

Bachelor's thesis

Bachelor of Information and Communication Technology

2018

Nguyen Thanh Tung

WEATHER STATION AND ELECTRONIC DEVICE CONTROLLER

– With Texas Instruments SimpleLink CC3200
Wi-Fi SensorTag and Raspberry Pi

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Bachelor of Information and Communication Technology

2018 | 48 pages

Nguyen Thanh Tung

WEATHER STATION AND ELECTRONIC DEVICE CONTROLLER

- With Texas Instruments SimpleLink CC3200 Wi-Fi SensorTag and Raspberry Pi

In this thesis, the author introduces a demonstration of a weather station and electronic controller web-based application with main components including Raspberry Pi 3 model B and Texas Instruments SimpleLink CC3200 Wi-Fi SensorTag. The application wirelessly collects all sensor data and shows them on real-time graphical reports. Those sensor data are used to help existing consumer electronics work according to surrounding environmental changes. The application also allows the user to switch on/off connected electronic devices manually.

The developer uses Node-RED editor, Node.js modules, and JavaScript programming language to build an expected application. In this project, the SensorTag kit sends all collected data to IBM Watson IoT Platform and Node-RED hosted by Raspberry Pi which receives and processes those data for a real-time graphical report. Those data are used to control a 4 Relay Module that connects to Raspberry Pi GPIO pins in order to open/close 220V AC power circuit.

The outcome of the thesis is a web-based application with all expected functions. After the testing phases, obstacles, future improvements, and limitations are pointed out. Although the SensorTag is under-developing with limited functions, the demonstration is a concept for the next IoT development. The author also gain a tremendous amount of knowledge about technology, programming, and electronics after developing this project.

KEYWORDS:

Raspberry Pi, SensorTag kit, IoT, Node.js, Node-RED, weather station, electronic controller

CONTENT

LIST OF ABBREVIATIONS (OR) SYMBOLS	5
1 INTRODUCTION	6
1.1 Background	6
1.2 Motivation	7
1.3 Objectives	8
1.4 The structure of the thesis	8
2 TECHNOLOGY	9
2.1 Wi-Fi	9
2.2 Node.js	9
2.3 Node-RED	10
2.4 Raspberry Pi	10
2.5 Texas Instruments SimpleLink Wi-Fi CC3200 SensorTag	11
3 PROCESS	12
4 IMPLEMENTATION	14
4.1 Installation of Node.js and Node-RED on Raspberry Pi	14
4.2 Installation of additional nodes and flows on Node-RED	14
4.3 Connection of SensorTag kit and Raspberry Pi	15
4.4 The connection of Raspberry Pi and the 4 Relay Module.	16
4.5 Node-RED, nodes, and flows.	17
5 PERFORMANCE	23
6 DISCUSSION	25
6.1 Obstacles	25
6.2 Potential Future Improvement	26
6.3 Limitations	27
7 CONCLUSION	29
REFERENCES	30

APPENDICES

Appendix 1. Source code used in Node-RED application.

FIGURES

Figure 1. CC3200 Wi-Fi SensorTag block diagram (Texas Instruments Incorporated, 2016).	11
Figure 2. The implementation structure.	12
Figure 3. Switching to Station mode.	16
Figure 4. Adding Wi-Fi profile for establishing a connection to the local wireless network.	16
Figure 5. The connection of Raspberry Pi and 4 Relay Module.	17
Figure 6. Setting up 'ibmiot in' node.	18
Figure 7. Editing rpi-gpio out node.	19
Figure 8. Editing gauge and chart node.	20
Figure 9. Editing 'switch' and 'rpi-gpio out' node.	21
Figure 10. All complete flows.	22
Figure 11. User Interface for environment monitor and electronic device controller.	22
Figure 12. IBM Watson IoT Quickstart notification.	27

TABLES

Table 1. Power consumption (Texas Instruments Incorporated, 2016).	28
--	----

CODE

Code 1. Installation of Node.js and Node-RED.	14
Code 2. Installation of npm.	14
Code 3. Updating 'Python RPi.GPIO' module and add a user to GIPO group.	15
Code 4. Installation of additional nodes and flows.	15
Code 5. Humidity function node.	18
Code 6. Turn on/off node.	19

LIST OF ABBREVIATIONS (OR) SYMBOLS

Abbreviation	Explanation of abbreviation (Source)
IoT	Internet of Thing
GPIO	General Purpose Input-Output
MEMS	Microelectromechanical systems
JSON	JavaScript Object Notation
IEEE 802.11	Wireless computer networking standard
I/O	Input/Output
IBM	International Business Machines Corporation
CPU	Central Processing Unit
RAM	Random Access Memory
USB	Universal Serial Bus
AAA	Standard size of dry cell battery
MEMS	Micro-electro-mechanical systems
TLS	Transport Layer Security
MQTT	Message Queuing Telemetry Transport
HTTP	Hypertext Transfer Protocol
npm	Non-Parametric Mapping Utility
AC	Alternating Current
DC	Direct Current
mAh	milliampere hour
Ah	ampere hour

1 INTRODUCTION

1.1 Background

According to Toschi, Campos, and Cugnasca, “in the past, automation was restricted to high-value domains, for example, industry, military, or the space race efforts, but it is currently much more intimately woven into everyone's daily life”. This system gathers data from several sensors and actuators to analyze home environments (Toschi, Campos, & Cugnasca, 2017). Home users can connect to this system from wired and wireless devices for managing their building. Home automation has been widespread since the 80's; however, it has been reaching to an advanced reformation after the revolution of IoT with wireless technology and mobile devices (Mihalache, 2017).

There are various benefits when the wireless connection is available in the home automation system. First, the wireless connection significantly reduces installation costs as home automation does not require cables to connect between devices. Second, upgrade for existing home automation systems is increasingly simple thanks to no cable required. Especially, the manager of historical buildings often strictly forbid any structure changes during maintenance. Vinay Sagar and Kusuma (2015) believe that the improvement in the scalability and extension of the system “makes wireless installation a seminal investment”. Finally, the users find it easy to access to the home automation system from their mobile phone, which makes the user experience indeed convenient. However, the wireless system also has drawbacks that need to be considered.

During deploying a new wireless system, the technician has to “face three main challenges: high costs, vulnerability, and difficulty in achieving security” (Mihalache, 2017). The price of new application and equipment is one of the challenges for both the service provider and the home user. Besides, the manufacturer needs to update their technology when a new standard is applied to the home automation system, and research how that technology maximizes their product. Also, the home user has to learn how to use the new system with the supplier's technical support. Moreover, the final challenge for the wireless home automation system is to ensure security when an unauthorized device in the network tries to connect.

However, after a long time of development, home automation systems are developed to meet new requirements and reduce risks for both home users and suppliers. As a result,

wireless hardware is cheaper, faster and more reliable. Home users thus can manage their home automation system through the web interface on personal computers or mobile devices. Wi-Fi is chosen to improve system security.

Home automation can be controlled by either microcontroller or computer (Han, Yun, Jang, & Park, 2010) (Gamba, Gonella, & Palazzi, 2015). However, while home automation system performs with multiple processes at the same, the microcontroller is too weak to handle the whole system. In addition, home automation system which uses a computer as a central unit will raise the hardware and electric bills (Patchava, Kandala, & Babu, 2015). For reasons mentioned above, the microcontroller and computer are not employed in this project. Instead, according to Charles (2013), Raspberry Pi which contains GPIO and Wi-Fi adapter can easily communicate with other components. Therefore, Raspberry Pi is selected for the implementation of this project.

Since Raspberry Pi can process data and control electronic devices, home automation system still requires environment data input from different sensors, and those data need to be sent to Raspberry Pi wirelessly. Wi-Fi is the most promising connecting solution because “user equipment is already widely available” (Kaczmarek, Ruminski, & Bujnowski, 2016). Texas Instruments SimpleLink CC3200 Wi-Fi SensorTag complies with connection requirements, so its capacity of connection is stable. This SensorTag kit collects ambient environment data by nine low-power MEMS sensors, then send them to cloud with the built-in Wi-Fi connectivity (Texas Instruments Incorporated, 2016).

1.2 Motivation

Because wireless home automation system is developed rapidly, and wireless hardware is cheap and accessible, there is a great opportunity for students to understand and future-proof automation systems.

As an engineering student, after taking some physics and electronic courses at school, I am interested in building the wireless home automation system for improving my daily life. Now, all of electronic devices at my home become smart and take care of my home better with the automation system.

1.3 Objectives

This thesis demonstrates the process of establishing the web-based application which has two core functions of monitoring the home environment and controlling the electronic devices. Based on the data from SensorTag, the web-based application is capable of controlling the consumer electronics and allows users to modify manually. The real-time graphics report in the application shows the current situation of the home environment comprising temperature, humidity, pressure, and lightness in limited time. The application is hosted by Raspberry Pi and accessed from mobile phones, tablets, and computers which are connected to the same network with Raspberry Pi.

