

Department of Electronic & Telecommunication Engineering

University of Moratuwa

EN 2560 – Internet of Things Design and Competition



Course Project Report

Group Number : 09

Group Members :

<u>Name</u>	<u>Index Number</u>
Udara A.W.T.	180650P
Sewwandi B.L.P.N.	180589K
Rathnayaka R.G.H.V.	180529E

July 8, 2021

This Report is submitted as a partial fulfillment of the requirements for the module EN2560 – Internet of Things Design and Competition.

Department of Electronic and Telecommunication Engineering

University of Moratuwa

Problem intended to be solved

As a citizen living in a society, we all have to be updated with the news revolving around us. But it follows a priority for each citizen. Something happens at a corner of world may take a less importance to a someone in the other corner of the world. So, it is important to have a priority or a nice sort out of things happening around us. This is an implementation on **sorting out news we want from thrusted sources**. As mobile phones have become a versatile device now, it is so easy if we can get trusted news to our mobile. The traditional news that we are following through social media comprises of low credibility. Some of them might be useless for a certain one. So, through this implementation we are able to **receive news from different thrusted sources by taking precedence according to popularity and get sorted as the client's wish to the own mobile phone**. Implementations were done using IoT concepts, standards, technologies and tools.

System overview and operation

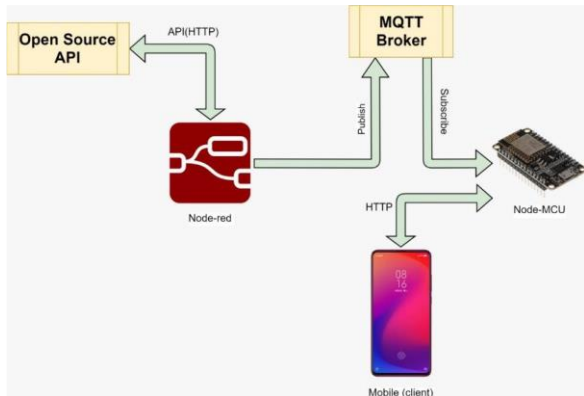


Figure 01– Block diagram of system overview

Overall system development is based on Node Red and Arduino. Here as the data source, we are using an open source API called **News API** through a HTTP request as our data source. More information on this API is available at : <https://newsapi.org/docs/get-started>. **Node red** will provide the user with a user friendly dashboard which enhances the ability of sorting out news according to the user's desire. It will process the data received from the API and send then towards the **Node MCU** via a **Message Queuing Telemetry Transport (MQTT) broker**. Here <http://www.mqtt-dashboard.com/> is used as the MQTT broker which establishes the connection in between Node red and Node MCU. Node MCU will act as a client in this publish subscribe service. Through number of computations, it will retrieve the processed news to the mobile phone in a user friendly manner which can be accessed by local IP Address of the client. The *Figure 01* above depicts the block diagram of the system overview.

The system provides a user friendly dashboard where the client can select the mode of his/her favour with news through number of selections. This can be handled by the client or someone assigned by the client who is remotely residing with the client through a PC. Manipulation of user requirements can be even done by even a mobile device as we are uploading this Node red simulations in a cloud environment. After submitting user preferences, the user will able to receive the **Top 5 news** to his mobile phone. This can be accessed by the link of Local IP Address received as a text message to the mobile number. This link will redirect the client toward a HTML page depicting the top 5 news of the selected category.

Implementation

The system implementation is almost all a software implementation which needs only a **Node MCU (ESP8266 module)** as a **hardware** device rather than a **PC** and a **mobile phone**. Here, Node MCU act as a client which subscribes the data from MQTT broker.

When considering on the **software** implementation, we used **Node red** and **Arduino 1.8.13** as software. Initial implementations are done with Node red. Node red will receive the required data from the open API data source through a **HTTP (Hyper Text Transfer Protocol) request**. The required URL and API key were obtained from the **News API** website. This was the basic **online resource** that we used. We can get different types of news by making small changes to the URL. The News API website explains how to use the URL. Node red flows will implement a user friendly interface where the client will able to input his preferences. We have provided 7 ways to get news in our Node-

RED dashboard. Out of those 7 ways, the user has to activate the way he/she wants to get news. The *Figure 02* below shows the initial interface of the Node-RED dashboard before activating the way.

