SMART WATER-QUALITY MONITORING SYSTEM

PROJECT OBJECTIVE:

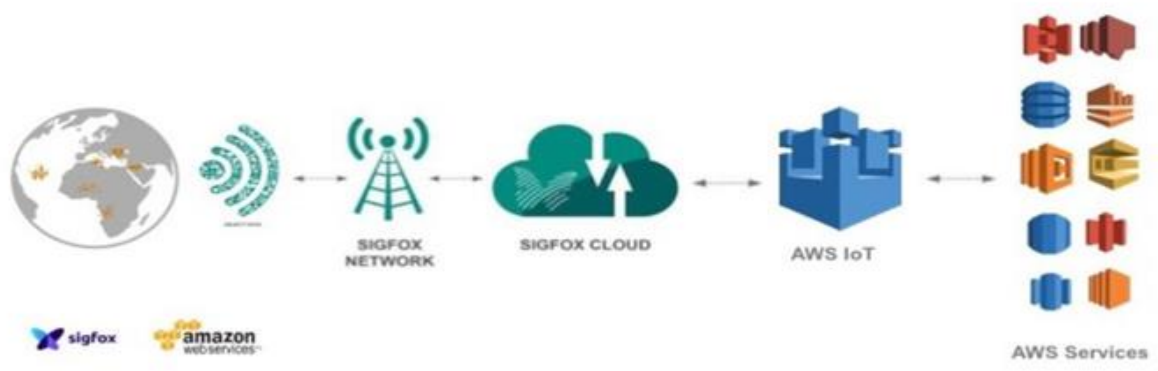
The goal of this project is to formulate a simpler solution for one of the most complex problems that is developing in the present. The main objective is to develop an inexpensive, scalable solution that allows remote monitoring and tracking of Water-Quality of large Water bodies. Even though there are many devices that are available in the market to measure specific water parameters like pH, temperature, etc. there is no commercially feasible product that gives a comprehensive quality measurement that allows for remote monitoring.

PROJECT DESCRIPTION:

The aim of this project is to develop a Smart Water Quality Monitoring System for testing the quality of water, using pH and turbidity as the parameters, and a User-Friendly Real-time Tracking Interface. The collective data is utilized to evaluate the quality of water, and to determine if the water quality is sufficient enough for domestic usage. The main goal is to develop a compact, cost-effective, bio-safe device to measure the appropriate parameters through the respective sensors, that uploads the compressed data to the cloud in a periodic basis, and to develop a Webpage that allows to monitor the quality levels on a real-time basis.

DESIGN OVERVIEW:

A combination of pH, turbidity sensors, and an algorithm is implemented that evaluates the quality of the water and a water level sensor is used to protect the device. Once, it’s evaluated, a packet is created with these data concatenated into a single data frame and transmitted via Sigfox module to the Sigfox cloud. Here, a number of callback functions are implemented to trigger a set of AWS IoT Rules, that perform a variety of actions including pushing data to the database, sending Email and SMS notifications to the Administrators and Subscribed users.



SYSTEM IMPLEMENTATION:

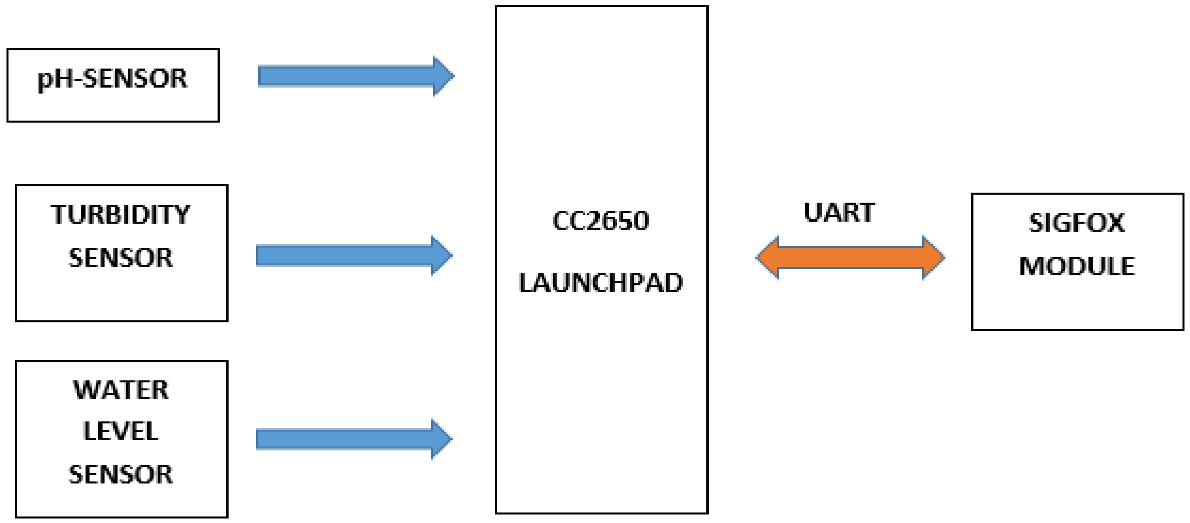
HARDWARE:

The hardware end of the project is implemented with 3 sensors, namely pH, turbidity and water-level indicator. The brain of the project is the CC2650 Launchpad, which integrates the data acquired by the sensors and feeds into the Sigfox network using Sigfox module.

The following is a list of above-mentioned components that are used in the building of the monitoring device and is accompanied with the specifications and the cost for corresponding components. It is followed by links to the vendors for all the components.

COMPONENTS	**MODEL NUMBER**	**COST**
Turbidity Sensor	SEN0189	$19.69
pH Sensor	YY6237	$33.59
Water-Level Indicator	MC050-5	$3.5
TI Launchpad	CC2650	$30
Sigfox Module	RC1692HP	$25
TOTAL		$111.78

Table 1



TURBIDITY SENSOR:

The turbidity sensor measures the quality of water by measuring its turbidity. The principle is that light is transmitted to detect particles that are suspended in water by measuring the scattering rate. This scattering rate changes with amount of Total Suspended Solids (TSS). Scattering rate is more when TSS is more, which in turn returns high voltage. If the liquid is clear, then there is no scattering and thus, low voltage is observed. Figure 1 shows the SEN0189 Turbidity sensor that has been integrated in the module.

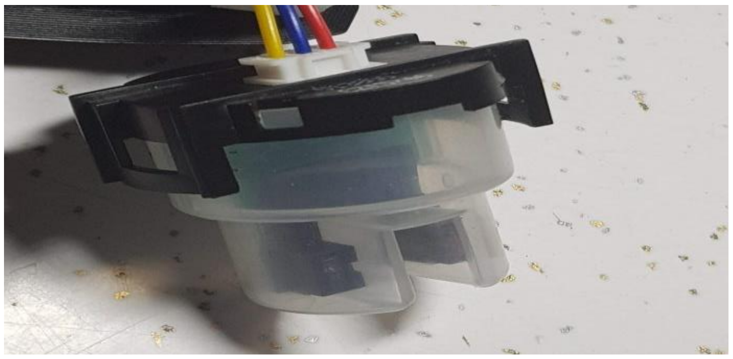


Fig. 1 Turbidity Sensor

Specifications:

Operating voltage	5V DC Supply
Operating Current	40mA (MAX)
Response Time	<500ms
Operating Temperature	5℃ - 90℃
Output Method	Analog Output (0 – 5V)
Special Instruction	Top of probe is not waterproof

From the figure 1, it is observed that 3 colored wires of the turbidity sensor namely yellow, blue and red are extended through a RMC cable and is connected to its module. This module in turn, communicates with the CC2650 Launchpad to provide the voltage for turbidity. The 3 wires are connected to their respective ports in the module.

Red	Pin 1
Blue	Pin 2
Yellow	Pin 3

These wire connections are static and shouldn't be interchanged in their places or removed from circuit.

Connections:

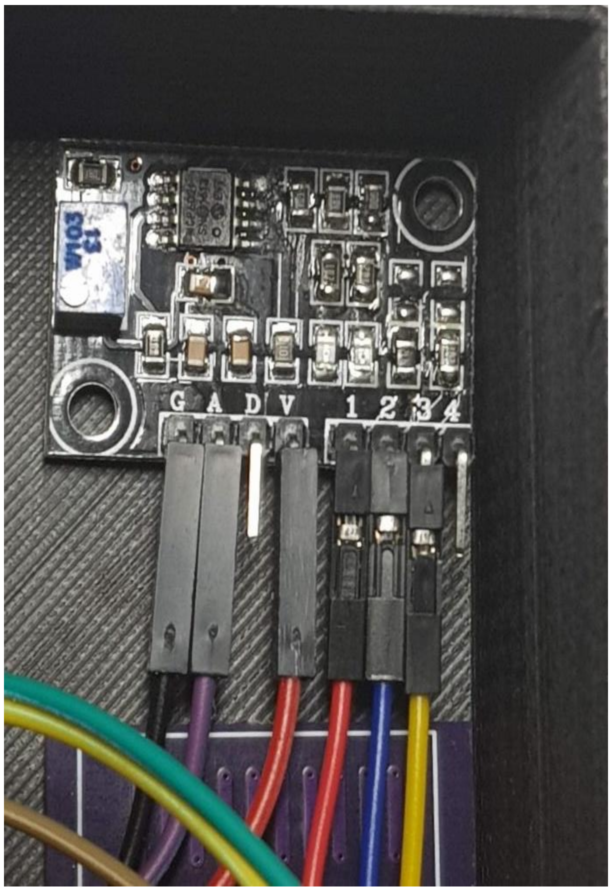


Fig. 2 Turbidity Sensor Module

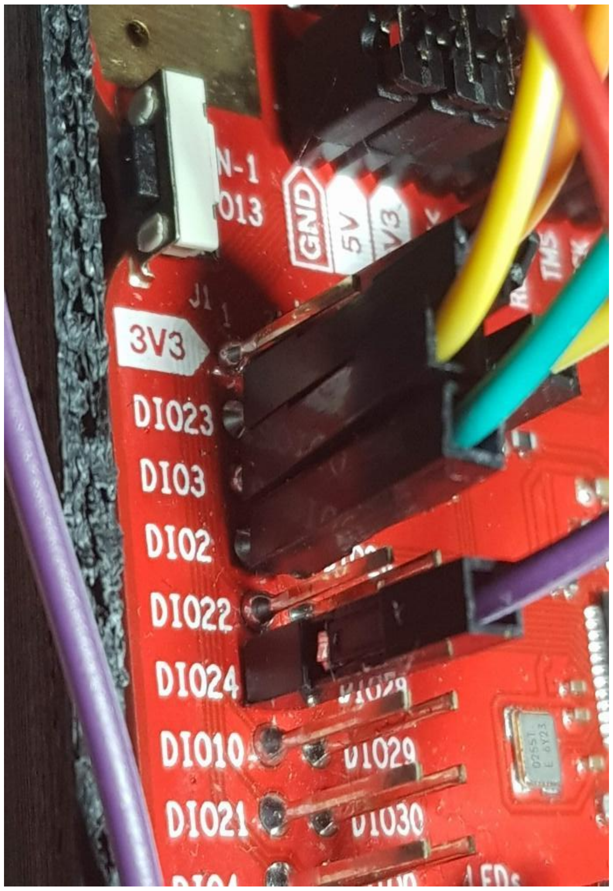


Fig 3. Connection to CC2650 Launchpad

Module	**CC2650 Launchpad**
G → Ground	GND
A → Analog o/p	DIO24
V → Voltage	5V

pH SENSOR:

pH is an indication of the concentration of Hydrogen ions present in a given solution. This can be quantified by using a pH sensor, which has 2 electrodes already embedded in it. The first one is a reference electrode (typically Ag/AgCl) and the other electrode is sensitive to the hydrogen ions. The potential difference created by these electrodes in the given solution is used to estimate the pH of that solution. This is the working principle of the pH sensor. pH scale ranges from 0 – 14 to estimate the acidity or alkalinity, where 0~7 denotes acidic solutions and 7~14 denotes basic solutions. pH value of 7 is considered to be neutral.