1.4 The structure of the thesis

Following this introduction, the next chapter describes technological tools utilized in this project. The thesis presents definitions and operations of several technologies such as IBM Watson IoT Platform, NodeJS, Raspberry Pi and the SensorTag kit.

While the third chapter explains how this project works in theory, the Implementation chapter will show the process of setting up every single component and how they connect to a system.

The fifth chapter named Performance will illustrate how the developer optimizes the workflow among components. The assessment chapter will suggest several obstacles, potential future improvement as well as limitation of the work.

The thesis will conclude with the seventh chapter which once again offers an overall look of the project.

2 TECHNOLOGY

2.1 Wi-Fi

Wi-Fi is a wireless local network technology, based on the IEEE 802.11 standard. It is a trademark of the Wi-Fi Alliance. Since most of the electronic devices nowadays are able to transmit the wireless signal. Wi-Fi becomes one of the most popular means for transferring data at home and in business. A wireless router allows other devices to have Internet access or share data together in a limited range. There is another way of using Wi-Fi is Wi-Fi hotspot which is used to share connectivity with other devices (Pinola, 2018).

Wi-Fi is chosen for this project because both Raspberry Pi and the SensorTag kit support Wi-Fi connection. The installation process does not require wiring and can be extended easily such as adding more SensorTag kit in different positions. Most of the current mobile phones, tablets, and computers support Wi-Fi connection so that users can access the system without difficulty and extra equipment.

2.2 Node.js

According to nodejs.org, "Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world" (nodejs.org, 2018). With the fast growth of JavaScript in recent years, Node.js has gained a vital role in web-based application development. Node.js helps the developer control both front-end and back-end part with JavaScript, by which a web development project is accomplishable with shorter time and fewer resources.

Node.js is applied in this project as it is one of the most productive choices for a project conducted by a single developer. With Node.js, the developer can understand every part and debug when errors appear in the code of the application. There are countless libraries based on Node.js, so the developer saves time on coding and concentrates more in-depth on programming.

2.3 Node-RED

Node-RED is a flow-based programming tool developed by IBM Emerging Technology Services. Based on Node.js, Node-RED allow the developer to edit and adjust the application by a visual editor on any web browser. In a program built by Node-RED, there are a series of nodes with specific purposes. Each node proceeds the given data according to the way developer designs, then pass those data to the next node in the same series. Node-RED provides a visual user interface which is easy to understand, modify and debug. An existing Node-RED application can be shared with other developers by exporting JSON files.

In this home automation project, because the application needs to work fluently with both the SensorTag kit and the GPIO pins on Raspberry Pi, the Node-RED is employed. The application processes sensor data for real-time graphics reports, then decide how electronic devices which connect to Raspberry Pi GPIO pins responds to surrounding environment changes.

2.4 Raspberry Pi

Raspberry Pi is a single-board computer developed by the Raspberry Pi Foundation – the UK for computer science education. Although Raspberry Pi has an affordable price, compact size, and efficient operation, it entirely functions as a computer for education and small business. This computer has Quad Core 1.2GHz CPU, 1GB RAM, 4 USB ports, 40-pin extended GPIO, Wi-Fi adapter and other connection options. Raspberry Pi runs Raspbian – the official operating system of Raspberry Pi Foundation and a few third-party operating systems (Raspberry Pi Foundation, 2016).

The author chooses Raspberry Pi 3 Model B because it is the latest model at the time the thesis is written. The Raspberry Pi is economical for setting up as a computer and replacing if the developer breaks it during installation or tests. It is simple to set up and perform all tasks in this project although Raspberry Pi Foundation does not supply it with powerful specifications. The Raspberry Pi Foundation optimizes the software to make this credit-card-size computer process run smoothly for hundreds of projects, such as a tablet with a touchscreen or a remote-control RC toy car. Indeed, the most critical reason

for selecting Raspberry Pi is that Node-RED can run on both Raspberry Pi and other computers in the same local network with all current web browser.

2.5 Texas Instruments SimpleLink Wi-Fi CC3200 SensorTag

Texas Instruments SimpleLink Wi-Fi CC3200 SensorTag is an important device to send data to the Raspberry Pi in this project. Based on the SimpleLink Wi-Fi CC3200 wireless MCU, with built-in Wi-Fi connection and powered by 2 AAA batteries, this development kit is produced by Texas Instruments Incorporated manufacturer. On this SensorTag kit, there is nine low-power MEMS sensor; hence the power can last long enough and is easy to be replaced with the AAA battery. Figure 1 shows how this kit designed. The Wi-Fi connection is configured to link to IBM Watson IoT Platform, and Texas Instruments claim that the SensorTag can be ready to use in 3 minutes without programming experiences (Texas Instruments Incorporated, 2016).

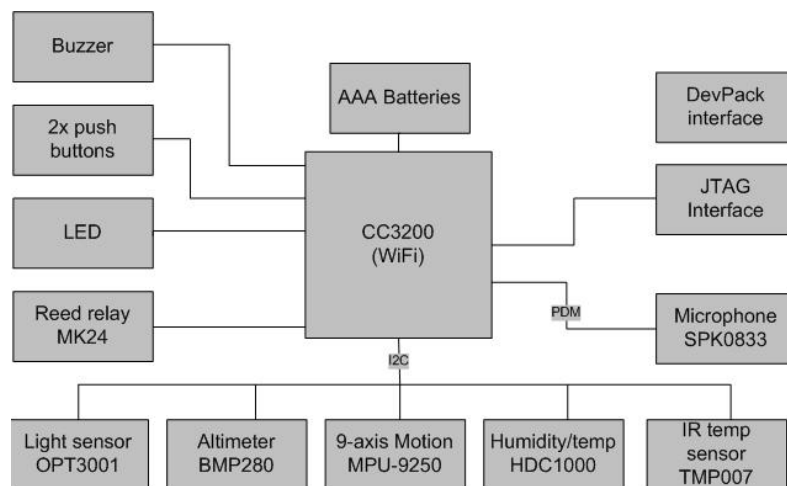


Figure 1. CC3200 Wi-Fi SensorTag block diagram (Texas Instruments Incorporated, 2016).

This SensorTag kit has nine sensors to collect home environment data on a single board. Thus the developer can avoid connectivity errors during the process of collecting data from each sensor. Because all sensor data can collect and send data directly to the cloud, the developer can add a substantial amount of those kits to one system in different positions. That means this system can extend in the future for a larger development. A larger capacity Lithium-ion battery can replace two AAA batteries in this kit for longer working time.

3 PROCESS

As the major hardware of the project, Raspberry Pi 3 Model B and Texas Instruments SimpleLink Wi-Fi CC3200 SensorTag communicate with each other by the local wireless network and the Internet. Most functions in this project will be programmed in Node-RED. Last but not least, the 4 Relay Module receives digital signals from Raspberry Pi through GPIO and turns on or off the existing 220V electronic devices.

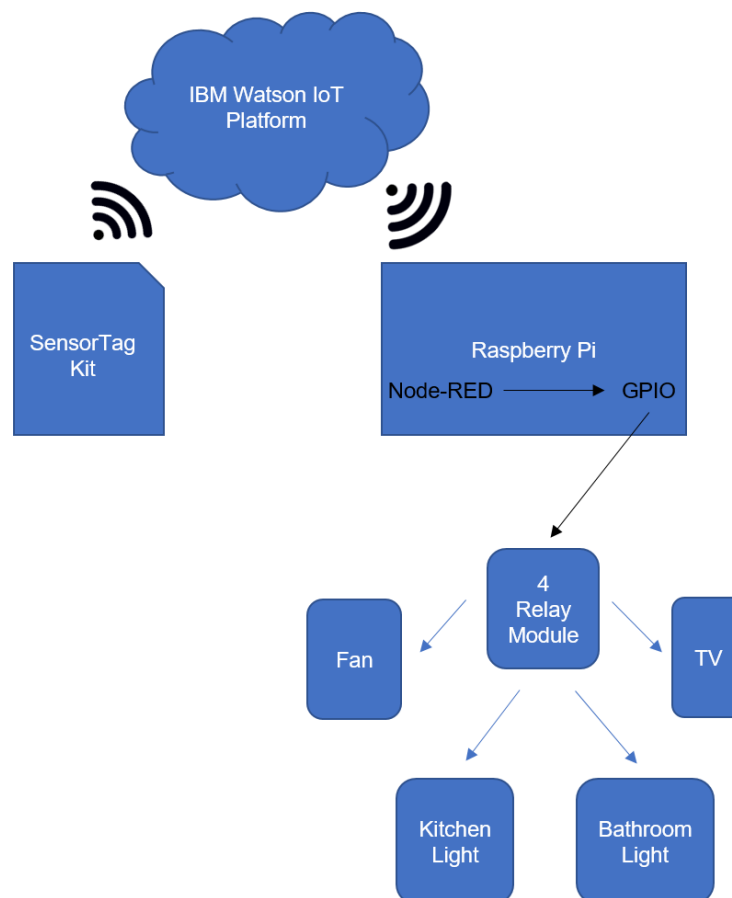


Figure 2. The implementation structure.