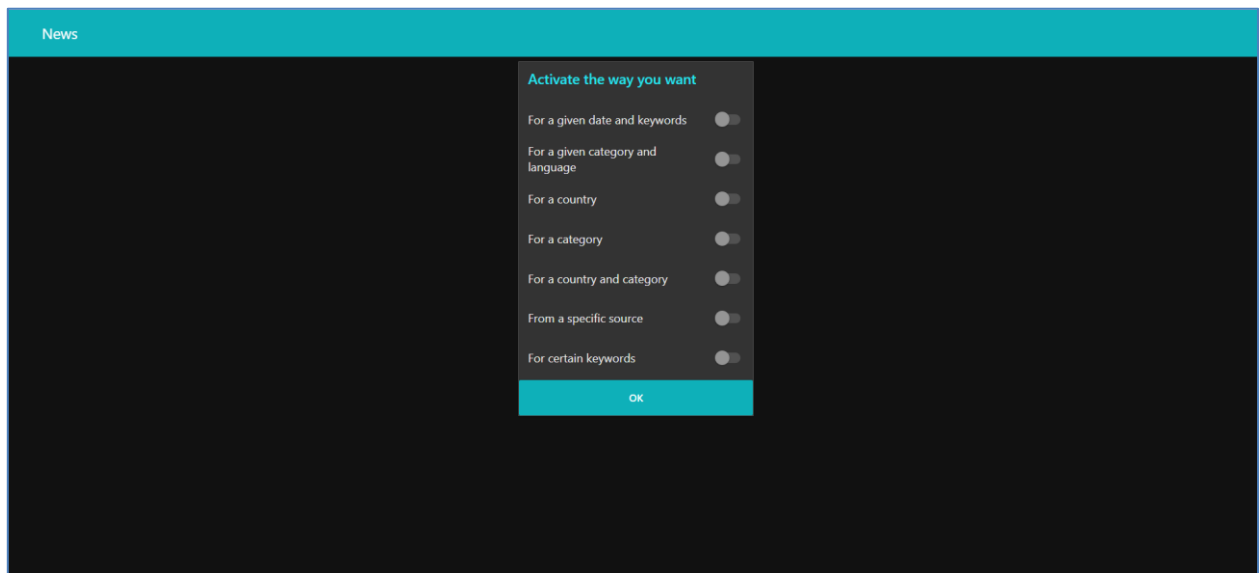


Figure 02 – User interface of the Node-RED dashboard before activating the way

From these input criteria node red flows are arranged as suit to retrieve the most suitable five output news items from the open API source. In a Node-RED flow, we create the URL as a string according to the requirements using the function nodes. Then that URL string is converted to the URL of the http request node using change nodes. Then the news is taken using http request node. The first five news are selected using function node and then sent to the Node MCU via Message Queuing Telemetry Transport (MQTT). The part of node red flow designs required to activate the way according to the user are given below under *Figure 03*.