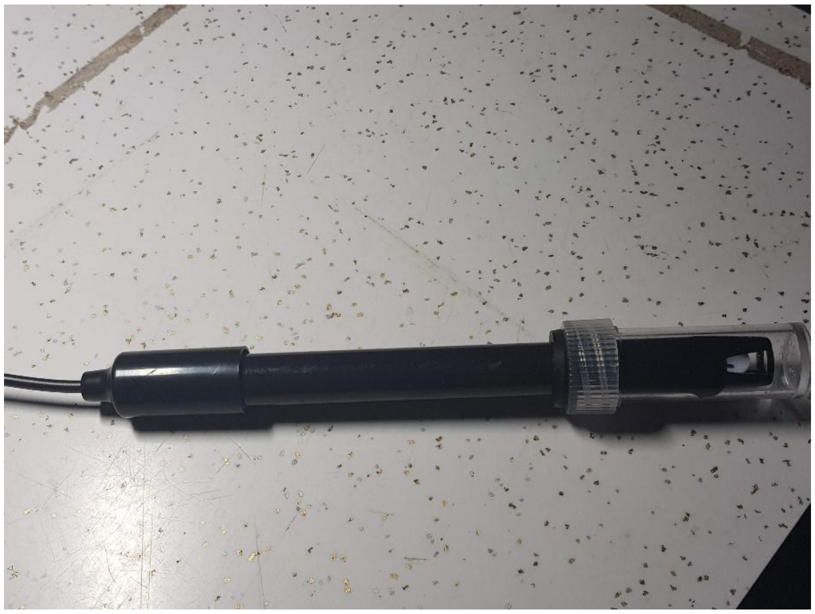


Fig. 4 pH sensor

Probe Specifications:

Electrode Type	**pH Range**	**Temp**	**Zero Point**	**Response Time**
65-1	0-14	0-80℃	7 ± 1	< 2 minutes

Circuit Specifications:

Supply Voltage	5V
Current	5 – 10mA
Consumption	≤ 0.5W
Working Temperature	10-50℃
Green LED	Indication for Power

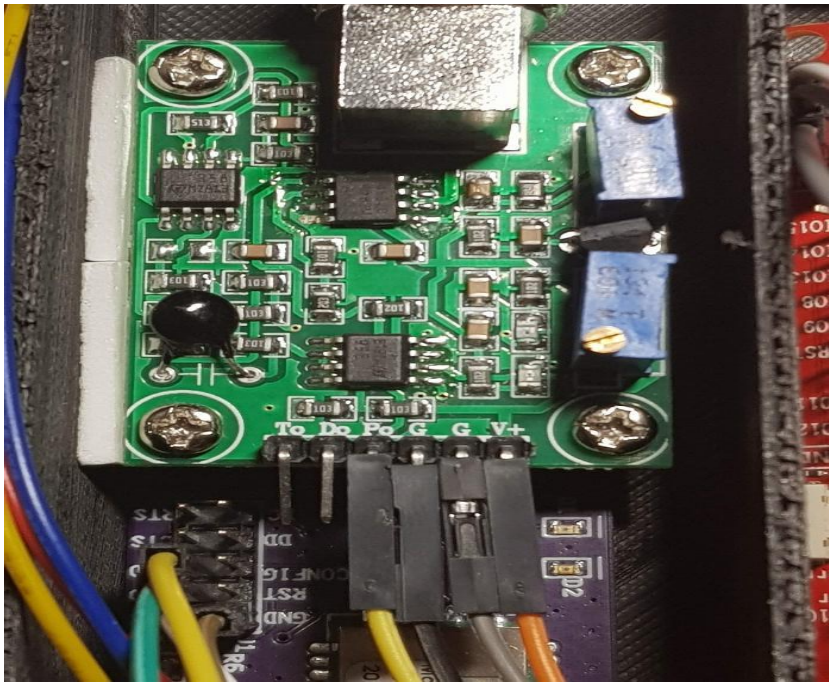


Fig. 5 pH Sensor Module

<u>Sensor Calibration:</u>

The pH sensor has to be handled carefully while immersing in solutions. The sensor takes a min or two to accurately display the voltage. In order to check or calibrate the sensor to a known solution of certain concentration, buffer powders can be used which are commercially available in the market in the pH values of 4.00 and 6.86. They have to be mixed with distilled water before the probe is inserted into the solution.

Since the probe returns only voltages on the console, it is necessary to take two readings of the voltage during calibration: one voltage using pH 4.0 solution and second voltage using 6.86 pH solution. The voltages are fed to the 2-point form,

$$\mathbf{y = mx + b}$$

in order to deduce the equation for converting the measured voltage to pH. Here, y is the pH value and x is the voltage.

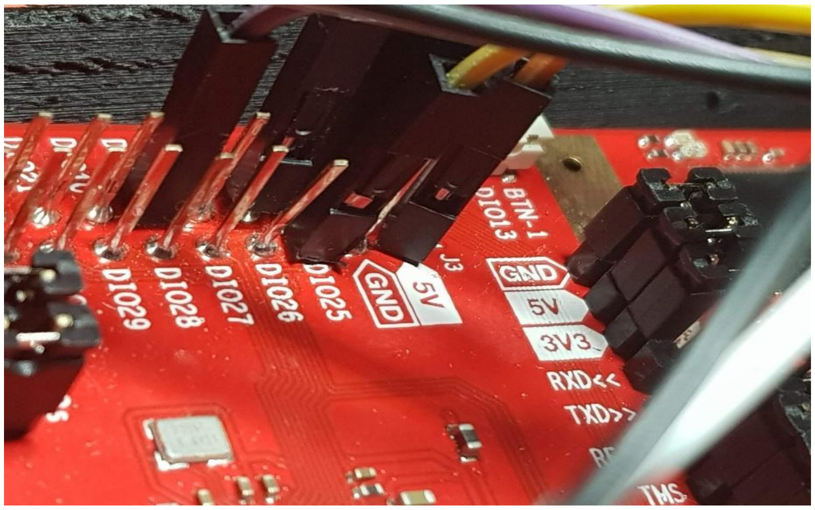


Fig. 6 Connection to CC2650 Launchpad

Connections:

Module	CC2650 Launchpad
V+ → Supply voltage	5V
G → Ground	GND
G → Ground	GND
Po → Analog o/p	DIO25

WATER LEVEL SENSOR:

The water level sensor is used in the project in order to make sure the test water doesn't cross the critical limit, because beyond this point the circuitry is not water compatible. Filaments in parallel are exposed on the body of the sensor, which sense the level of water when dipped into it. Based on the height of water, a parameter is sent through the 'S' pin of the water level sensor to the analog DIO23 pin of the Launchpad. It is then converted into equivalent voltage, which can be coded in a way suitable for the 3D printed design. Figure 7 shows the MC050-5 water level sensor.

Specifications:

Supply Voltage	5V DC Supply
Operating Current	< 20mA
Output Method	Analog
Detection/Sensing Area	40mm x 16mm
Operating Temperature	10°C - 30°C
Operating Humidity	10% - 90% non-condensing
Product Dimensions	65mm x 20mm x 8mm

As shown in Figure 3, the output of the water level sensor is connected to the DIO23 pin of CC2650 Launchpad, while the '+' and '-' pins of the sensor indicate voltage and ground respectively.

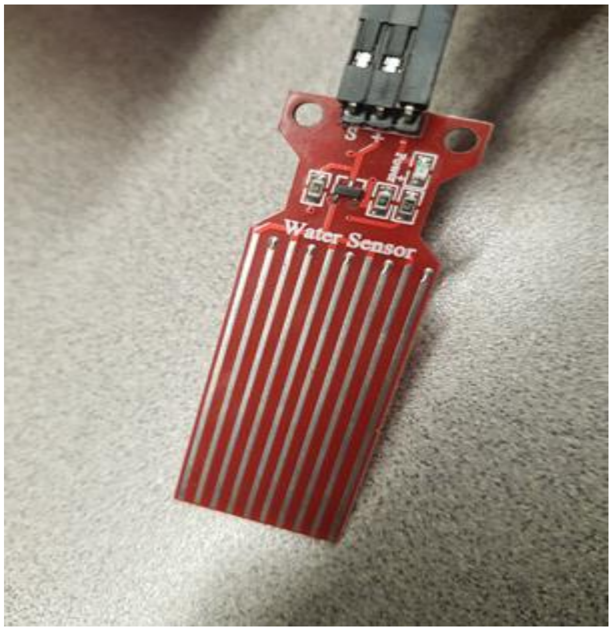


Fig. 7 Water-Level Indicator Sensor

Connections:

Water-Level Sensor	CC2650 Launchpad
S → Analog Signal	DIO23
+ → Voltage supply	5V
- → Ground	GND

TEXAS INSTRUMENTS CC2650 LAUNCHPAD:

CC2650 Launchpad is ideal for the project as it aims at Low Power Wide Area Networks (LPWAN). It has a built-in 32-bit ARM processor and runs at the frequency of 48MHz. A sensor controller is embedded in the device which is an ultra-low power sensor and is best suitable for interfacing physical sensors to collect both analog and digital values and puts the rest of the system in sleep mode. Also, this device supports a variety of protocols such as SPI, I2C to name a few.

Specifications:

Core	ARM Cortex M3
RF Frequency	2.4 GHz
Interface	GPIO, I2C, SPI, UART, USB
Operating Voltage	3.3V, 5V
Operating Temperature	-25°C to 70°C
Data Bus Width	32 bit

The pH, turbidity and the water-level sensors are interfaced to the CC2650 Launchpad in their corresponding pin positions. ADC conversion is used to convert the raw values read out by these water-environment sensors and are meant to present in a readable and perceptible manner for understanding. The CC2650 Launchpad has pins projected on top and bottom parts of the board for convenient interface. This feature is helpful in the project as all the 3 sensors use 5V supply while one side of the board (top or bottom) has only 2 pins for 5V supply.