According to Figure 2, all data collected from nine sensors on the SensorTag kit are sent to IBM Watson IoT Platform in JSON format through the wireless Internet connection. Next, Raspberry Pi which is not necessary to be operated on the same local network with SensorTag receives those data for further processing in Node-RED. Following, Node-RED uses programmable nodes to divide JSON format data into different flows for particular purposes. The data of each sensor will be shown in the graphics of the web-

based application for monitoring. Besides, Raspberry Pi is not capable of turn on/off electronic devices by itself since most electronic devices work with 220V AC power, so the 4 Relay Module receives the digital signals on GPIO to turn on/off 220V AC power.

The web-based application automatically controls electronic devices through GPIO pins by analyzing the sensor data. Moreover, that web-based application allows users to control the electronic devices manually. Users can access the web-based application from any mobile device and computer in the same local network.

4 IMPLEMENTATION

4.1 Installation of Node.js and Node-RED on Raspberry Pi

In this project, Raspberry Pi 3 Model B is the main component which collects data from input device – the SensorTag kit to report information on the web-based application, then control output devices connected on GPIO pins. Raspbian can be downloaded and installed on the Raspberry Pi official website as mentioned earlier. Raspbian is a Linux-based operating system that enables the developer to finish modification and development on the command terminal without limitations.

The following command in Code 1 will install Node.js, Node-RED, then set Node-RED start when the Raspberry Pi boots up:

Code 1. Installation of Node.js and Node-RED.

```
1 bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)
2 sudo systemctl enable nodered.service
```

4.2 Installation of additional nodes and flows on Node-RED

Before installing additional nodes and flows, npm is required first. The following code in Code 2 shows commands are used to install and update npm

Code 2. Installation of npm.

```
1 sudo apt-get install npm
2 sudo npm install -g npm
```

After installing libraries, the developer needs to access Node-RED root folder and start adding necessary modules. Before accessing GPIO pins with Node-RED, the 'Python RPi.GPIO' module must be installed and updated to the latest version. Code 3 lists the used commands to update those libraries and add the current user to the GPIO group member.

Code 3. Updating 'Python RPi.GPIO' module and add a user to GIPO group.

```
1 sudo apt-get update && sudo apt-get install python-rpi.gpio  
2 sudo adduser $USER gpio
```

The second module is 'node-red-contrib-scx-ibmiotapp'. This module connects securely and automatically to IBM Watson IoT Platform with TLS, MQTT and HTTP protocols, in order that it works both inside and outside that platform. The node supports JSON, buffer, number, string, and boolean formats when sending and receiving data (Mangalvedkar & Dare, 2016). The command line 1 of Code 4 installs this module.

The third module is named as 'node-red-dashboard'. This module provides a set of Node-RED nodes to create live data graphs and controllers on the web-based application. The layout of web-based application can be modified by a dashboard tab on the left side of the Node-RED editor. In Layout tab, the order of tabs, groups can be adjusted and edited. The next tab is called Theme tab, which contains options of theme styles, base colors, and fronts. In Site tab, Node-RED provides modifications of site titles, groups and widgets size. The command line 2 in Code 4 installs this module.

Code 4. Installation of additional nodes and flows.

```
1 npm install node-red-contrib-scx-ibmiotapp  
2 npm install node-red-dashboard  
3 node-red-stop  
4 node-red-start
```

After installing compulsory modules, Node-RED is needed to be restarted by commands on line 3 and 4 of Code 4 in order that the Node-RED can refresh the flow editor and load the new nodes.

4.3 Connection of SensorTag kit and Raspberry Pi

The Texas Instruments SimpleLink Wi-Fi CC3200 SensorTag is a SensorTag kit with nine sensors on the same board and connects to the local wireless network by a Wi-Fi chipset; it is simple and straightforward for sending data to other devices through the Internet. To collect those data, the received device is not necessary to be on the same local network as that SensorTag kit. After accessing to a web page with SensorTag IP

address, the developer can modify settings of the Sensor tag according to the project's requirements.

The SensorTag kit is in Access Point mode by default, hence the developer has to switch Device mode to Station mode in Device Config tab, which is displayed in Figure 3.

The screenshot shows a configuration window titled "Device". It contains two sections. The first section has "Device Name:" with a text input field containing "SensorTag-48" and "Domain Name:" with a text input field containing "sensortag.net". Below these fields is an "Apply" button. The second section has "Device Mode:" with a dropdown menu set to "Station" and another "Apply" button. At the bottom of this section, it says "Mode will take effect after Reset".

Figure 3. Switching to Station mode.

The wireless connection, which is established by filling a form shown in Figure 4, connects the SensorTag kit to IBM Watson IoT platform.

The screenshot shows a configuration window titled "Add Profile". It contains four fields: "SSID:" with a text input field containing asterisks; "Security Type:" with radio buttons for "Open", "WEP", "WPA1", and "WPA2", where "WPA2" is selected; "Security Key:" with a text input field containing asterisks and a note "Hexadecimal digits - any combination of 0-9, a-f and A-F"; and "Profile Priority:" with a text input field containing "1" and a note "Enter a value 0-7 (0 = highest)". Below these fields is an "Add" button. At the bottom of the window, it says "The new Profile will take affect only after reset".

Figure 4. Adding Wi-Fi profile for establishing a connection to the local wireless network.

As a result, the SensorTag kit data can be read on IBM IoT platform Quickstart website.

4.4 The connection of Raspberry Pi and the 4 Relay Module.

The 4 Relay Module is added to this project since the Raspberry Pi cannot work with the consumer electronics by itself. This equipment receives digital signal from GPIO pins of the Raspberry Pi for closing or opening 220V AC power circuit. Because there are 4 red LEDs displaying the status of 4 output ports on that Relay Module, it is unnecessary for the module to attach to actual electronic devices for testing purposes. Besides,

Raspberry Pi's GPIO and the 4 Relay Module are connected to each other by 6 pins including a ground connection pin, 4 input-signal pins, and a 5V DC connection pin. In this project, while the ground connection pin connects to a 6 – Ground pin, the 5V DC connection pin links to a 2 – 5V Power pin. The 4 input-signal pins connect to the 32 – GPIO12, 36 – GPIO16, 38 – GPIO20 and 40 - GPIO21 pins. All the connections are exhibited in Figure 5. The process of connecting which is described above completes the hardware connection in this project.



Figure 5. The connection of Raspberry Pi and 4 Relay Module.

4.5 Node-RED, nodes, and flows.

After connecting necessary equipment and installing applications and nodes, the Node-RED editor is ready for this project and able to get started by visiting <http://theRaspberryPiIPAddress:1880>.

Figure 6. Setting up 'ibmiot in' node.

As we can see in Figure 6 above, the first node is the 'ibmiot in' node which connects from the Node-RED to IBM Watson IoT Platform. In the node properties tab, the developer needs to set Quickstart in Authentication, Device Event in Input Type from the drop-down list. Next, Device ID is different for each device, and Name is typed as cc3200stk. The sensors' data will be downloaded from IBM Watson IoT Platform to Node-RED in JSON format.

After establishing a connection to the SensorTag kit, 5 sensors will gather data for this project consisting of humidity, object temperature, ambient temperature, pressure and ambient light. All sensor data will be processed in the same method. So, in this chapter, only the humidity sensor data is presented as an illustration. The processed code of other 4 sensors can be found in Appendix 1.

For the humidity sensor, a function node is required to separate the data of 9 sensors into the humidity sensor data, and then pass those data to the next node. The command used in this function node is Code 5 below.

Code 5. Humidity function node.

```

1 var o = msg.payload;
2 msg.payload = o.d.humidity;
3 return msg;

```

After the humidity function node, a turn on/off function node decides how the connected input pin in the 4 Relay Module works. This function gets the humidity value in the payload element, then it will send a new payload valued 1 when the humidity value exceeds 30 degrees Celsius. Otherwise, it will send 0 to the next nodes. All used command in this node are exposed in Code 6.

Code 6. Turn on/off node.

```

1 if (msg.payload < 30) {
2     msg.payload = 1;
3 }
4 else{
5     msg.payload = 0;
6 }
7 return msg;

```

Edit rpi-gpio out node

node properties

Pin

3.3V Power - 1	2 - 5V Power
SDA1 - GPIO02 - 3	4 - 5V Power
SCL1 - GPIO03 - 5	6 - Ground
GPIO04 - 7	8 - GPIO14 - TxD
Ground - 9	10 - GPIO15 - RxD
GPIO17 - 11	12 - GPIO18
GPIO27 - 13	14 - Ground
GPIO22 - 15	16 - GPIO23
3.3V Power - 17	18 - GPIO24
MOSI - GPIO10 - 19	20 - Ground
MISO - GPIO09 - 21	22 - GPIO25
SCLK - GPIO11 - 23	24 - GPIO8 - CE0
Ground - 25	26 - GPIO7 - CE1
SD - 27	28 - SC
GPIO05 - 29	30 - Ground
GPIO06 - 31	32 - GPIO12
GPIO13 - 33	34 - Ground
GPIO19 - 35	36 - GPIO16
GPIO26 - 37	38 - GPIO20
Ground - 39	40 - GPIO21

Type:

Initialise pin state?

