in programming. Let’s see another powerful example. We are going to create a user interface in a very easy way.

A. Set-up Stage

Before creating an UI, you need to download some node in your raspberry PI. In order to do so, one needs to go in the created node-red folder (cd ~\.node-red), and tape the following command line in your command terminal:

```
npm install node-red-contrib-ui
```

Check on your browser. The following node must have been inserted in your node bar (Fig. 16)

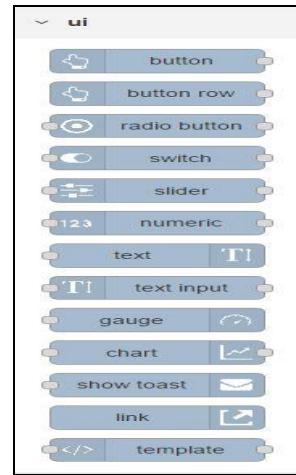


Figure 16: Node bar menu.

As can be seen, the previous installation has provided several UI nodes. Let’s discover how to use them now

B. A very first UI chart

Note: After downloading the UI package on your RPI, you have to open, in your browser a new page and tape: IP address of your RPI + “:1880/ui/” (Fig. 17).



Figure 17: IP address entry for RPI.

The User Interface we are going to create can be considerably complex as very big menu that can contain several tabs and each tab can contain several groups of components i.e., charts, button, gauge... (see Fig. 18). Maybe it is not clear for the moment but we will see few examples that should help you to understand. In order to see the power of Node-RED, we are going to create a very simple example: each time we click on the mouse button, we are going to generate a random number between 0 and 100 and display it on a graph and see what way this number changes.

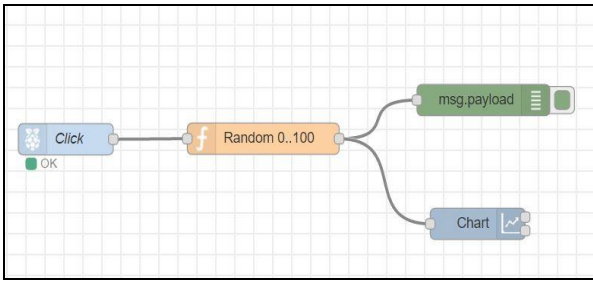


Figure 18: Definition of the node function.

Figure 19: The *Random* tab node function.

Figure 20: UI chart node selection panel.

Figure 21: UI node configuration selection panel.

Note: *Math* is a JavaScript Library and contains prebuilt functions (Fig. 19).

1. The node on the left is a RPI node, nothing new from what we have previously learnt.
2. The orange node in the middle is a function which, when the mouse button is pressed, returns an integer between 0 and 100
3. We are also using a debug node to have a direct look at the integer the function returns.
4. Finally, we have inserted the *chart UI node* (Fig. 20). This node must be configured as following:

- **TAB:** As described before, the interface currently created can have several tabs. In this field, one must choose in which tab the chart is to be inserted. If there was no tab being created before, one needs to create the first one. In order to do it, a user must click on the small pen icon on the right of the first line in the chart node panel. Once the pencil icon is clicked on, a new window opens (Fig. 21). Users need to provide the name to the UI chart node tab. Several tabs can be created, and users can choose in which positions these tabs are to be placed in. This is done by putting a newly created tab in the field position. At this stage, when only one tab is used the default order position is 1.
- **NAME:** user must provide the component's name that is being currently created. If a chart is to be created we can call it a Chart. Users might decide to call it "0 to 100 random Chart".
- **GROUP:** as mentioned before, in the new tab that was just created, one can add several group. For simplicity, one can just select the group wanted to be inserted in the chart. If there is a need for several group in the tab, then a user must also choose its individual position.
- **OLD AFTER:** Here, one should choose how long the data input must remain in the chart. Once the period is over, the data will be deleted and replaced by the new set of data. This aims to define data periodicity and a determined scale.

C. Results

Once you've followed all the step, open a new page on your browser and see what's going on when we press the left button on the mouse. The same User Interface display data charts can be seen when using a smartphone or a desktop computer (Fig. 22). In fact, one can enter the same address when using the desktop computer.

A. Display SensorTag information

Let's do something more useful now. We are going to read information from the Texas Instrument SensorTag that we studied before and display this information in the UI. One more time, nothing complicated.