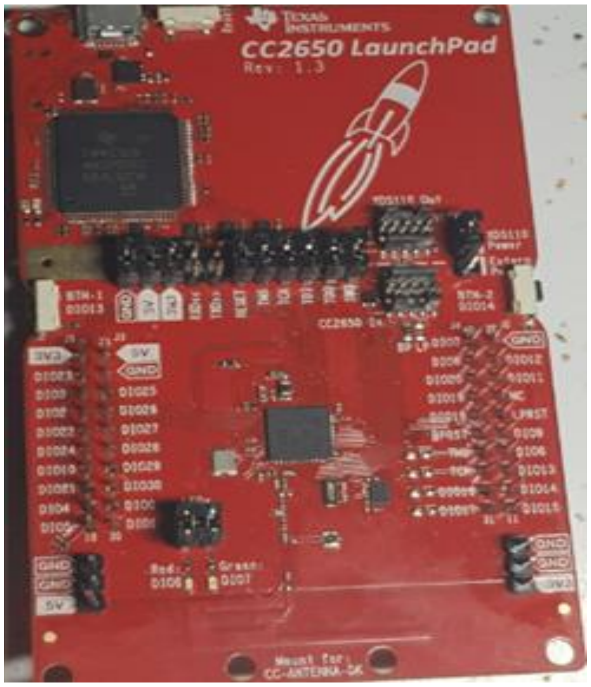
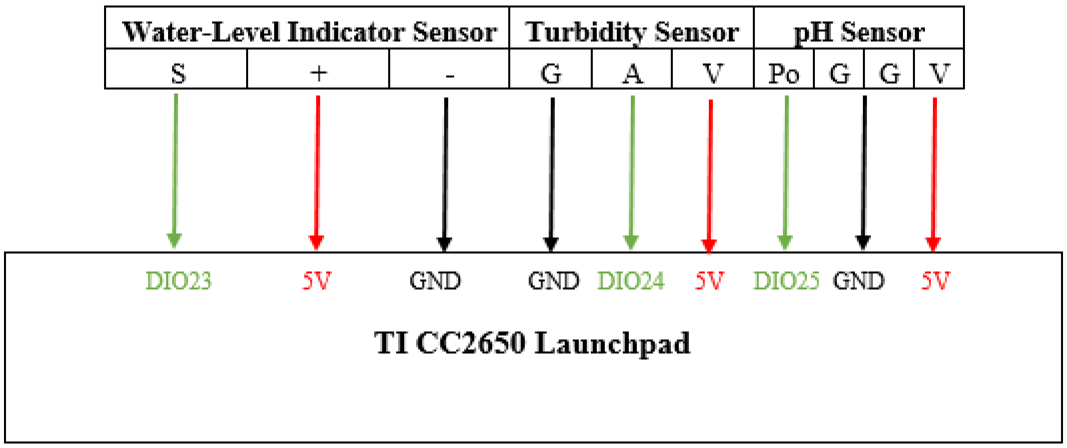


Fig 8. CC2650 Launchpad



SIGFOX:

Sigfox is a Low Powered Wide Area Network (LPWAN) which acts as a provider for Internet of Things (IoT) applications. Key aspects of Sigfox are:

- Low power - Power consumption is minimal as most of the assets are unpowered.
- Long range - The coverage with one station ranges up to 10 miles.
- Ease of use - Integration of assets and backend.
- Interference - Highly resistant

It uses Ultra Narrow Band BPSK (Binary Phase Shift Keying) modulation. Each Sigfox device also has its own ID which is unique from all other Sigfox devices. Synchronization is not required in a Sigfox network, thus jamming the receiver has no effect practically.

In our project, the sensors collect all the data which is integrated in the Launchpad. Then, a connection is established with the Sigfox network with various commands, including the device ID for authentication. When the connection is successfully established, the data is pushed to the Sigfox base station, which in turn delivers the data to the Sigfox cloud. The data can be viewed in the backend.

Figure 10 shows the Sigfox Module that was used in our project. Since it was designed by the EE Department of UB itself, the cost is comparatively inexpensive than the original Sigfox device.

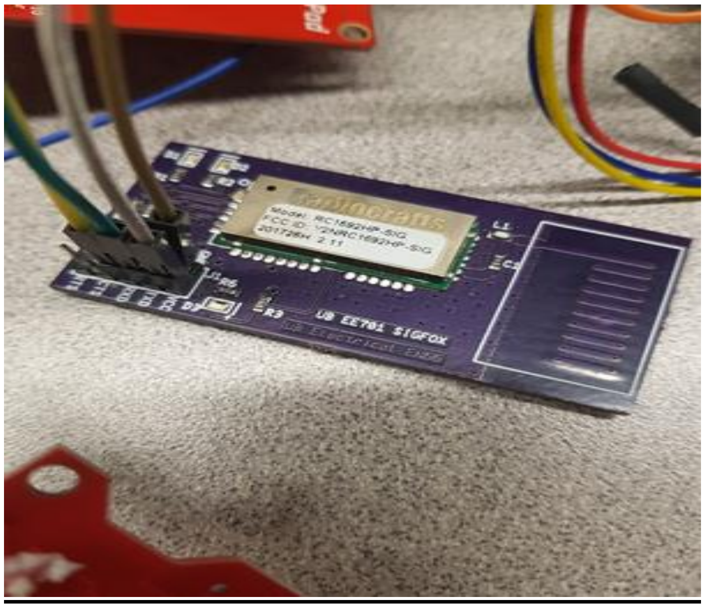
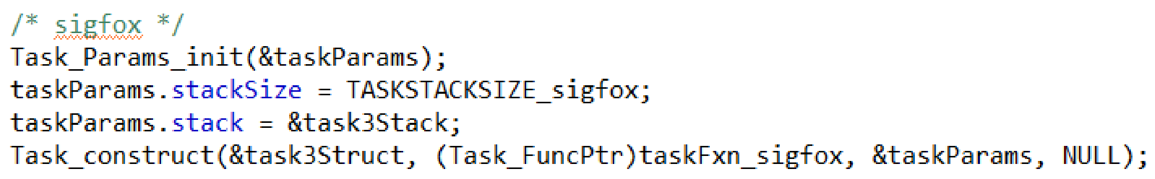


Fig. 10 Sigfox Module

SOFTWARE:

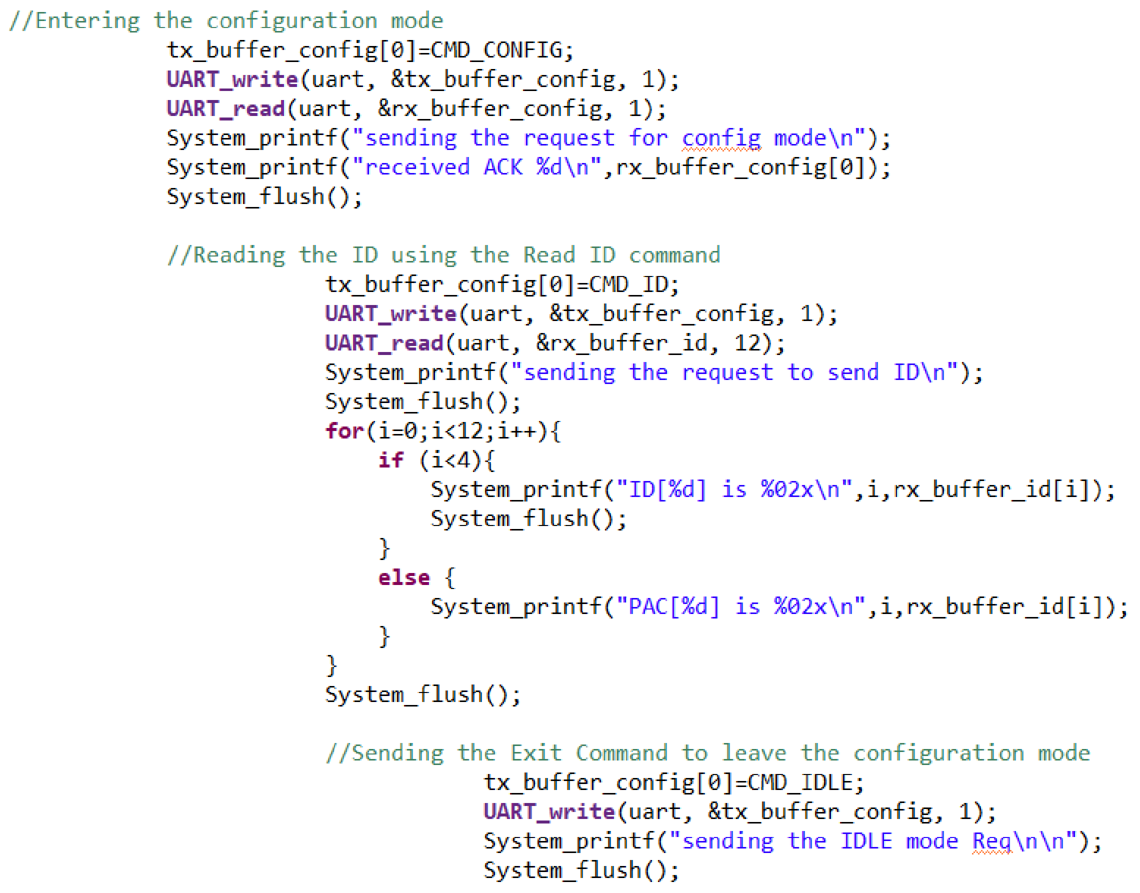
Firmware for Hardware:

In the main function, there is only one task for sigfox.