Name:

Pins in Use:

Tip: For digital output - input must be 0 or 1.

Figure 7. Editing rpi-gpio out node.

For controlling GPIO pin, a 'rpi-gpio out' node is connected to the 'turn on/off' node, which can be viewed in Figure 7. The Pin 32 – GPIO12 that is picked out to control the 'input 1' pin on the 4 Relay Module will turn on/off the moisturizer machine of this project. This node has digital outputs which have only two values 0 or 1. Gauge node and chart node are used to show the humidity sensor data in the web-based application. In the gauge node editing window, the author respectively chooses 'Humidity [Home Office]' in Group tab, 'Donut' in Type tap, '%' in Units tab. Afterwards, Range tab is set from 0 to 100, which is shown on the left side of Figure 8. Moving to the set-up of the chart node, the setting will be presented on the right side of Figure 8. The chart node will be in the same group as the gauge node. The Type tab is set to 'Line chart', the X-axis tab is chosen to last 10 minutes with 'HH:mm:ss' label format.

Edit gauge node

Delete Cancel Done

▼ node properties

Group: Humidity [Home Office]

Size: auto

Type: Donut

I Label:

I Value format: {{value}}

I Units: %

Range: min 0 max 100

Colour gradient: [Green] [Yellow] [Red]

Sectors: 0 ... optional ... optional ... 100

Name: humidity

Edit chart node

Delete Cancel Done

▼ node properties

Group: Humidity [Home Office]

Size: auto

I Label: optional chart title

Type: Line chart enlarge points

X-axis: last 10 minute OR 1000 points

X-axis Label: HH:mm:ss

Y-axis: min max

Legend: None Interpolate linear

Series Colours: [Blue] [Light Blue] [Orange] [Green] [Light Green] [Red] [Pink] [Purple] [Light Purple]

Blank label: display this text before valid data arrives

Use deprecated (pre 0.2.5) data format.

Name: humidity

Figure 8. Editing gauge and chart node.

To manually control the moisturizer machine, we need a pair of a 'switch' node and a 'rpi-gpio out' node as displayed in Figure 9. For the 'switch' node, 'Controller [Home Office]' is set for 'Group' and 'Moisturizer Machine' is selected in both 'Label' and 'Name'. The 'On Payload' and 'Off Payload' lines are the important set-ups to make this pair of node operate. Number formats that valued 1 and 0 are respectively selected for the 'On Payload' and 'Off Payload'

That is how the humidity sensor data are processed to show in graphs and control the moisturizer machine. This process also applies to 4 other sensors encompassing object temperature, ambient temperature, pressure and ambient light sensors. After 5 sensors experience the similar process above, the Node-RED application is completed for deploying. The result is illustrated in Figure 10 and Figure 11. It is noted that the 'rpi-gpio out' node shows its current state, for example 0 when the connected electronic device is off and 1 when it is on.

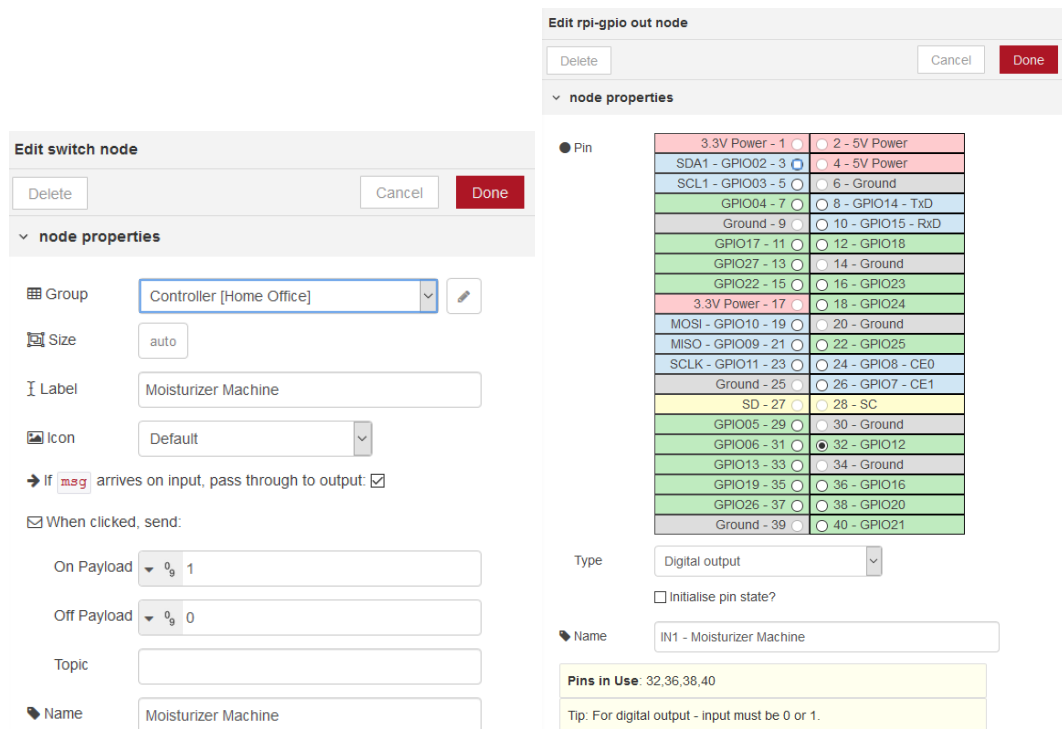


Figure 9. Editing 'switch' and 'rpi-gpio out' node.

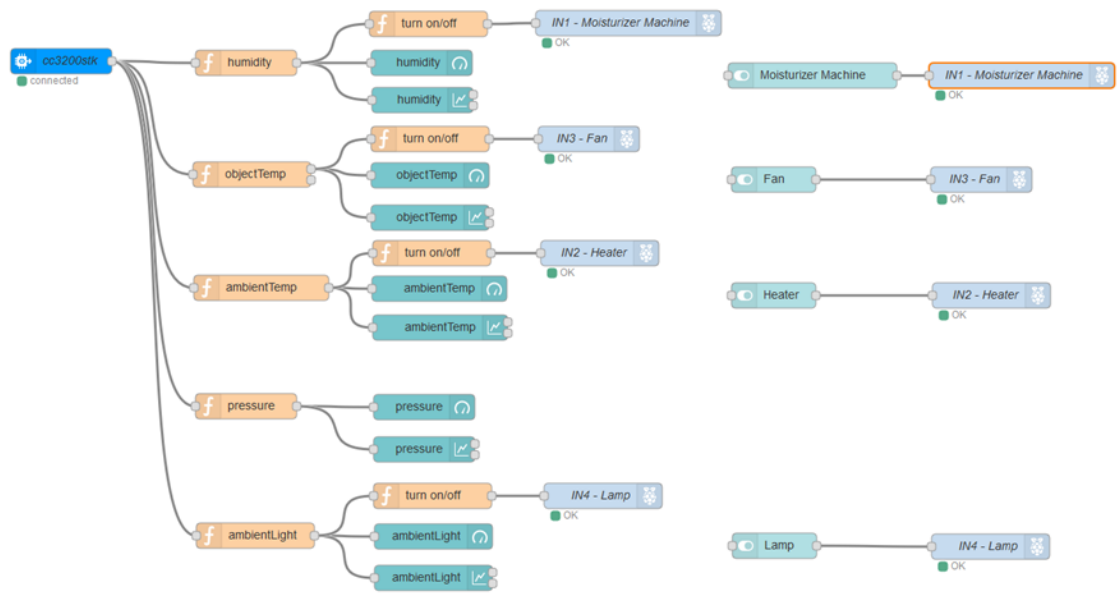


Figure 10. All complete flows.

As we can see from Figure 11, there are 6 different tabs. Among these tabs, five of them show data from sensors while the last one allows user control all connected electronic devices. The developer can add more subpages for different rooms when more SensorTag kits are added to the network.