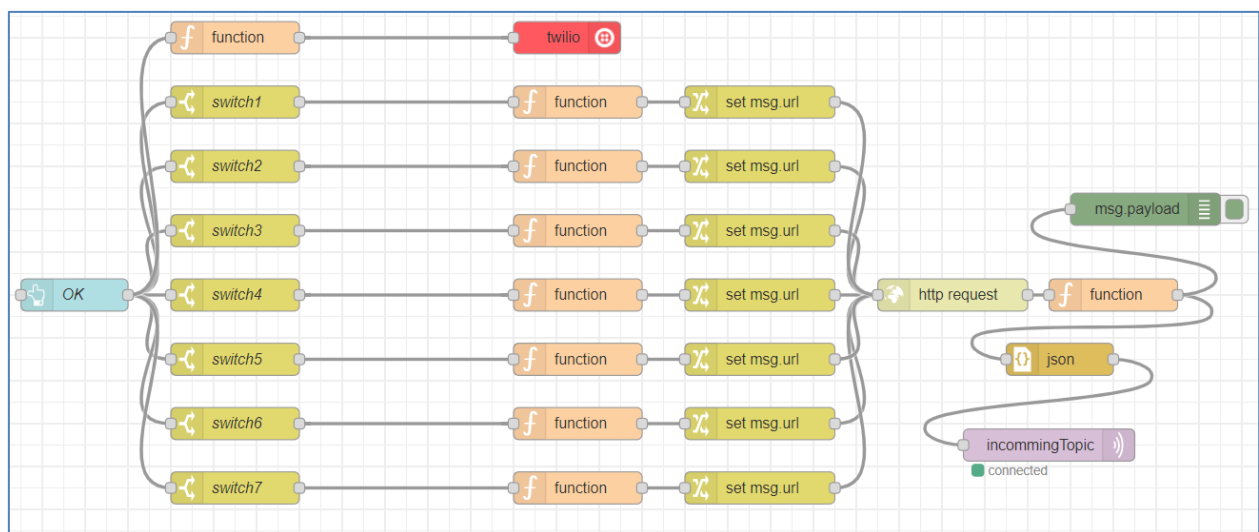


Figure 03 – The part of Node red flows which activates the way according to the user

In order to notify the client with the news, we are using a system where we can send a **SMS to the user's mobile number** indicating the **Local IP Address of him/her** which is the front end of the output of this platform. To facilitate this, we are using **Twilio**, which is a good **cloud communication platform** which allows to send and receive text messages programmatically. Flows required for this are also have been added to the node red flows.

The array of five news items received is sent through the <http://www.mqtt-dashboard.com/> which acts as a **MQTT Broker** to the Node MCU. MQTT Broker will facilitate the publish subscribe service between Node red and Node MCU. "Broker.mqttdashboard.com" act as the

MQTT server for the Node MCU and the array is received as a string. Node MCU is connected to a local area network and the back end of this process is done through **Arduino 1.8.13** software. Then, the received string of array which contains top 5 news item is processed with simple Arduino coding to separate their values as they are received as a **string of an array comprise of Json**. In order to handle Jsons we used **ArdiunoJson library**. The coding segments for deserialization of Json data is shown in the *Figure 04* beside.

Afterwards, HTTP server is initiated at the local host to represent these variables in a webpage. Required coding parts are depicted in the *Figure 05* below.

```
// Start LMNR responder
LMNR.begin("esp8266");
Serial.println("LMNR responder started");
// Start HTTP server
server.on("/", 1) {}
String page = "";
page += "<!DOCTYPE html>\n";
page += "<html>\n";
page += "<body>\n";
page += "<h1 style='font-size:300%;text-align:center;'>" + (String)title[0] + "</h1>\n";
page += "<h2>" + (String)name[0] + " - " + (String)author[0] + "</h2>\n";
```