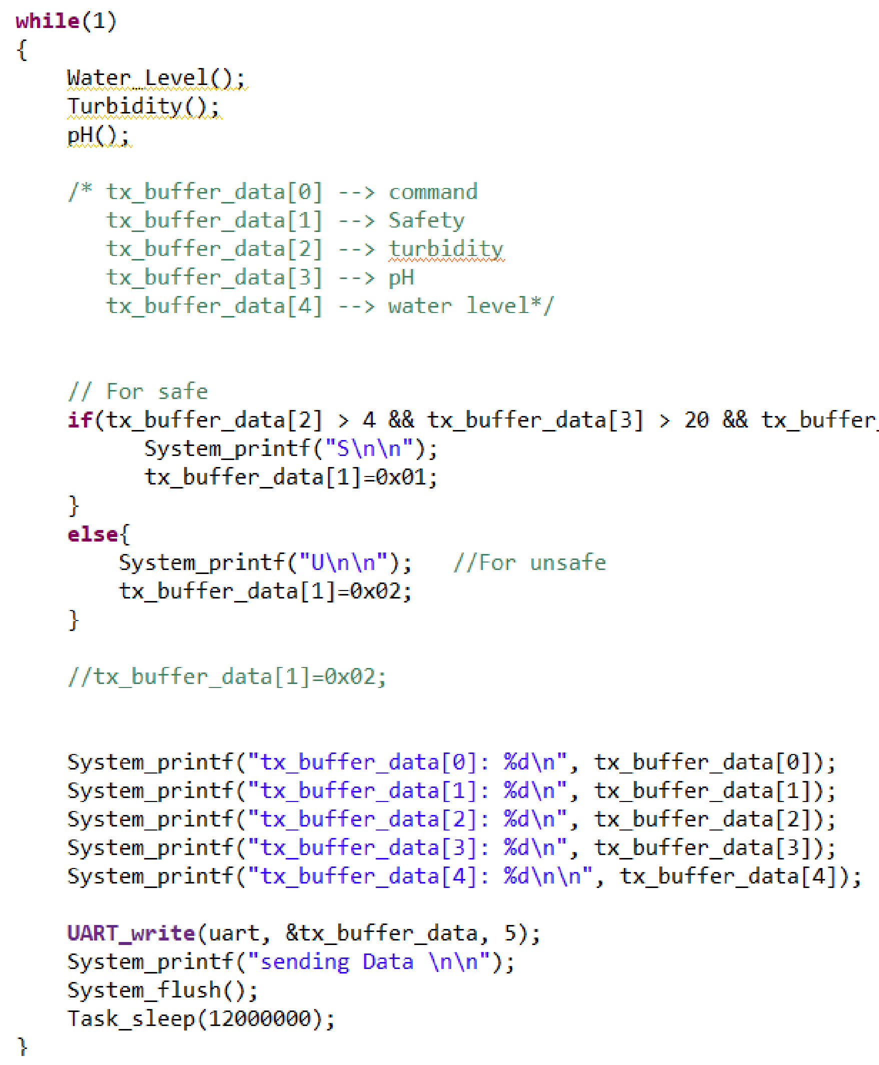
```
/* sigfox */
Task_Params_init(&taskParams);
taskParams.stackSize = TASKSTACKSIZE_sigfox;
taskParams.stack = &task3Stack;
Task_construct(&task3Struct, (Task_FuncPtr)taskFxn_sigfox, &taskParams, NULL);
```

In the sigfox task, the first step is the configuration of sigfox.



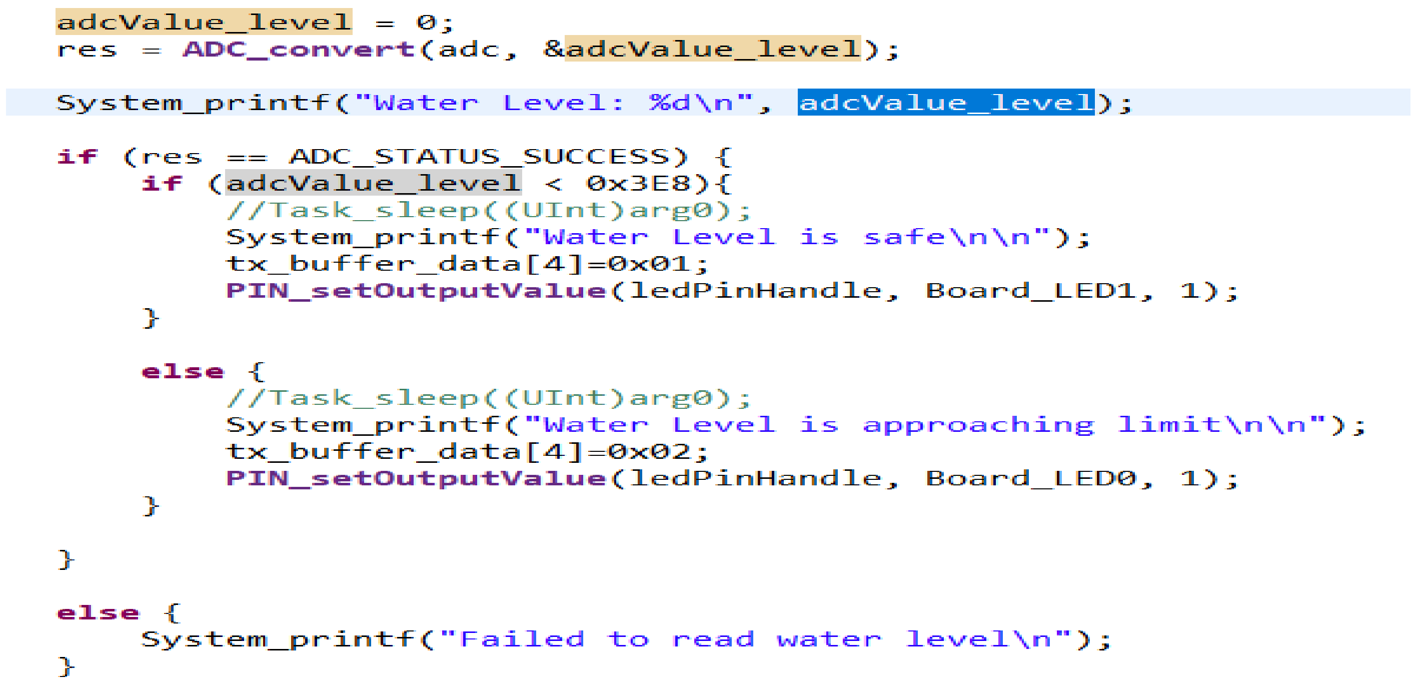
```
//Entering the configuration mode
            tx_buffer_config[0]=CMD_CONFIG;
            UART_write(uart, &tx_buffer_config, 1);
            UART_read(uart, &rx_buffer_config, 1);
            System_printf("sending the request for config mode\n");
            System_printf("received ACK %d\n",rx_buffer_config[0]);
            System_flush();

            //Reading the ID using the Read ID command
                        tx_buffer_config[0]=CMD_ID;
                        UART_write(uart, &tx_buffer_config, 1);
                        UART_read(uart, &rx_buffer_id, 12);
                        System_printf("sending the request to send ID\n");
                        System_flush();
                        for(i=0;i<12;i++){
                            if (i<4){
                                System_printf("ID[%d] is %02x\n",i,rx_buffer_id[i]);
                                System_flush();
                            }
                            else {
                                System_printf("PAC[%d] is %02x\n",i,rx_buffer_id[i]);
                            }
                        }
                        System_flush();

                        //Sending the Exit Command to leave the configuration mode
                                    tx_buffer_config[0]=CMD_IDLE;
                                    UART_write(uart, &tx_buffer_config, 1);
                                    System_printf("sending the IDLE mode Req\n\n");
                                    System_flush();
```

then “while(1)” loop is used. In this loop, a water_level function, a turbidity function and a ph function are called. According to the information collected from the turbidity function and the ph function, “0x01” meaning “water is safe” or “0x02” meaning “water is unsafe” is given to tx_buffer_data[1]. By using “UART_write”, all the data is transmitted. At the end of the loop, “task_sleep” can control the interval of every two transmissions.



```
while(1)
{
    Water_Level();
    Turbidity();
    pH();

    /* tx_buffer_data[0] --> command
       tx_buffer_data[1] --> Safety
       tx_buffer_data[2] --> turbidity
       tx_buffer_data[3] --> pH
       tx_buffer_data[4] --> water level*/


    // For safe
    if(tx_buffer_data[2] > 4 && tx_buffer_data[3] > 20 && tx_buffer_data[3] < 30){
          System_printf("S\n\n");
          tx_buffer_data[1]=0x01;
    }
    else{
        System_printf("U\n\n");   //For unsafe
        tx_buffer_data[1]=0x02;
    }

    //tx_buffer_data[1]=0x02;


    System_printf("tx_buffer_data[0]: %d\n", tx_buffer_data[0]);
    System_printf("tx_buffer_data[1]: %d\n", tx_buffer_data[1]);
    System_printf("tx_buffer_data[2]: %d\n", tx_buffer_data[2]);
    System_printf("tx_buffer_data[3]: %d\n", tx_buffer_data[3]);
    System_printf("tx_buffer_data[4]: %d\n\n", tx_buffer_data[4]);

    UART_write(uart, &tx_buffer_data, 5);
    System_printf("sending Data \n\n");
    System_flush();
    Task_sleep(12000000);
}
```

The water level sensor is used to protect the device. In Water_Level function, through "ADC_convert", "adcValue_level" receives the data from the water level sensor. If "adcValue_level" < 0x3E8, the device is safe and "0x01" is given to tx_buffer_data[4]. Else, "0x02" is given to tx_buffer_data[4].



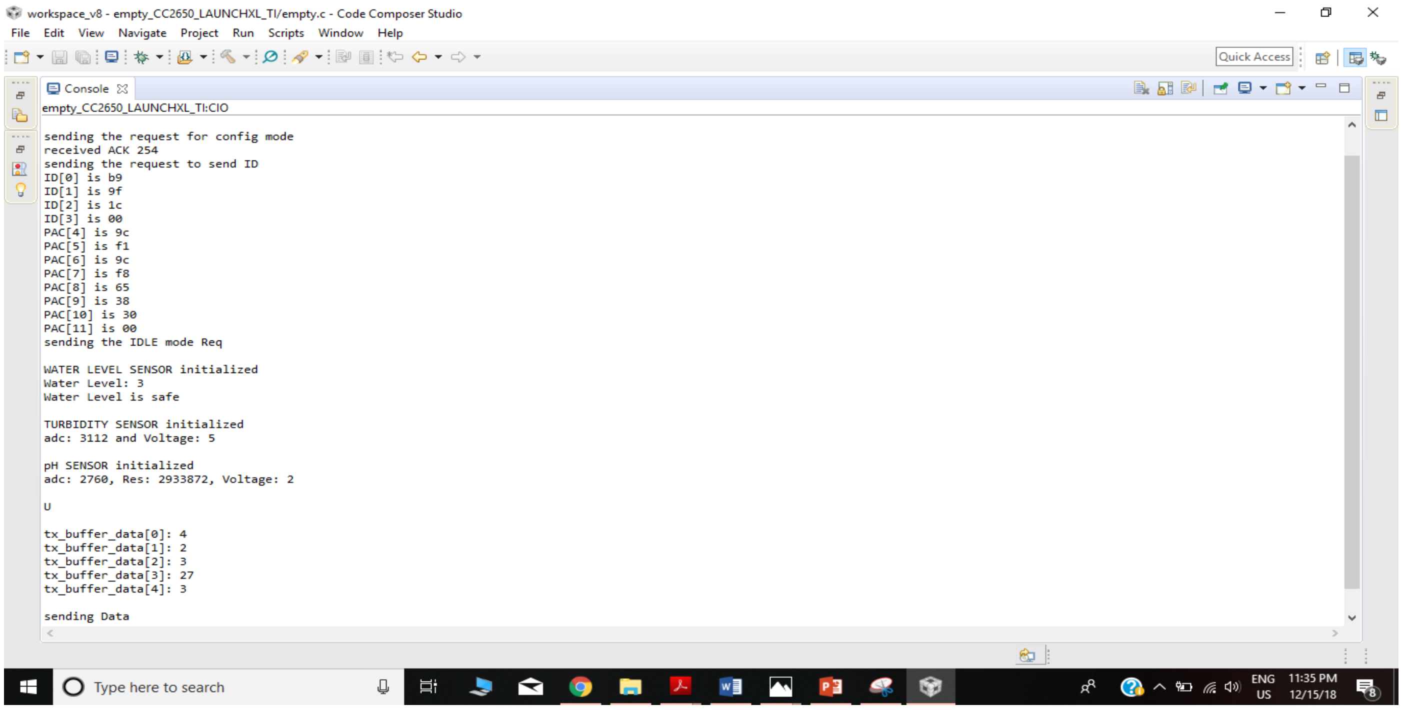
```
adcValue_level = 0;
res = ADC_convert(adc, &adcValue_level);

System_printf("Water Level: %d\n", adcValue_level);

if (res == ADC_STATUS_SUCCESS) {
    if (adcValue_level < 0x3E8){
        //Task_sleep((UInt)arg0);
        System_printf("Water Level is safe\n\n");
        tx_buffer_data[4]=0x01;
        PIN_setOutputValue(ledPinHandle, Board_LED1, 1);
    }

    else {
        //Task_sleep((UInt)arg0);
        System_printf("Water Level is approaching limit\n\n");
        tx_buffer_data[4]=0x02;
        PIN_setOutputValue(ledPinHandle, Board_LED0, 1);
    }

}

else {
    System_printf("Failed to read water level\n");
}
```