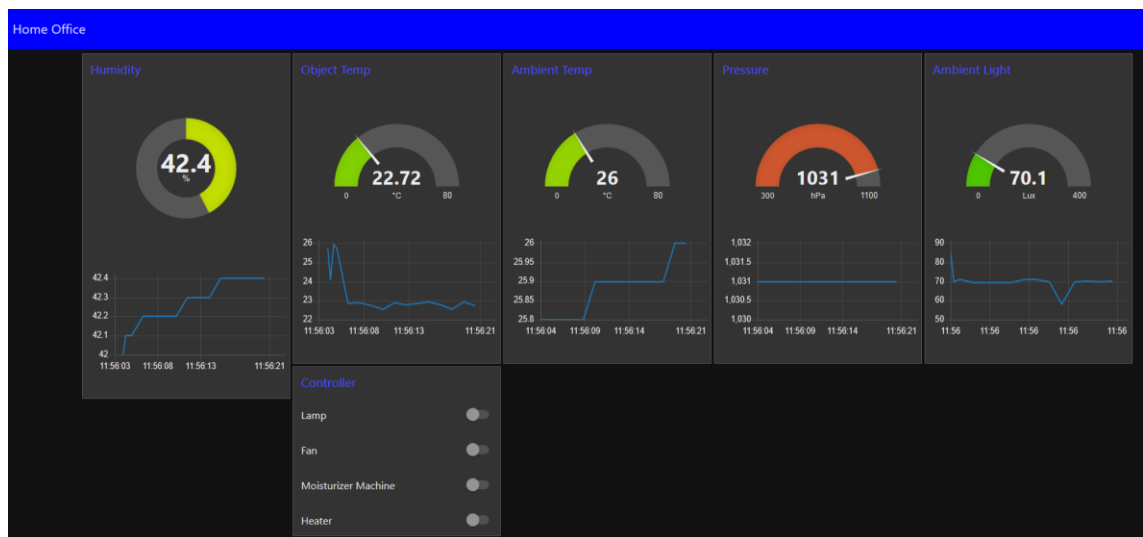


Figure 11. User Interface for environment monitor and electronic device controller.

5 PERFORMANCE

In developing web-based application, performance is a crucial criterion to evaluate the quality of the web-based application. In order to optimize an application's performance, the developer has to minimize all processing which delays the user experiences. That required improvements in both client and server's side. However, the server side is not a big concern when most of modern servers are powerful and well-modified by the service provider. Also, this application does not need a high-end server to process tasks so the developer will be free from avoiding heavy load on servers (Bursztein, 2011). In this chapter, the developer presents five critical factors that are essential for optimizing the performance of this project.

First of all, a SensorTag kit with the built-in cloud connection is chosen as the input devices. A huge amount data of nine individual sensors is transferred directly to IBM Watson IoT Platform and received by a Raspberry Pi with Internet connection. That method maximizes the advantage of the Wi-Fi population and large bandwidth connection, thus the connection is guaranteed to be stable and avoid overload. The Wi-Fi system is more expandable as an improvement than the wired or Bluetooth connection system.

Second, Node.js is the main programming technology in this project which progresses rapidly in both server and client sides. The developer can create and expand this project with one programming language, which saves countless time and line of code. A massive amount of node is created in Node.js specifically for Node-RED in this case, so the developing process is likely straightforward from an idea to a ready product. For the SensorTag kit data, Node.js converts the values to the chart-compatible type for all different sensors with specific IDs. That procedure avoids mistakes among a variety of data sources during processing information.

Thirdly, the end-user application is supposed to have two main functions including showing the number and chart of sensor data graphically and controlling electronic devices automatically when surrounding environment changes. Node-RED is believed to be a promising option to achieve that goal without limitation of time and programming knowledge. It means the developer can control and select functions of the GPIO pin solely with Node.js without the experience of Python which is the original programming language for controlling those pins.

Fourth, the application is tested to make sure both automatic and manual electronic device controllers functional. The environment changes are analyzed by the Node-RED node and tested how electrical devices react to those changes. The behaviors of the controller must meet required expectations and respond to the surrounding environment change without delay. On the other hand, when the user wants to turn on or off any electronic devices with the switch on the web-based application. Additionally, the manual controller also needs to respond instantly to the right device as the automatic controller.

Last but not least, the application has to be a responsive design which is a standard for most of current websites and web-based applications in the market nowadays. Particularly in this project, the web-based application is used for users who usually do not want to access their home automation system from a computer. The typical controller for this system should be a smartphone, tablet or wall-mounted station. In addition, these devices have different screen sizes, so the web-based application needs to render well in various web browsers and screen resolutions.

6 DISCUSSION

This project covers a fair number of technologies from hardware to software, and then combine them to produce a functional wireless home automation system. A vast body of technical research and testings were done to find out appropriate technologies for this projects. During development, there are several problems arising which help the developer learns not only technical but also problem-solving skills by finding out how to face those challenges. The whole project duration is supposed to be 567 working hours but it takes shorter time to build a home automation system from scratching. It is obvious that many functions still can be developed further.

6.1 Obstacles

The web-based application meets most of requirements, but some problems still exist. Therefore, further developments can be taken into account.

First, the measurement of the ambient environment condition by the SensorTag kit still shows some minor errors. During the testing stage, some sensors within the SensorTag kit display an unusual value that can cause strange behaviors of electronic devices or severe damages for the whole system. For instance, the object temperature sensor returns over 100 degrees Celsius when it is attached to a human body. After replacing the battery, cleaning the circuit board and updating the SensorTag firmware, that error is not solved. In conclusion, it could be a calibration issue.

Second, the connection from the SensorTag kit to the application is deployed or even not responded at all. At the time of writing this report, the manufacturer limits the SensorTag connection to IBM Watson IoT Platform only. Even though both the SensorTag kit and Raspberry Pi have a built-in Wi-Fi connection, there are no method to connect them directly. The data which flows from sensors to the application can be interrupted by the Internet disconnection failure of different components such as wireless router, Internet cables, and IBM server performance.

Third, the connection of GPIO pins and the relay module can be unstable, which makes the controller not responsive as expected. All pins on the Raspberry Pi and the relay

module are not only tiny but also crowded; therefore, the lack of electronic assembling skills can damage those pins or even the device if a short circuit happens.

Lastly, the connection to automation system is not adequately secured. The access credential is not required when visiting the web-based application, which is a favorable opportunity for unauthorized accesses harm the system or override the controlling priority of homeowner. The wireless router has their own security solution, and the Node-RED editor allows the developer set up a password login before being able to edit any flow. However, these solutions are all third-party solutions and not specifically built for this project. The automation system can be controlled from any device as long as this device get access to the local wireless Internet network. As a result, to ensure safety and privacy, it is not recommended for adding security home equipment such as door locker, door alarms, and security cameras into this automation system.

6.2 Potential Future Improvement

The sensor which has not been in use in the SensorTag kit can be employed for more additional functions in this project. This nine-axis sensor reports the activity of some moving subjects indoor, such as a door, window, and clothes closet. For example, the automation can be noticed when the owner is back home, then turn on selected electronic devices to welcome him/her. Alternatively, if it is applied to the clothes closet, mirror lights and music can be turned on automatically when the system detected that the door is opened, and someone is coming in.

After developing functions from existing equipment, the developer can add more equipment which is comparable to the GPIO pins. Because Raspberry Pi is well-developed in the past few years, a vast number of external modules are created for Raspberry Pi. The developer can use those modules to serve more purpose. The 4 Relay Module in this project is an example. Besides, the IR sensor is a popular supplement to Raspberry Pi. A large body of projects is built based on this sensor. This system can receive voice command by adding any voice assistant devices such as Amazon Alexa, Google Assistant or Apple Siri which are typical products with frequent updates in the market.

In addition to the convenience of automation system, a user might want to analyze their home environment condition. The large-scale report function is vital to create for

reviewing for weeks, months or even years. A hard disk drive becomes a must add-on module to the Raspberry Pi, or a network attached storage can be another good solution particularly when SensorTag kits in different positions are added to this system. However, the user can estimate the home condition and support home maintenance after years of ownership with that massive amount of data.

6.3 Limitations

The most serious issue after developing this automation system is the connection from SensorTag kit to Raspberry Pi. The manufacturer Texas Instruments Incorporated only supports IBM Watson IoT Platform for collecting data from SensorTag kit. That means there is only one option to receive those sensor data by connecting Raspberry Pi to the IoT Platform, which leads to delays or unresponsive command during processing. As a result, all automatic functions will freeze. During the testing stage, the application does not show any value from the SensorTag kit because of the connection delay from SensorTag kit to IBM Watson IoT Platform. The IBM Watson IoT Quickstart page displays a notification in Figure 12 as following:

The image shows a notification message from the IBM Watson IoT Quickstart interface. The text is displayed in a light gray font on a white background. The message states: "The device ID you have entered is valid, but we are waiting for your device to send us a message. Get your device to send us some valid data - we'll visualize it for you!"

Figure 12. IBM Watson IoT Quickstart notification.

Apart from the connection delay, SensorTag kit battery consumption is a drawback before applying this automation system to consumer use. The SensorTag kit is required to work continually to track all environment changes in real-time. In a testing condition, around 20 degrees Celsius ambient temperature, the SensorTag kit work for a few days with a pair of full-charged AAA batteries. The detail power consumption for each sensor on SensorTag kit is listed in Table 1 below.

Table 1. Power consumption (Texas Instruments Incorporated, 2016).