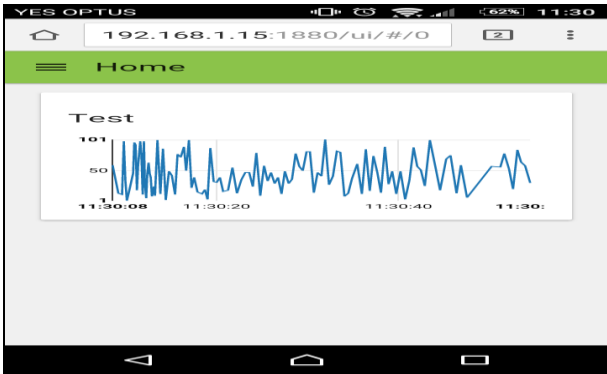


Figure 22: Display SensorTag information

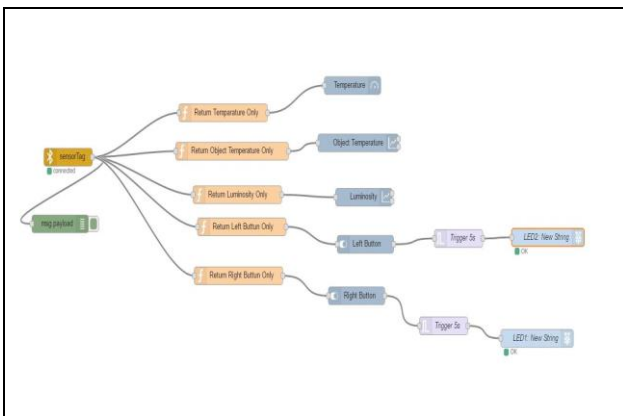


Figure 23: Flow description

```
[{"id":"6f0eae08.74ba7","type":"ui_tab","z":"8edb8f02.e30bc","name":"Home","icon":"dashboard","order":"1"}, {"id":"371e3dda.b4b8e2","type":"function","z":"8edb8f02.e30bc","name":"Random 0..100","func":"if(msg.payload){\nmsg.payload = Math.round(Math.random()*100)+1;\nreturn msg;\n}\n\nelse\nreturn\nnull;","outputs":1,"noerr":0,"x":260.9630126953125,"y":184.9801025390625,"wires":[[{"id":"751f8a17.43eb24","5b561aa1.f0b3f4"}]], {"id":"751f8a17.43eb24","type":"debug","z":"8edb8f02.e30bc","name":"","active":true,"console":"false","complete":"false","x":501.9602508544922,"y":136.9232940673828,"wires":[]}, {"id":"c1087c6e.6917c","type":"rpi-mouse","z":"8edb8f02.e30bc","name":"Click","butt":"1","x":80.95738220214844,"y":259.4090881347656,"wires":[[{"id":"371e3dda.b4b8e2"}]], {"id":"5b561aa1.f0b3f4","type":"ui_char","z":"8edb8f02.e30bc","tab":"6f0eae08.74ba7","name":"Chart","group":"Test","order":1,"interpolate":"linear","nodata":"NoData","removeOlder":1,"removeOlderUnit":"60","x":472.96022033691406,"y":274.53692626953125,"wires":[[],[]]]
```

Figure 24: Flow description report message

V. THE WEATHER INFO SCENARIO

Node-RED, thanks to Open Weather Map includes a node which can receive a life-feed of the current weather data. This allows students to develop an application that sends a notification (mail or SMS) about the weather conditions before the users leave their homes (Fig 23) . A message may provide an advice to take an umbrella, if it is raining or a hat, if it sunny. In order to receive the life-feed of weather data a user may sign up with the Openweathermap service [8].

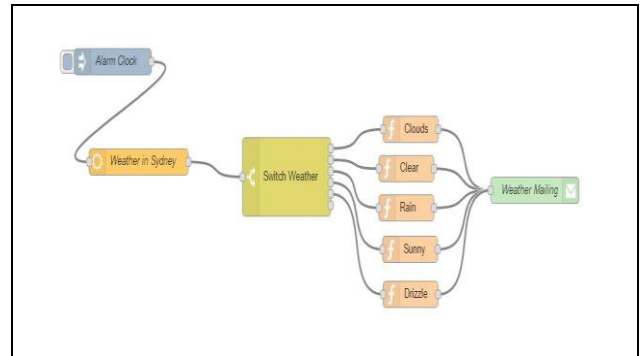


Figure 25: Sending a weather-dependent email flow.



Figure 26: Weather-dependent data display

The user needs to enter a username and an email address. Once this is done, one needs to go to API Keys page and copy the given API key the website has just generated. The user will need these in later stages.

A. Using OpenWeather Node Scenario

By default, OpenWeather already exists as an option in the range of nodes available within the Node-RED (Fig. 25). In our case, we can use are going to use the following node



Figure 27: OpenWeather node

Drag the icon (Fig. 27) into the working flow and let's set it up using the dedicated panel entries (Fig. 25).

- **API Key:** This is where you should paste the API key you have copied above
- **Location:** If the exact coordinates of the place are known and the user wants to know the details about the weather, the option “Coordinates” needs to be chosen, and for the remaining information, the user needs to enter the details for “City” and “Country” data.
- **Name:** user needs to provide the name of the node.

B. Sending a weather-dependent email

Let's consider another scenario, it is morning, you are in your bed, you have just opened your eyes and you are wondering what the weather looks like outside and how you should get dressed. Thanks to Node-RED, we can easily create an application that would check the

weather when your alarm clock is ringing and send you an email to your smartphone that informs you about the weather.

C. Node used

As we can see in the previous diagram, we only need a few nodes that already exist in Node-RED. What is new is the Node “**Inject**”: This node is called “Alarm Clock”. It aims to regularly send a stat signal to the application (at 7am every working day for example). This node also aims to inject a signal when the button on the left side of the node is clicked at any time when programming.