<u>In Turbidity function</u>, “adcValue_turbidity” stores the data converted by “ADC_convert”. Then send the data to tx_buffer_data[2].

```
adcValue_turbidity = 0;

res = ADC_convert(adc, &adcValue_turbidity);

adcValue_turbidity=adcValue_turbidity/1000;
tx_buffer_data[2]=adcValue_turbidity;
```

<u>In pH function</u>, “adcValue_pH” stores the data converted by “ADC_convert”. Then send the data to tx_buffer_data[3].

```
adcValue_pH = 0;

res = ADC_convert(adc, &adcValue_pH);

adcValue_pH=adcValue_pH/100;
tx_buffer_data[3]=adcValue_pH;
```


workspace_v8 - empty_CC2650_LAUNCHXL_TI/empty.c - Code Composer Studio
File Edit View Navigate Project Run Scripts Window Help
Quick Access
Console
Type here to search
ENG US
11:35 PM 12/15/18

```
empty_CC2650_LAUNCHXL_TI:CIO

sending the request for config mode
received ACK 254
sending the request to send ID
ID[0] is b9
ID[1] is 9f
ID[2] is 1c
ID[3] is 00
PAC[4] is 9c
PAC[5] is f1
PAC[6] is 9c
PAC[7] is f8
PAC[8] is 65
PAC[9] is 38
PAC[10] is 30
PAC[11] is 00
sending the IDLE mode Req

WATER LEVEL SENSOR initialized
Water Level: 3
Water Level is safe

TURBIDITY SENSOR initialized
adc: 3112 and Voltage: 5

pH SENSOR initialized
adc: 2760, Res: 2933872, Voltage: 2

U

tx_buffer_data[0]: 4
tx_buffer_data[1]: 2
tx_buffer_data[2]: 3
tx_buffer_data[3]: 27
tx_buffer_data[4]: 3

sending Data
```

The first part shows the sigfox module configuration is successful.

In the water level sensor part, if the value from the sensor is lower than a threshold (0x3E8), the device is considered safe. Because the device is not waterproof.

In the turbidity sensor part, according to the drinking water standard, if “adc” > 4000, which means the turbidity is lower than 1 NTU, the water quality is safe. If not, the water quality is unsafe.

In the ph sensor part, according to the drinking water standard, if 2000 <= “adc” <= 3000, which means the ph is from 4 to 10, the water quality is considered safe. If not, the water quality is unsafe.

According to the “adc” of turbidity and the “adc” of ph, “U” is printed as unsafe, “S” is printed as safe.

The next part is the sigfox packet structure.

“tx_buffer_data[0] = 4” means the data length is 4 bytes.

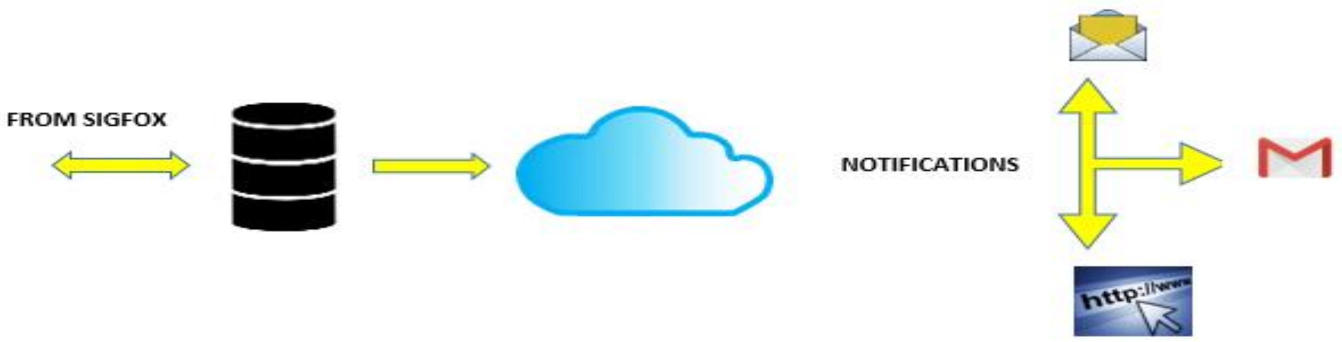
“tx_buffer_data[1] ” stores the water quality information. “1” represents “safe” and “2” represents “unsafe”.

“tx_buffer_data[2]” stores the turbidity information.

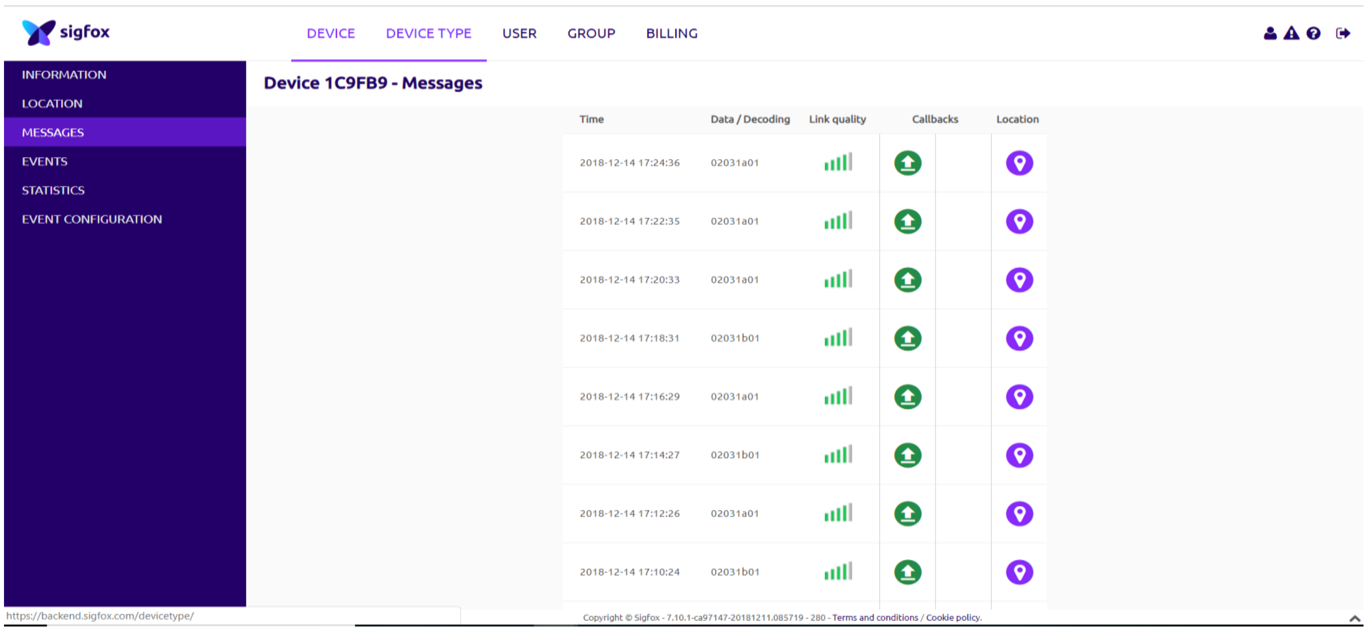
“tx_buffer_data[3]” stores the pH information.

“tx_buffer_data[4]” stores the water level information.

Sigfox Backend Callbacks:



Once the sensor readings are obtained and the quality of water has been evaluated, the data packet is constructed and transmitted the Sigfox cloud.