Sensors	Current in milliamp (mA)
MPU-9250 Nine-Axis MEMS MotionTracking Device	3.5
TMP007 Infrared Thermopile Sensor	0.27
BMP280 Barometric Pressure Sensor	0.0027
HDC1000 Digital Humidity Sensor	0.0012
OPT3001 Ambient Light Sensor	0.0018
Total current	3.7757

According to real-time testing, a pair of full-charged AAA batteries, approximately 1500 mAh of total capacity, can keep the SensorTag kit working about 397 hours continuously in an ideal condition. In this case, an external battery or power bank gives the SensorTag kit a longer battery life which reduces maintenance costs.

7 CONCLUSION

Consequently, the weather station and electronic device controlling system effectively function and meet all expectations listed as follows:

- All surrounding environmental elements in a home office such as humidity, temperature, pressure, and light quantity are measured.
- A real-time graphic report is easy to understand and shows selected sensor values.
- A web-based application allows users control consumer electronics without delay.
- A web-based application which compatible with all current web browsers can be accessed from any device in the same local wireless network.
- All chosen electronic devices can react as expected when the surrounding environmental condition changes.

To achieve that goal, the JavaScript programming language and electronic assembling skills are mandatory for this project. Many studies are made to master those skills despite the lack of technical knowledge and experience from the author. Although the mentioned challenges take the developer longer to complete the project, a web-based application is successfully built with the help of Node-RED and extra Node.js modules. Most functions work up to expectations, except that some of them would be better if the hardware limitations are addressed by the manufacturer.

At the time of developing this project, the SensorTag kit has been recently released and unpopular. Although several functions are still under developing or limited by Texas Instruments Incorporated, a large number of functions can be established based on this SensorTag kit. This thesis is believed to be an encouraging concept for not only the author but also other developers in the future project.

REFERENCES

- Bursztein, E. (2011, March). *Elie*. Retrieved from Analyzing web application performance: <https://elie.net/blog/web/analyzing-web-application-performance>
- Gamba, M., Gonella, A., & Palazzi, C. E. (2015). Design issues and solutions in a modern home automation system. *Proceedings of International Conference on Computing Networking and Communications (ICNC 2015)*, 1111-1115.
- Han, J., Yun, J., Jang, J., & Park, K.-R. (2010). User-friendly home automation based on 3D virtual world. *IEEE Transactions on Consumer Electronics*, 1843-1847.
- Johnston, S., & Cox, S. (2017). The Raspberry Pi: A Technology Disrupter, and the Enabler of Dreams. *Electronics*, 51.
- Kaczmarek, M., Ruminski, J., & Bujnowski, A. (2016). Accuracy analysis of the RSSI BLE SensorTag signal for indoor localization purposes. *Computer Science and Information Systems (FedCSIS)*, 1413-1416.
- Mangalvedkar, A. M., & Dare, J. (2016). *node-red-contrib-scx-ibmiotapp*. Retrieved from nodered.org: <https://flows.nodered.org/node/node-red-contrib-scx-ibmiotapp>
- Mihalache, A. (2017). Wireless Home Automation System using IoTq. *Informatica Economica*, 17-32.
- nodejs.org*. (2018, May 30). Retrieved from Node.js Web Site: <https://nodejs.org/en/>
- Patchava, V., Kandala, H. B., & Babu, P. R. (2015). A Smart Home Automation technique with Raspberry Pi using IoT. *International Conference on Smart Sensors and Systems (IC-SSS)*.
- Pinola, M. (2018, March 6). *Understanding Wi-Fi and How it Works*. Retrieved from Lifewire: <https://www.lifewire.com/what-is-wi-fi-2377430>
- Raspberry Pi Foundation. (2016). *Raspberry Pi 3 Model B*. Retrieved from www.raspberrypi.org: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- Severance, C. (2013). Eben Upton: Raspberry Pi. *Computer*, 14-16.

- Texas Instruments Incorporated. (2016). *www.ti.com*. Retrieved from *www.ti.com*: <http://www.ti.com/tool/cc3200stk-wifimk#1>
- Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to Build High-Performance Network Programs. *IEEE Internet Computing*, 80-83.
- Toschi, G. M., Campos, L. B., & Cugnasca, C. E. (2017). Home automation networks: A survey. *Computer Standards & Interfaces*, 42-54.
- Vinay Sagar, K. N., & Kusuma, S. M. (2015). Home Automation Using Internet of Things. *International Research Journal of Engineering and Technology*, 1965-1970.

Source code used in Node-RED application

```
[
  {
    "id": "c78ab654.242b38",
    "type": "tab",
    "label": "Flow 1",
    "disabled": false
  },
  {
    "id": "c7c56e23.3ed39",
    "type": "ui_group",
    "z": "c78ab654.242b38",
    "name": "IR Temperature",
    "tab": "6539d37c.35bd0c",
    "order": 2,
    "disp": true,
    "width": "6"
  },
  {
    "id": "3f3925c1.c2c18a",
    "type": "ui_group",
    "z": "c78ab654.242b38",
    "name": "Ambient Temperature",
    "tab": "6539d37c.35bd0c",
    "order": 3,
    "disp": true,
    "width": "6"
  },
  {
    "id": "9fb7a5b7.c4b998",
    "type": "ui_group",
    "z": "c78ab654.242b38",
    "name": "Barometer",
    "tab": "6539d37c.35bd0c",
    "order": 4,
    "disp": true,
    "width": "6"
  },
  {
    "id": "f897b6d3.483358",
    "type": "ui_group",
    "z": "c78ab654.242b38",
    "name": "Ambient Light",
    "tab": "6539d37c.35bd0c",
    "order": 5,
    "disp": true,
    "width": "6"
  },
  {
    "id": "ea6a165.b926fe8",
    "type": "ui_group",
    "z": "c78ab654.242b38",
    "name": "Humidity",
    "tab": "6539d37c.35bd0c",
    "order": 1,
    "disp": true,

```



```

    "width": "6"
  },
  {
    "id": "e39ac663.7aaeb8",
    "type": "ui_tab",
    "z": "c78ab654.242b38",
    "name": "Home Office",
    "icon": "dashboard",
    "order": 2
  },
  {
    "id": "3a869c8.8601764",
    "type": "ui_base",
    "theme": {
      "name": "theme-dark",
      "lightTheme": {
        "default": "#0094CE",
        "baseColor": "#0094CE",
        "baseFont": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
        "edited": true,
        "reset": false
      },
      "darkTheme": {
        "default": "#097479",
        "baseColor": "#0000ff",
        "baseFont": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
        "edited": true,
        "reset": false
      },
      "customTheme": {
        "name": "Untitled Theme 1",
        "default": "#4B7930",
        "baseColor": "#4B7930",
        "baseFont": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
        "reset": false
      },
      "themeState": {
        "base-color": {
          "default": "#097479",
          "value": "#097479",
          "edited": false
        },
        "page-titlebar-backgroundColor": {
          "value": "#0000ff",
          "edited": false
        },
        "page-backgroundColor": {
          "value": "#111111",
          "edited": false
        },
        "page-sidebar-backgroundColor": {
          "value": "#000000",
          "edited": false
        },
        "group-textColor": {
          "value": "#4d4dff",

```

```

        "edited": false
    },
    "group-borderColor": {
        "value": "#555555",
        "edited": false
    },
    "group-backgroundColor": {
        "value": "#333333",
        "edited": false
    },
    "widget-textColor": {
        "value": "#eeeeee",
        "edited": false
    },
    "widget-backgroundColor": {
        "value": "#0000ff",
        "edited": false
    },
    "widget-borderColor": {
        "value": "#333333",
        "edited": false
    },
    "base-font": {
        "value": "-apple-system,BlinkMacSystemFont, Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
    }
},
"site": {
    "name": "Node-RED Dashboard",
    "hideToolbar": "false",
    "allowSwipe": "false",
    "dateFormat": "DD/MM/YYYY",
    "sizes": {
        "sx": 48,
        "sy": 48,
        "gx": 6,
        "gy": 6,
        "cx": 6,
        "cy": 6,
        "px": 0,
        "py": 0
    }
},
{
    "id": "df2bb508.6bb7f8",
    "type": "ui_group",
    "z": "",
    "name": "Object Temp",
    "tab": "e39ac663.7aaeb8",
    "order": 2,
    "disp": true,
    "width": "9"
},
{
    "id": "75186061.6a979",
    "type": "ui_group",
    "z": "",

```

```

    "name": "Ambient Temp",
    "tab": "e39ac663.7aaeb8",
    "order": 3,
    "disp": true,
    "width": "9"
  },
  {
    "id": "f9012aac.c516b8",
    "type": "ui_group",
    "z": "",
    "name": "Pressure",
    "tab": "e39ac663.7aaeb8",
    "order": 4,
    "disp": true,
    "width": "9"
  },
  {
    "id": "c805fa77.a72e78",
    "type": "ui_group",
    "z": "",
    "name": "Ambient Light",
    "tab": "e39ac663.7aaeb8",
    "order": 5,
    "disp": true,
    "width": "9"
  },
  {
    "id": "80fcb4a3.e9d128",
    "type": "ui_group",
    "z": "",
    "name": "Humidity",
    "tab": "e39ac663.7aaeb8",
    "order": 1,
    "disp": true,
    "width": "9"
  },
  {
    "id": "4ae148b.6dd8cb8",
    "type": "ui_group",
    "z": "",
    "name": "Controller",
    "tab": "e39ac663.7aaeb8",
    "order": 6,
    "disp": true,
    "width": "9"
  },
  {
    "id": "37188da2.e5a852",
    "type": "ui_tab",
    "name": "Tab 2",
    "icon": "dashboard",
    "order": 2
  },
  {
    "id": "408b98.9f193c68",
    "type": "ui_group",
    "name": "Group 1",
    "tab": "37188da2.e5a852",
    "order": 1,