Figure 05 – Code segment for starting a HTTP server and assigning data for the website

```
page += "<h2>" + (String)name[1] + " - " + (String)author[1] + "</h2>\n";
page += "<p style='font-size:100%;>" + (String)description[1] + "</p>\n";
page += "<p><a href='\" + (String)url[1] + \"'>Click here for more details!</a></p>\n";
page += "<img src='\" + (String)urlToImage[1] + \"' width='500' height='400'>\n";
page += "<h1 style='font-size:300%;text-align:center;'>" + (String)title[2] + "</h1>\n";
page += "<h2>" + (String)name[2] + " - " + (String)author[2] + "</h2>\n";
page += "<p style='font-size:100%;>" + (String)description[2] + "</p>\n";
page += "<p><a href='\" + (String)url[2] + \"'>Click here for more details!</a></p>\n";
page += "<img src='\" + (String)urlToImage[2] + \"' width='500' height='400'>\n";
page += "<h1 style='font-size:300%;text-align:center;'>" + (String)title[3] + "</h1>\n";
page += "<h2>" + (String)name[3] + " - " + (String)author[3] + "</h2>\n";
page += "<p style='font-size:100%;>" + (String)description[3] + "</p>\n";
page += "<p><a href='\" + (String)url[3] + \"'>Click here for more details!</a></p>\n";
page += "<img src='\" + (String)urlToImage[3] + \"' width='500' height='400'>\n";
page += "<h1 style='font-size:300%;text-align:center;'>" + (String)title[4] + "</h1>\n";
page += "<h2>" + (String)name[4] + " - " + (String)author[4] + "</h2>\n";
page += "<p style='font-size:100%;>" + (String)description[4] + "</p>\n";
page += "<p><a href='\" + (String)url[4] + \"'>Click here for more details!</a></p>\n";
page += "<img src='\" + (String)urlToImage[4] + \"' width='500' height='400'>\n";
page += "</body>\n";
page += "</html>\n";
server.send(200, "text/html", page);
});
server.begin();
Serial.println("HTTP server started");
}
```

```
Final_Arduino | Arduino 1.8.13
File Edit Sketch Tools Help

Final_Arduino $
50
51 void callback(char* topic, byte* payload, unsigned int length) {
52   char str[length+1];
53   int i = 0;
54   for (int i = 0; i < length; i++) {
55     str[i] = (char)payload[i];
56     Serial.print(str[i]);
57   }
58   str[i] = 0;
59   Serial.println();
60   StaticJsonDocument <32768> doc;
61   deserializeJson(doc, payload);
62   JsonArray arr = doc.as<JsonArray>();
63   int j = 0;
64   for (JsonObject repo : arr) {
65     id[j] = (const char*)repo["source"]["id"];
66     name[j] = (const char*)repo["source"]["name"];
67     author[j] = (const char*)repo["author"];
68     title[j] = (const char*)repo["title"];
69     description[j] = (const char*)repo["description"];
70     url[j] = (const char*)repo["url"];
71     publishedAt[j] = (const char*)repo["publishedAt"];
72     urlToImage[j] = (const char*)repo["urlToImage"];
73     j++;
74   }
75   for (int k = 0; k < 5; k++) {
76     if (name[k] == NULL)
77       name[k] = "Source is not given";
78     if (author[k] == NULL)
79       author[k] = "Anonymous";
80     if (title[k] == NULL)
81       title[k] = "No Title";
82     if (description[k] == NULL)
83       description[k] = " ";
84     if (url[k] == NULL)
85       url[k] = " ";
86     if (urlToImage[k] == NULL)
87       urlToImage[k] = " ";
88     Serial.println("");
89     Serial.println(name[k]);
90     Serial.println(author[k]);
91     Serial.println(title[k]);
92     Serial.println(description[k]);
93     Serial.println(url[k]);
94     Serial.println(urlToImage[k]);
95   }
96 }
```

Figure 04 – code segment for deserialization Json Data and assigning them to arrays

By uploading the whole Arduino code to the ESP8266 Module (Node MCU) it is able to see a webpage has been constructed at client's local IP Address which depicts the top five news for the desired need that he/she has provided to the Node red dashboard.

System functionality achieved

Once the user has activated the desired mode, the user has ability to enter the required inputs. For an example, user can select which category of news in which country he/she wants. The dashboard appearance for this scenario is clearly visible on the *Figure 06* below.

News

Activate the way you want

For a given date and keywords

For a given category and language

For a country

For a category

For a country and category

From a specific source

For certain keywords

OK

Categories

Entertainment

Country

New Zealand

Figure 06 – Scenario of activating news of certain category from a certain country.

After pressing the “OK” button, a link which represents the local IP Address of the client is received to the user’s mobile number as a text message. The user can get redirected to the HTML webpage that represents the news by clicking that link. We only give the first five pieces of news to the user at a time. The prototype message received to a mobile number is given beside in *Figure 07*.

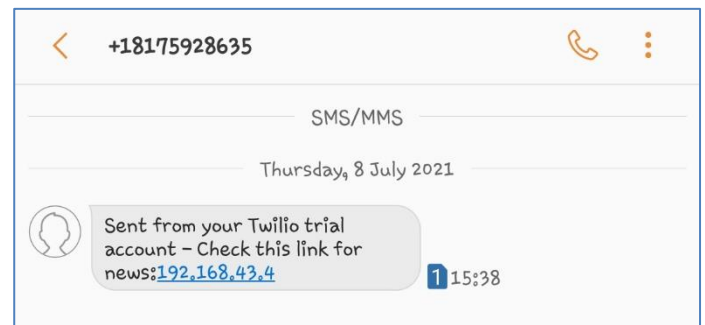


Figure 07 – SMS Notification received to the mobile number through Twilio

The HTML page designed at local IP address will represent the Top 5 news items according to the wish of the client or someone assigned by the client expressed through node red dashboard. By implementing the node red flows in **IBM Cloud**, the client will be able to input his/her news desires even through mobile phone also.

A sample output received by activating **For country and category** and then **New Zealand** as the **country** and **Entertainment** as the **category**, we can receive the five news items as below from the mobile phone. The screenshots taken by the mobile phone are illustrated below in *Figure 08*.