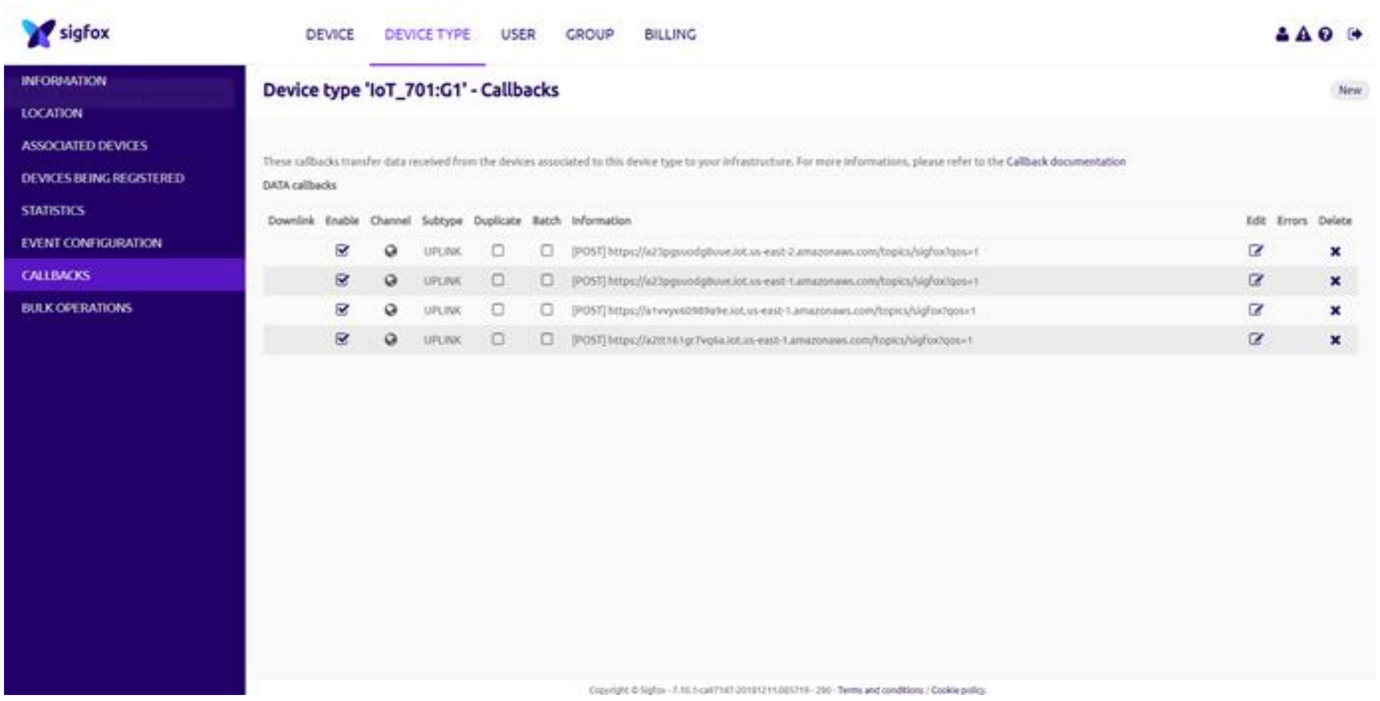
Device 1C9FB9 - Messages

Time	Data / Decoding	Link quality	Callbacks	Location
2018-12-14 17:24:36	02031a01	▂▄▆█	⬆	📍
2018-12-14 17:22:35	02031a01	▂▄▆█	⬆	📍
2018-12-14 17:20:33	02031a01	▂▄▆█	⬆	📍
2018-12-14 17:18:31	02031b01	▂▄▆█	⬆	📍
2018-12-14 17:16:29	02031a01	▂▄▆█	⬆	📍
2018-12-14 17:14:27	02031b01	▂▄▆█	⬆	📍
2018-12-14 17:12:26	02031a01	▂▄▆█	⬆	📍
2018-12-14 17:10:24	02031b01	▂▄▆█	⬆	📍

https://backend.sigfox.com/devicetype/

Copyright © Sigfox - 7.10.1-ca97147-20181211.085719 - 280 - Terms and conditions / Cookie policy.

Once the transmitted packets are received in the Sigfox cloud, a number of callback functions are executed.



DEVICE | DEVICE TYPE | USER | GROUP | BILLING

INFORMATION · LOCATION · ASSOCIATED DEVICES · DEVICES BEING REGISTERED · STATISTICS · EVENT CONFIGURATION · CALLBACKS · BULK OPERATIONS

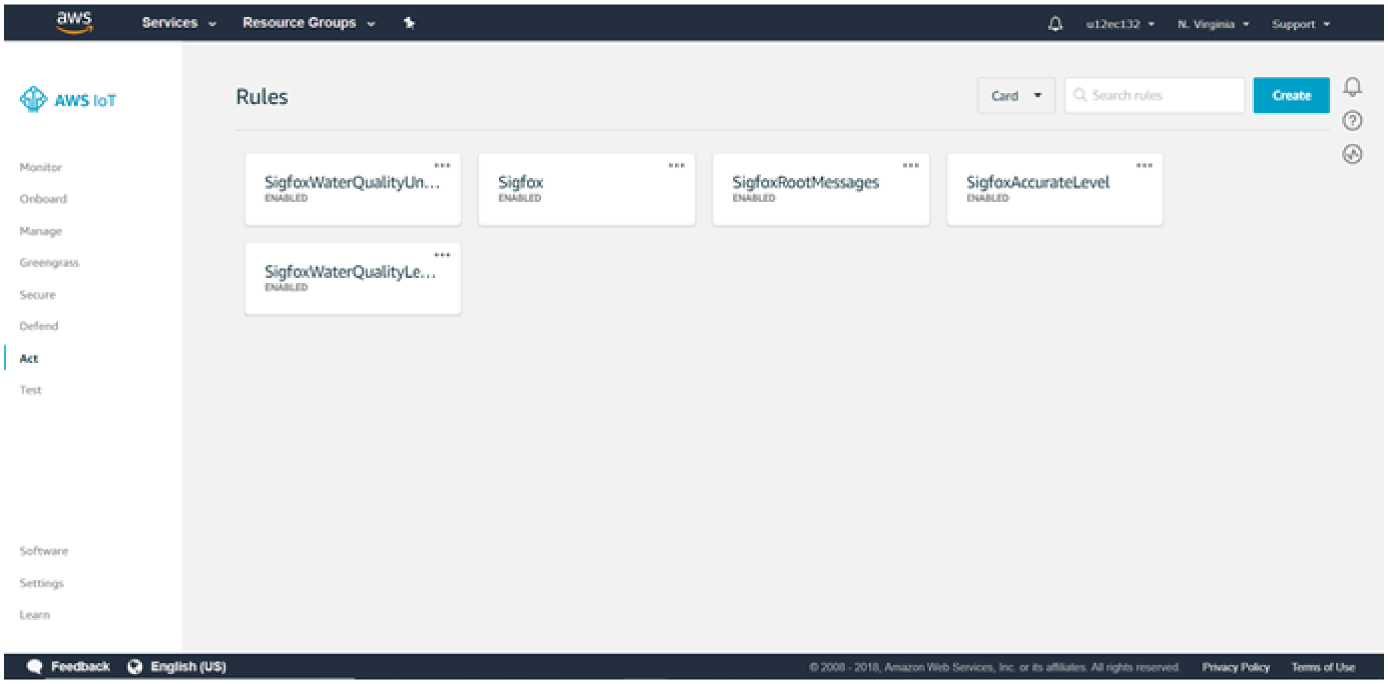
Device type 'IoT_701:G1' - Callbacks

These callbacks transfer data received from the devices associated to this device type to your infrastructure. For more informations, please refer to the Callback documentation

DATA callbacks

Downlink	Enable	Channel	Subtype	Duplicate	Batch	Information	Edit	Errors	Delete
	✓	●	UPLINK	☐	☐	[POST] https://a21pgsuodg8uve.iot.us-east-2.amazonaws.com/topics/sigfox?qos=1			✗
	✓	●	UPLINK	☐	☐	[POST] https://a21pgsuodg8uve.iot.us-east-1.amazonaws.com/topics/sigfox?qos=1			✗
	✓	●	UPLINK	☐	☐	[POST] https://a1vvyv60989a9e.iot.us-east-1.amazonaws.com/topics/sigfox?qos=1			✗
	✓	●	UPLINK	☐	☐	[POST] https://a2tt161gr7vq6a.iot.us-east-1.amazonaws.com/topics/sigfox?qos=1			✗

These callback functions are linked to different AWS Services which will be triggered based on the payload, by AWS IoT Rules defined in the AWS IoT Core.



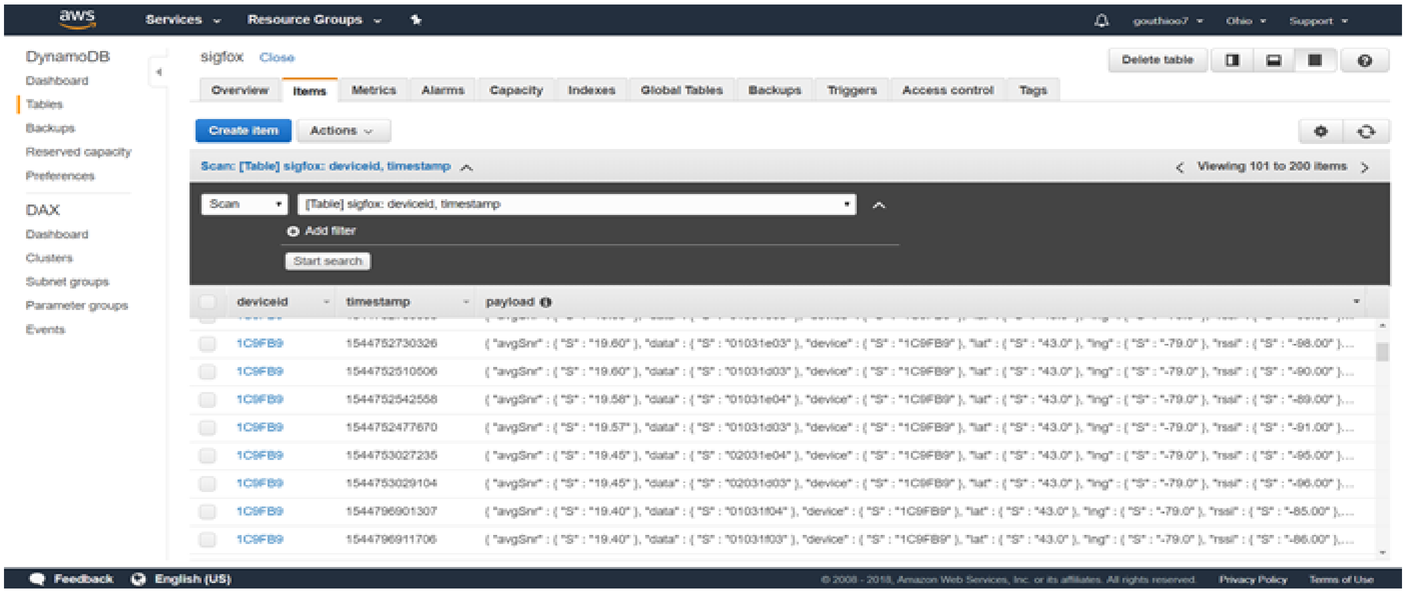
AWS IoT — Rules

SigfoxWaterQualityUn... ENABLED · Sigfox ENABLED · SigfoxRootMessages ENABLED · SigfoxAccurateLevel ENABLED · SigfoxWaterQualityLe... ENABLED

Rule Definitions:

<u>Sigfox:</u>

This rule is used to check for all incoming data from sigfox backend and pushes all the data with the entire specifications to two different DynamoDB tables using the 'deviceid' as 'Partition Key' and the 'timestamp' as the 'Sort key'. This is done to account for the extremely rare case that data fails to be pushed to one of the tables so that another table can be utilized as a fail-safe mechanism.