```

```

        "disp": true,
        "width": 6
    },
    {
        "id": "4423a4a.e6d285c",
        "type": "function",
        "z": "c78ab654.242b38",
        "name": "humidity",
        "func": "var o = msg.payload\nmsg.payload =
o.d.humidity;\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 319,
        "y": 91,
        "wires": [
            [
                "927a002d.0cda3",
                "a8b1202b.593d6",
                "80f6d802.72e2c8"
            ]
        ]
    },
    {
        "id": "cb65d904.bef908",
        "type": "function",
        "z": "c78ab654.242b38",
        "name": "objectTemp",
        "func": "var o = msg.payload\nmsg.payload =
o.d.objectTemp;\nreturn msg;",
        "outputs": "2",
        "noerr": 0,
        "x": 326,
        "y": 223,
        "wires": [
            [
                "e1b3f475.9c5088",
                "87252df7.9d318",
                "91c8fc58.2643e"
            ],
            []
        ]
    },
    {
        "id": "a00aee2e.50a0d",
        "type": "function",
        "z": "c78ab654.242b38",
        "name": "ambientTemp",
        "func": "var o = msg.payload\nmsg.payload =
o.d.ambientTemp;\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 337,
        "y": 357,
        "wires": [
            [
                "d9e5889d.3d18f8",
                "a4b0eda3.4752",
                "2a1f1680.c3fc2a"
            ]
        ]
    }
}

```

```

    ]
  },
  {
    "id": "951f4f76.be40f",
    "type": "function",
    "z": "c78ab654.242b38",
    "name": "pressure",
    "func": "var o = msg.payload\nmsg.payload =
o.d.pressure;\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 319,
    "y": 497,
    "wires": [
      [
        "f2482df6.d490f",
        "ce147e28.058c6"
      ]
    ]
  },
  {
    "id": "38572e76.3dbbd2",
    "type": "function",
    "z": "c78ab654.242b38",
    "name": "ambientLight",
    "func": "var o = msg.payload\nmsg.payload =
o.d.ambientLight;\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 330,
    "y": 650,
    "wires": [
      [
        "36bd43cc.79321c",
        "f143e8f.49e5118",
        "f9d25499.a413a8"
      ]
    ]
  },
  {
    "id": "e1b3f475.9c5088",
    "type": "ui_gauge",
    "z": "c78ab654.242b38",
    "name": "objectTemp",
    "group": "df2bb508.6bb7f8",
    "order": 0,
    "width": 0,
    "height": 0,
    "gtype": "gage",
    "title": "",
    "label": "°C",
    "format": "{{value}}",
    "min": 0,
    "max": "80",
    "colors": [
      "#00b500",
      "#e6e600",
      "#ca3838"
    ]
  },
],

```

```

    "seg1": "",
    "seg2": "",
    "x": 537,
    "y": 224,
    "wires": []
  },
  {
    "id": "d9e5889d.3d18f8",
    "type": "ui_gauge",
    "z": "c78ab654.242b38",
    "name": "ambientTemp",
    "group": "75186061.6a979",
    "order": 0,
    "width": 0,
    "height": 0,
    "gtype": "gage",
    "title": "",
    "label": "°C",
    "format": "{{value}}",
    "min": 0,
    "max": "80",
    "colors": [
      "#00b500",
      "#e6e600",
      "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "x": 548,
    "y": 358,
    "wires": []
  },
  {
    "id": "f2482df6.d490f",
    "type": "ui_gauge",
    "z": "c78ab654.242b38",
    "name": "pressure",
    "group": "f9012aac.c516b8",
    "order": 0,
    "width": 0,
    "height": 0,
    "gtype": "gage",
    "title": "",
    "label": "hPa",
    "format": "{{value}}",
    "min": "300",
    "max": "1100",
    "colors": [
      "#00b500",
      "#e6e600",
      "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "x": 530,
    "y": 498,
    "wires": []
  },
  {

```

```

    "id": "36bd43cc.79321c",
    "type": "ui_gauge",
    "z": "c78ab654.242b38",
    "name": "ambientLight",
    "group": "c805fa77.a72e78",
    "order": 0,
    "width": 0,
    "height": 0,
    "gtype": "gage",
    "title": "",
    "label": "Lux",
    "format": "{{value}}",
    "min": 0,
    "max": "400",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "x": 541,
    "y": 651,
    "wires": []
},
{
    "id": "a8b1202b.593d6",
    "type": "ui_gauge",
    "z": "c78ab654.242b38",
    "name": "humidity",
    "group": "80fcb4a3.e9d128",
    "order": 1,
    "width": 0,
    "height": 0,
    "gtype": "donut",
    "title": "",
    "label": "%",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "x": 527,
    "y": 91,
    "wires": []
},
{
    "id": "f143e8f.49e5118",
    "type": "ui_chart",
    "z": "c78ab654.242b38",
    "name": "ambientLight",
    "group": "c805fa77.a72e78",
    "order": 0,
    "width": 0,

```

```

    "height": 0,
    "label": "",
    "chartType": "line",
    "legend": "false",
    "xformat": "HH:mm",
    "interpolate": "linear",
    "nodata": "",
    "dot": false,
    "ymin": "",
    "ymax": "",
    "removeOlder": "10",
    "removeOlderPoints": "",
    "removeOlderUnit": "60",
    "cutout": 0,
    "colors": [
      "#1f77b4",
      "#aec7e8",
      "#ff7f0e",
      "#2ca02c",
      "#98df8a",
      "#d62728",
      "#ff9896",
      "#9467bd",
      "#c5b0d5"
    ],
    "x": 541,
    "y": 700,
    "wires": [
      [],
      []
    ]
  },
  {
    "id": "87252df7.9d318",
    "type": "ui_chart",
    "z": "c78ab654.242b38",
    "name": "objectTemp",
    "group": "df2bb508.6bb7f8",
    "order": 0,
    "width": 0,
    "height": 0,
    "label": "",
    "chartType": "line",
    "legend": "false",
    "xformat": "HH:mm:ss",
    "interpolate": "linear",
    "nodata": "",
    "dot": false,
    "ymin": "",
    "ymax": "",
    "removeOlder": "10",
    "removeOlderPoints": "",
    "removeOlderUnit": "60",
    "cutout": 0,
    "colors": [
      "#1f77b4",
      "#aec7e8",
      "#ff7f0e",
      "#2ca02c",

```



```

        "#98df8a",
        "#d62728",
        "#ff9896",
        "#9467bd",
        "#c5b0d5"
    ],
    "x": 537,
    "y": 274,
    "wires": [
        [],
        []
    ]
}],
{
    "id": "a4b0eda3.4752",
    "type": "ui_chart",
    "z": "c78ab654.242b38",
    "name": "ambientTemp",
    "group": "75186061.6a979",
    "order": 0,
    "width": 0,
    "height": 0,
    "label": "",
    "chartType": "line",
    "legend": "false",
    "xformat": "HH:mm:ss",
    "interpolate": "linear",
    "nodata": "",
    "dot": false,
    "ymin": "",
    "ymax": "",
    "removeOlder": "10",
    "removeOlderPoints": "",
    "removeOlderUnit": "60",
    "cutout": 0,
    "colors": [
        "#1f77b4",
        "#aec7e8",
        "#ff7f0e",
        "#2ca02c",
        "#98df8a",
        "#d62728",
        "#ff9896",
        "#9467bd",
        "#c5b0d5"
    ],
    "x": 549,
    "y": 404,
    "wires": [
        [],
        []
    ]
}],
{
    "id": "ce147e28.058c6",
    "type": "ui_chart",
    "z": "c78ab654.242b38",
    "name": "pressure",
    "group": "f9012aac.c516b8",

```

```

"order": 0,
"width": 0,
"height": 0,
"label": "",
"chartType": "line",
"legend": "false",
"xformat": "HH:mm:ss",
"interpolate": "linear",
"nodata": "",
"dot": false,
"ymin": "",
"ymax": "",
"removeOlder": "10",
"removeOlderPoints": "",
"removeOlderUnit": "60",
"cutout": 0,
"colors": [
  "#1f77b4",
  "#aec7e8",
  "#ff7f0e",
  "#2ca02c",
  "#98df8a",
  "#d62728",
  "#ff9896",
  "#9467bd",
  "#c5b0d5"
],
"useOldStyle": true,
"x": 530,
"y": 548,
"wires": [
  [],
  []
]
},
{
  "id": "927a002d.0cda3",
  "type": "ui_chart",
  "z": "c78ab654.242b38",
  "name": "humidity",
  "group": "80fcb4a3.e9d128",
  "order": 2,
  "width": 0,
  "height": 0,
  "label": "",
  "chartType": "line",
  "legend": "false",
  "xformat": "HH:mm:ss",
  "interpolate": "linear",
  "nodata": "",
  "dot": false,
  "ymin": "",
  "ymax": "",
  "removeOlder": "10",
  "removeOlderPoints": "",
  "removeOlderUnit": "60",
  "cutout": 0,
  "colors": [
    "#1f77b4",