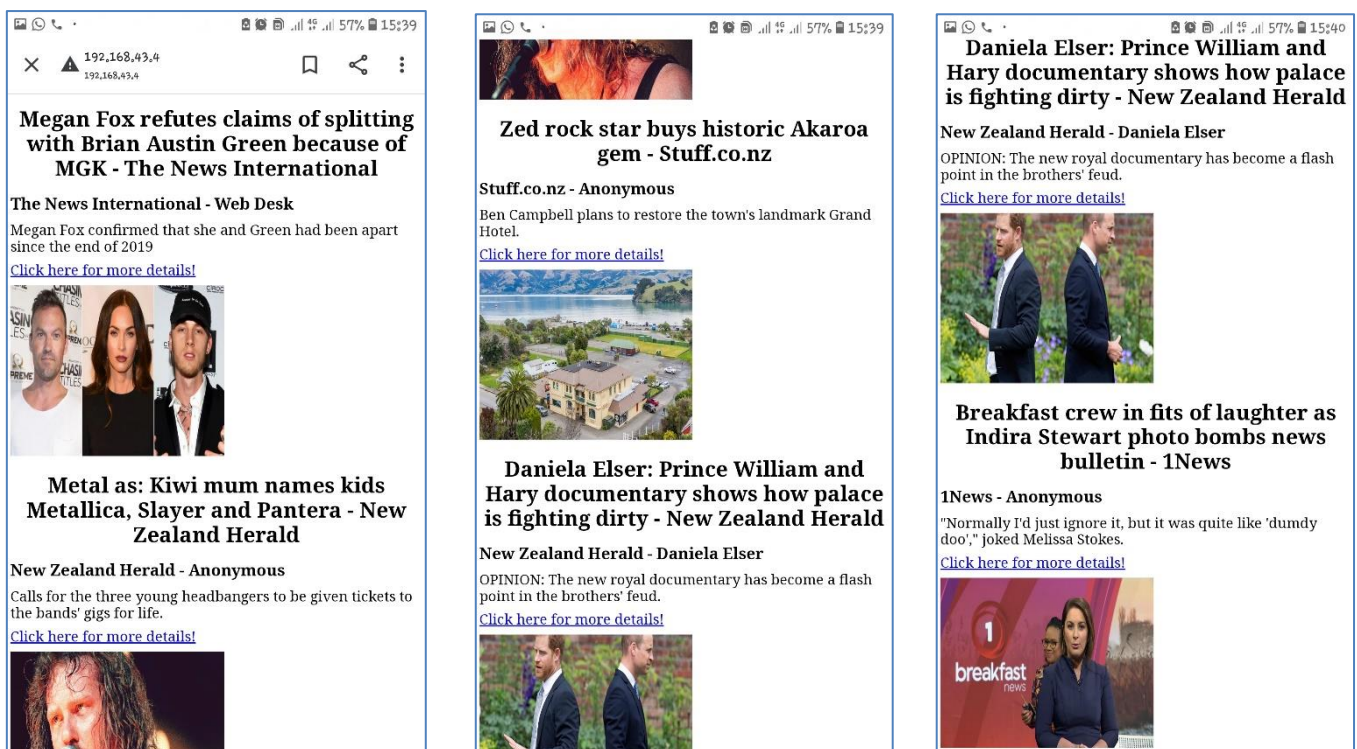


Figure 08 – Screenshots of the news window created at the received link for the given user requirements (Entertainment category in New Zealand)

Same as above, Top 5 news list of any user requirement can be received by client’s smart phone in a short while.

References :

- Get started - Documentation - News API, “Get started - Documentation - News API,” *Newsapi.org*, 2021. <https://newsapi.org/docs/get-started> (accessed Jul. 05, 2021).
- “MQTT Dashboard,” *Mqtt-dashboard.com*, 2021. <http://www.mqtt-dashboard.com/> (accessed Jul. 05, 2021).
- BenoitBlanchon, “ArduinoJson: Efficient JSON serialization for embedded C++,” *ArduinoJson*, 2021. <https://arduinojson.org/> (accessed Jul. 06, 2021).
- “Twilio Cloud Communications | Web Service API for building Voice and SMS Applications,” *Twilio.com*, 2015. <https://www.twilio.com/console> (accessed Jul. 06, 2021).