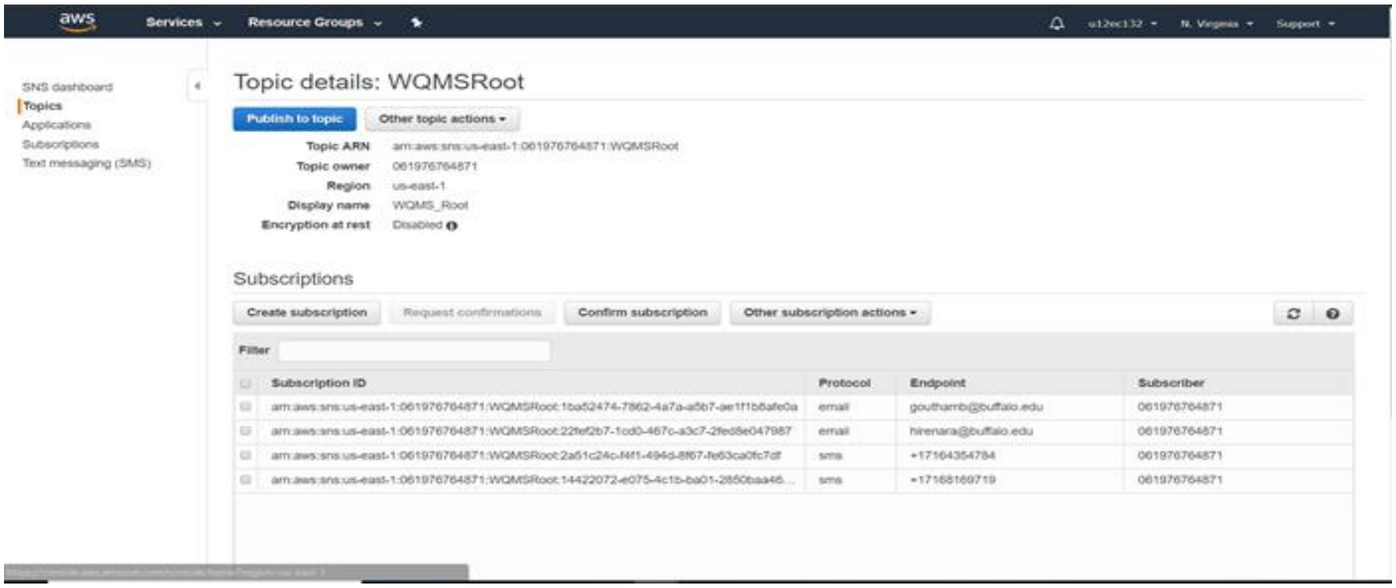


deviceid	timestamp	payload
1C9FB9	1544752730326	{ "avgSnr" : { "S" : "19.60" }, "data" : { "S" : "01031e03" }, "device" : { "S" : "1C9FB9" }, "lat" : { "S" : "43.0" }, "lng" : { "S" : "-79.0" }, "rssi" : { "S" : "-98.00" }...
1C9FB9	1544752510506	{ "avgSnr" : { "S" : "19.60" }, "data" : { "S" : "01031d03" }, "device" : { "S" : "1C9FB9" }, "lat" : { "S" : "43.0" }, "lng" : { "S" : "-79.0" }, "rssi" : { "S" : "-90.00" }...
1C9FB9	1544752542558	{ "avgSnr" : { "S" : "19.58" }, "data" : { "S" : "01031e04" }, "device" : { "S" : "1C9FB9" }, "lat" : { "S" : "43.0" }, "lng" : { "S" : "-79.0" }, "rssi" : { "S" : "-89.00" }...
1C9FB9	1544752477670	{ "avgSnr" : { "S" : "19.57" }, "data" : { "S" : "01031d03" }, "device" : { "S" : "1C9FB9" }, "lat" : { "S" : "43.0" }, "lng" : { "S" : "-79.0" }, "rssi" : { "S" : "-91.00" }...
1C9FB9	1544753027235	{ "avgSnr" : { "S" : "19.45" }, "data" : { "S" : "02031e04" }, "device" : { "S" : "1C9FB9" }, "lat" : { "S" : "43.0" }, "lng" : { "S" : "-79.0" }, "rssi" : { "S" : "-95.00" }...
1C9FB9	1544753029104	{ "avgSnr" : { "S" : "19.45" }, "data" : { "S" : "02031d03" }, "device" : { "S" : "1C9FB9" }, "lat" : { "S" : "43.0" }, "lng" : { "S" : "-79.0" }, "rssi" : { "S" : "-96.00" }...
1C9FB9	1544796901307	{ "avgSnr" : { "S" : "19.40" }, "data" : { "S" : "01031f04" }, "device" : { "S" : "1C9FB9" }, "lat" : { "S" : "43.0" }, "lng" : { "S" : "-79.0" }, "rssi" : { "S" : "-85.00" },...
1C9FB9	1544796911706	{ "avgSnr" : { "S" : "19.40" }, "data" : { "S" : "01031f03" }, "device" : { "S" : "1C9FB9" }, "lat" : { "S" : "43.0" }, "lng" : { "S" : "-79.0" }, "rssi" : { "S" : "-86.00" },...

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

<u>SigfoxRootMessages:</u>

This rule is used to trigger SMS and Email notifications for the Administrator accounts via an AWS SNS Service with the corresponding accounts (Email ID's and Phone Numbers) subscribed to the WQMSRoot topic that is linked to this SNS Service.



Topic details: WQMSRoot

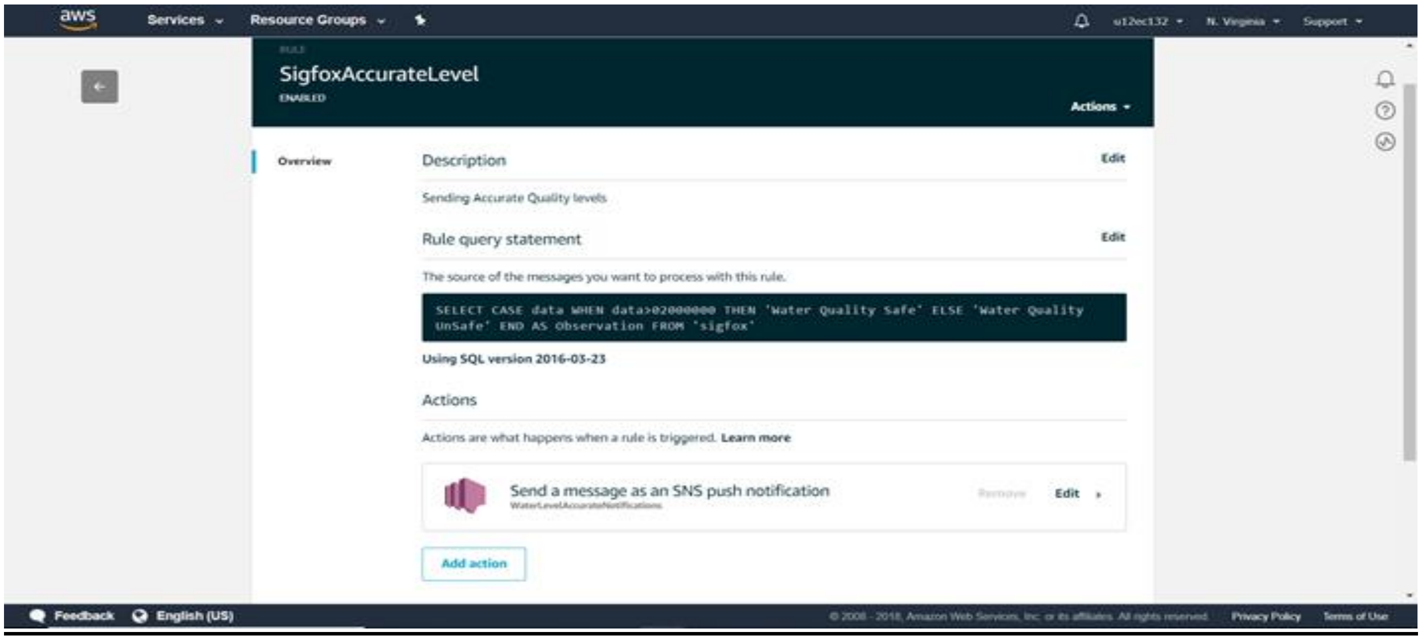
Topic ARN	arn:aws:sns:us-east-1:061976764871:WQMSRoot
Topic owner	061976764871
Region	us-east-1
Display name	WQMS_Root
Encryption at rest	Disabled

Subscriptions

Subscription ID	Protocol	Endpoint	Subscriber
arn:aws:sns:us-east-1:061976764871:WQMSRoot:1ba52474-7862-4a7a-a5b7-ae1f1b8afe0a	email	gouthamb@buffalo.edu	061976764871
arn:aws:sns:us-east-1:061976764871:WQMSRoot:22fef2b7-1cd0-467c-a3c7-2fed8e047987	email	hirenara@buffalo.edu	061976764871
arn:aws:sns:us-east-1:061976764871:WQMSRoot:2a51c24c-f4f1-494d-8f67-fe63ca0fc7df	sms	+17164354784	061976764871
arn:aws:sns:us-east-1:061976764871:WQMSRoot:14422072-e075-4c1b-ba01-2850baa46...	sms	+17168169719	061976764871

<u>SigfoxAccurateLevel:</u>

This IoT Rule implements a MySQL query that checks the Water-Quality Indicating byte in the received payload and based on this, a AWS SNS service is triggered which will notify the Primary Administrator accounts(Email and SMS notifications).



SigfoxAccurateLevel

ENABLED

Description

Sending Accurate Quality levels

Rule query statement

The source of the messages you want to process with this rule.

```
SELECT CASE data WHEN data>02000000 THEN 'Water Quality Safe' ELSE 'Water Quality
UnSafe' END AS Observation FROM 'sigfox'
```