```

```

        "#aec7e8",
        "#ff7f0e",
        "#2ca02c",
        "#98df8a",
        "#d62728",
        "#ff9896",
        "#9467bd",
        "#c5b0d5"
    ],
    "useOldStyle": true,
    "x": 528,
    "y": 135,
    "wires": [
        [],
        []
    ]
}],
{
    "id": "f159e75b.b51bd8",
    "type": "ibmiot in",
    "z": "c78ab654.242b38",
    "authentication": "quickstart",
    "apiKey": "",
    "inputType": "evt",
    "deviceId": "987bf3121b48",
    "applicationId": "",
    "deviceType": "+",
    "eventType": "+",
    "commandType": "",
    "format": "json",
    "name": "cc3200stk",
    "service": "quickstart",
    "allDevices": false,
    "allApplications": false,
    "allDeviceTypes": true,
    "allEvents": true,
    "allCommands": false,
    "allFormats": false,
    "x": 100,
    "y": 89,
    "wires": [
        [
            "4423a4a.e6d285c",
            "cb65d904.bef908",
            "a00aee2e.50a0d",
            "951f4f76.be40f",
            "38572e76.3dbbd2"
        ]
    ]
}],
{
    "id": "719da0b6.506c58",
    "type": "rpi-gpio out",
    "z": "c78ab654.242b38",
    "name": "IN4 - Lamp",
    "pin": "40",
    "set": "",
    "level": "0",
    "freq": ""
}

```

```

        "out": "out",
        "x": 742,
        "y": 603,
        "wires": []
    },
    {
        "id": "f9d25499.a413a8",
        "type": "function",
        "z": "c78ab654.242b38",
        "name": "turn on/off",
        "func": "if (msg.payload < 10) {\n        msg.payload = 1;\n}
\nelse{\n        msg.payload = 0;\n}\n\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 540,
        "y": 603,
        "wires": [
            [
                "719da0b6.506c58"
            ]
        ]
    },
    {
        "id": "413565c0.ab02d4",
        "type": "rpi-gpio out",
        "z": "c78ab654.242b38",
        "name": "IN3 - Fan",
        "pin": "38",
        "set": "",
        "level": "0",
        "freq": "",
        "out": "out",
        "x": 725,
        "y": 181,
        "wires": []
    },
    {
        "id": "b0846e6e.e2d46",
        "type": "rpi-gpio out",
        "z": "c78ab654.242b38",
        "name": "IN2 - Heater",
        "pin": "36",
        "set": "",
        "level": "0",
        "freq": "",
        "out": "out",
        "x": 738,
        "y": 316,
        "wires": []
    },
    {
        "id": "51ec1d9d.6908bc",
        "type": "rpi-gpio out",
        "z": "c78ab654.242b38",
        "name": "IN1 - Moisturizer Machine",
        "pin": "32",
        "set": "",
        "level": "0",
        "freq": "",

```

```

        "out": "out",
        "x": 772,
        "y": 44,
        "wires": []
    },
    {
        "id": "2a1f1680.c3fc2a",
        "type": "function",
        "z": "c78ab654.242b38",
        "name": "turn on/off",
        "func": "if (msg.payload < 20) {\n        msg.payload = 1;\n}
\nelse{\n        msg.payload = 0;\n}\n\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 539,
        "y": 316,
        "wires": [
            [
                "b0846e6e.e2d46"
            ]
        ]
    },
    {
        "id": "91c8fc58.2643e",
        "type": "function",
        "z": "c78ab654.242b38",
        "name": "turn on/off",
        "func": "if (msg.payload > 30) {\n        msg.payload = 1;\n}
\nelse{\n        msg.payload = 0;\n}\n\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 537,
        "y": 181,
        "wires": [
            [
                "413565c0.ab02d4"
            ]
        ]
    },
    {
        "id": "80f6d802.72e2c8",
        "type": "function",
        "z": "c78ab654.242b38",
        "name": "turn on/off",
        "func": "if (msg.payload < 30) {\n        msg.payload = 1;\n}
\nelse{\n        msg.payload = 0;\n}\n\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 535,
        "y": 45,
        "wires": [
            [
                "51ec1d9d.6908bc"
            ]
        ]
    },
    {
        "id": "4977d484.a41744",
        "type": "ui_switch",

```

```

    "z": "c78ab654.242b38",
    "name": "Lamp",
    "label": "Lamp",
    "group": "4ae148b.6dd8cb8",
    "order": 1,
    "width": 0,
    "height": 0,
    "passthru": true,
    "decouple": "false",
    "topic": "",
    "style": "",
    "onvalue": "1",
    "onvalueType": "num",
    "onicon": "",
    "oncolor": "",
    "offvalue": "0",
    "offvalueType": "num",
    "officon": "",
    "offcolor": "",
    "x": 945,
    "y": 662,
    "wires": [
      [
        "1131f36b.297895"
      ]
    ]
  },
  {
    "id": "138f97ab.34b59",
    "type": "ui_switch",
    "z": "c78ab654.242b38",
    "name": "Fan",
    "label": "Fan",
    "group": "4ae148b.6dd8cb8",
    "order": 2,
    "width": 0,
    "height": 0,
    "passthru": true,
    "decouple": "false",
    "topic": "",
    "style": "",
    "onvalue": "1",
    "onvalueType": "num",
    "onicon": "",
    "oncolor": "",
    "offvalue": "0",
    "offvalueType": "num",
    "officon": "",
    "offcolor": "",
    "x": 944,
    "y": 229,
    "wires": [
      [
        "af477c76.230d78"
      ]
    ]
  },
  {
    "id": "9c620ad.0150f78",

```

```

    "type": "ui_switch",
    "z": "c78ab654.242b38",
    "name": "Moisturizer Machine",
    "label": "Moisturizer Machine",
    "group": "4ae148b.6dd8cb8",
    "order": 3,
    "width": 0,
    "height": 0,
    "passthru": true,
    "decouple": "false",
    "topic": "",
    "style": "",
    "onvalue": "1",
    "onvalueType": "num",
    "onicon": "",
    "oncolor": "",
    "offvalue": "0",
    "offvalueType": "num",
    "officon": "",
    "offcolor": "",
    "x": 990,
    "y": 106,
    "wires": [
      [
        "7ad1bc55.0a05cc"
      ]
    ]
  },
  {
    "id": "1131f36b.297895",
    "type": "rpi-gpio out",
    "z": "c78ab654.242b38",
    "name": "IN4 - Lamp",
    "pin": "40",
    "set": "",
    "level": "0",
    "freq": "",
    "out": "out",
    "x": 1201,
    "y": 663,
    "wires": []
  },
  {
    "id": "af477c76.230d78",
    "type": "rpi-gpio out",
    "z": "c78ab654.242b38",
    "name": "IN3 - Fan",
    "pin": "38",
    "set": "",
    "level": "0",
    "freq": "",
    "out": "out",
    "x": 1190,
    "y": 229,
    "wires": []
  },
  {
    "id": "666ce049.ba035",
    "type": "rpi-gpio out",

```

```

    "z": "c78ab654.242b38",
    "name": "IN2 - Heater",
    "pin": "36",
    "set": "",
    "level": "0",
    "freq": "",
    "out": "out",
    "x": 1202,
    "y": 366,
    "wires": []
  },
  {
    "id": "7ad1bc55.0a05cc",
    "type": "rpi-gpio out",
    "z": "c78ab654.242b38",
    "name": "IN1 - Moisturizer Machine",
    "pin": "32",
    "set": "",
    "level": "0",
    "freq": "",
    "out": "out",
    "x": 1238,
    "y": 106,
    "wires": []
  },
  {
    "id": "f95f15c7.635ac",
    "type": "ui_switch",
    "z": "c78ab654.242b38",
    "name": "Heater",
    "label": "Heater",
    "group": "4ae148b.6dd8cb8",
    "order": 4,
    "width": 0,
    "height": 0,
    "passthru": true,
    "decouple": "false",
    "topic": "",
    "style": "",
    "onvalue": "1",
    "onvalueType": "num",
    "onicon": "",
    "oncolor": "",
    "offvalue": "0",
    "offvalueType": "num",
    "officon": "",
    "offcolor": "",
    "x": 944,
    "y": 366,
    "wires": [
      [
        "666ce049.ba035"
      ]
    ]
  }
]

```