1. Node “**Switch**”: is a node that defines what the weather node sends, “enables” one of the following nodes: Clouds, Clear, Rain, Sunny... (Fig. 28).
2. The function nodes include a text and a topic that will be sent thanks to the Email Node which is the last node of this application (Fig. 29). The mechanism how this node works was shown in the previous example.

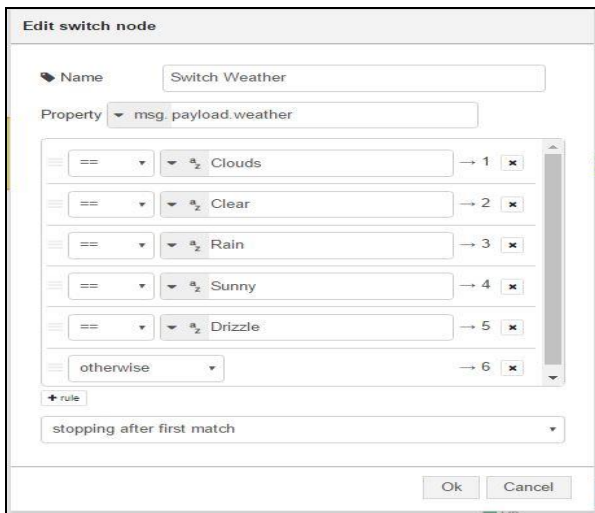


Figure 28: Listing of the switch weather node

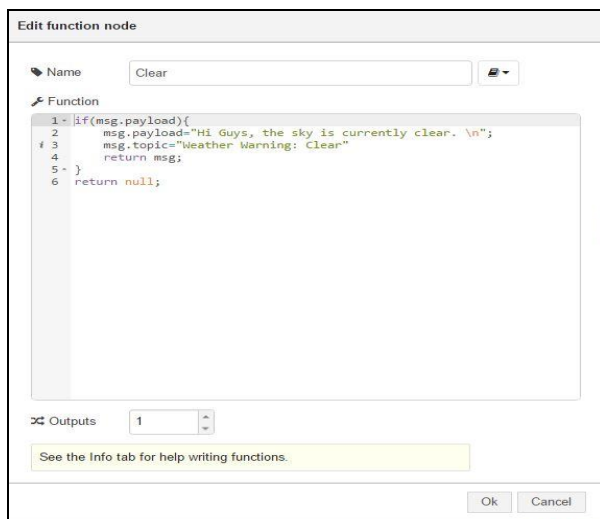


Figure 29: Listing of the weather node email message.

VI. CONCLUSION

This paper demonstrates the practicality of using open source tools such as: Raspberry Pi and the Node-RED platforms for teaching subject: the Internet of Things within a newly designed Data Engineering major at UTS. Starting from a relatively simple example and ending up with more sophisticated examples, IoT students learn the practical programming and hardware aspects of data acquisition, telemetry, data processing and visualization in context. The rapid development and engaging environment of Node-RED allowed our undergraduate students a successful and timely delivery of medium complexity IoT solutions with relative ease. The overall student feedback of the laboratory work was extremely positive; whilst the developed prototype solutions demonstrated a huge potential for adaptation in real-life applications. It was observed that our students have gained JavaScript programming skills, as well as, a measurable confidence in the development of their own IoT ideas.

REFERENCES

- [1] Node_RED, node-red contribution, <https://flows.nodered.org/node/node-red-contrib-pdf>, viewed: 12-04-2017.
- [2] Blackstock, M. and Lea, R. 2014. Toward a Distributed Data Flow Platform for the Web of Things (Distributed NodeRED). Proceedings of the 5th International Workshop on Web of Things (New York, NY, USA, 2014), 34–39.
- [3] IBM Bluemix - Next-Generation Cloud App Development Platform: <https://console.ng.bluemix.net/>, viewed: 12-04-2017.
- [4] Chang, F.C.; Chen, D.K.; Huang, H.C. Future Classroom with the Internet of Things A Service-Oriented Framework. J. Inf. Hiding Multimed. Signal Process. 2015, 6, 869–881.
- [5] Rasperry PI , “The definition of the Raspberry Pi, 2017, ” Available online: <https://www.raspberrypi.org/> (viewed on 24 March 2017).
- [6] Raspberry Pi—Teach, Learn, and Make with Raspberry Pi., Available online: <https://www.raspberrypi.org/> (accessed on 24 March 2017).
- [7] Managing Node-RED, init.d sript file, <https://learn.adafruit.com/raspberrypi-hosting-node-red/managing-node-red>,viewed April 2017.
- [8] Openweathermap,<http://home.openweathermap.org>, Viewed April 2017.
- [9] Navarro, M.; Tyler, W.D.; Villalba, G.; Li, Y.; Zhong, X.; Erratt, N.; Liang, X.; Liang, Y. Towards Long-Term Multi-Hop WSN Deployments for Environmental Monitoring: An Experimental Network Evaluation. J. Sens. Actuator Netw. 2014, 4, 297–330.