Using SQL version 2016-03-23

Actions

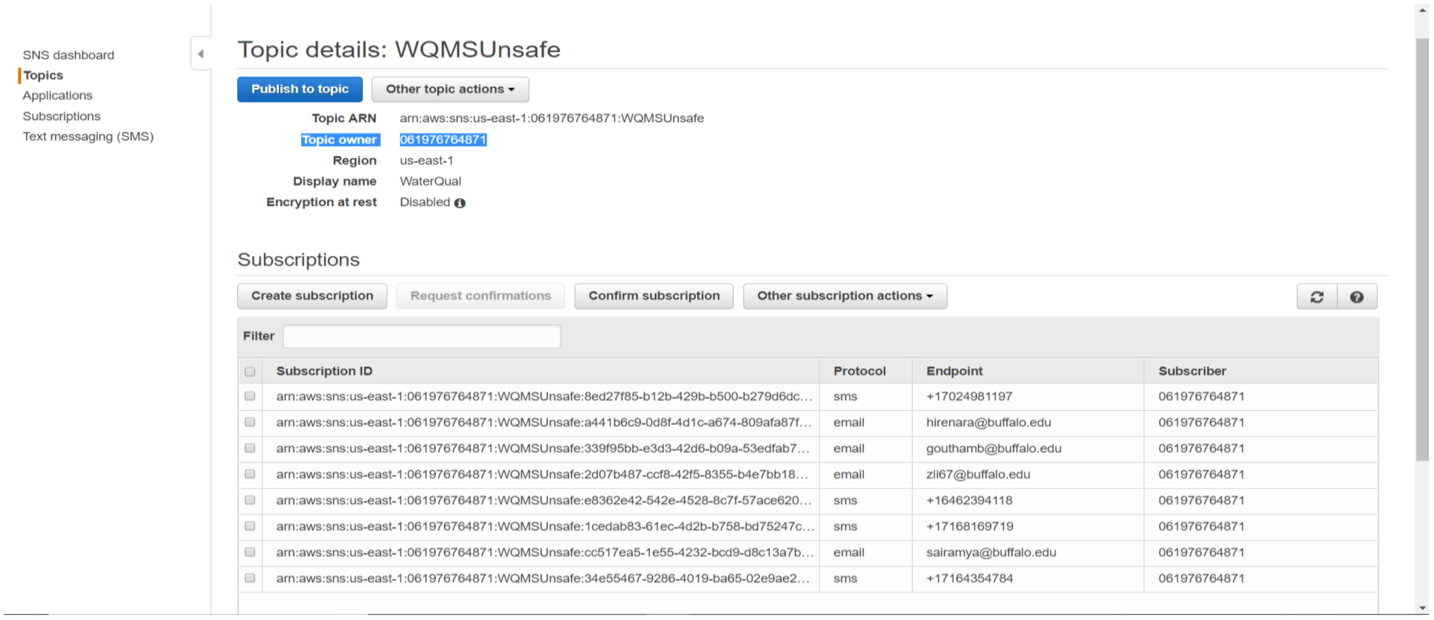
Actions are what happens when a rule is triggered. Learn more

Send a message as an SNS push notification
WaterLevelAccurateNotifications

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

SigfoxWaterQualityLevel:

This rule also checks the Water-Quality Indicating byte in the received payload and if the quality is determined to be unsafe, an AWS SNS Service is triggered which sends SMS and Email notifications to everyone subscribed for the service as well as administrators so necessary actions can be taken as early as possible.



Topic details: WQMSUnsafe

Topic ARN	arn:aws:sns:us-east-1:061976764871:WQMSUnsafe
Topic owner	061976764871
Region	us-east-1
Display name	WaterQual
Encryption at rest	Disabled

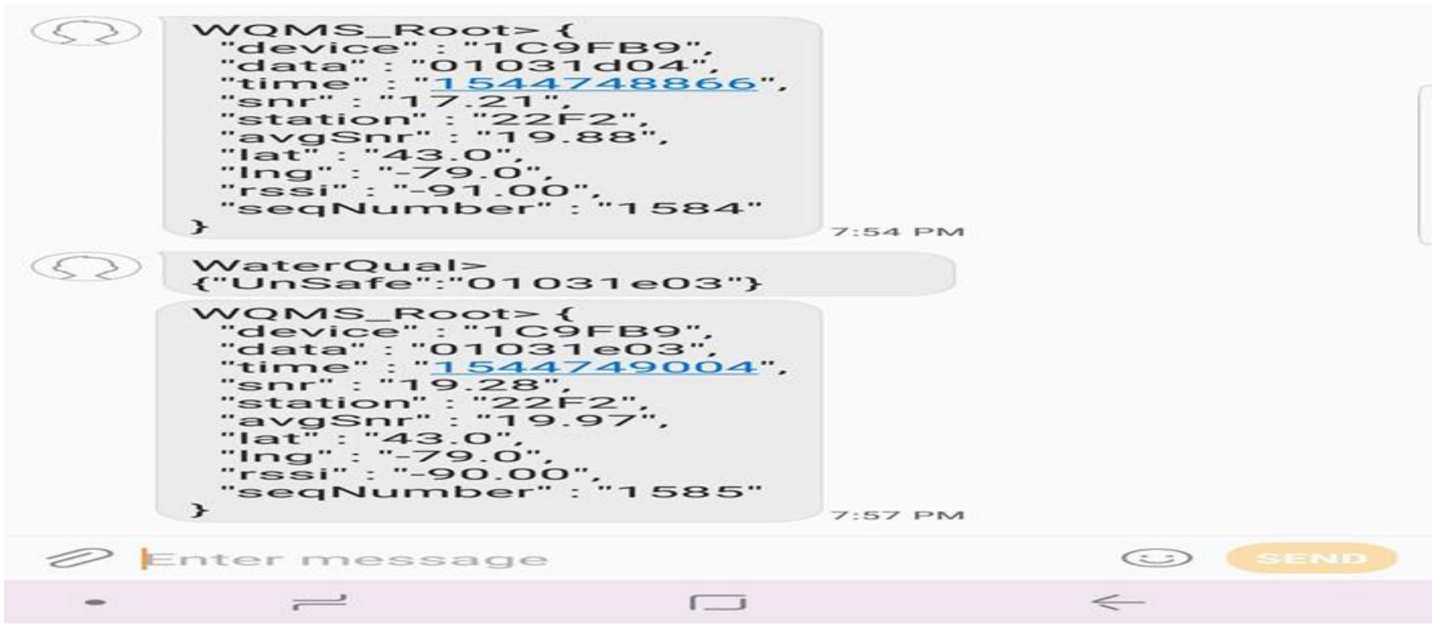
Subscriptions

Subscription ID	Protocol	Endpoint	Subscriber
arn:aws:sns:us-east-1:061976764871:WQMSUnsafe:8ed27f85-b12b-429b-b500-b279d6dc...	sms	+17024981197	061976764871
arn:aws:sns:us-east-1:061976764871:WQMSUnsafe:a441b6c9-0d8f-4d1c-a674-809afa87f...	email	hirenara@buffalo.edu	061976764871
arn:aws:sns:us-east-1:061976764871:WQMSUnsafe:339f95bb-e3d3-42d6-b09a-53edfab7...	email	gouthamb@buffalo.edu	061976764871
arn:aws:sns:us-east-1:061976764871:WQMSUnsafe:2d07b487-ccf8-42f5-8355-b4e7bb18...	email	zli67@buffalo.edu	061976764871
arn:aws:sns:us-east-1:061976764871:WQMSUnsafe:e8362e42-542e-4528-8c7f-57ace620...	sms	+16462394118	061976764871
arn:aws:sns:us-east-1:061976764871:WQMSUnsafe:1cedab83-61ec-4d2b-b758-bd75247c...	sms	+17168169719	061976764871
arn:aws:sns:us-east-1:061976764871:WQMSUnsafe:cc517ea5-1e55-4232-bcd9-d8c13a7b...	email	sairamya@buffalo.edu	061976764871
arn:aws:sns:us-east-1:061976764871:WQMSUnsafe:34e55467-9286-4019-ba65-02e9ae2...	sms	+17164354784	061976764871

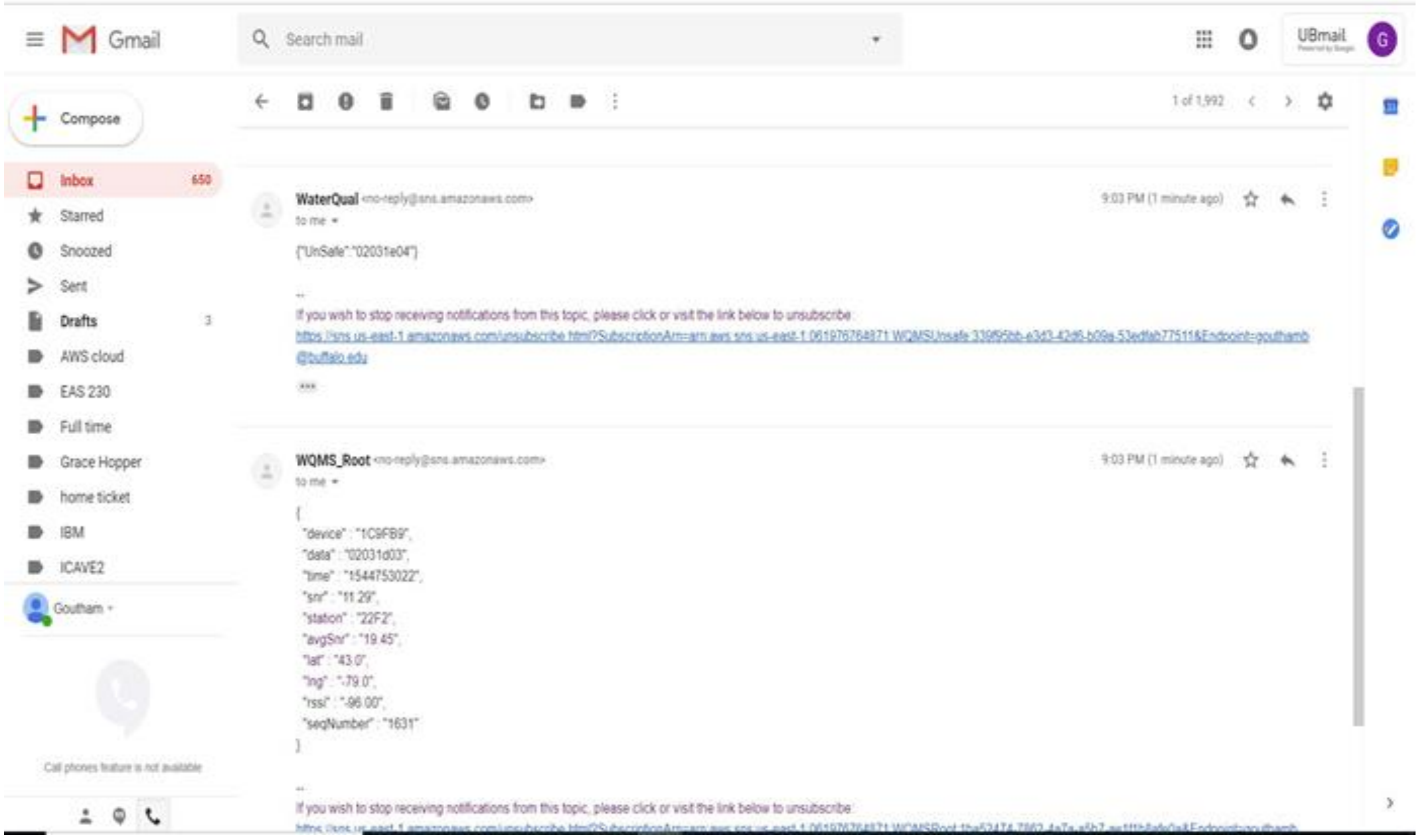
User-Interface:

As mentioned above, there are three User-Interfaces available, two of which are Internet Network connection based and one is Cellular Network connection based, to ensure that in the cases of failure of either of the systems, there is a backup mechanism that is implemented to deliver the notifications.

The following images are the notifications that are received by the Administrator Account via Email and SMS. The entire payload is displayed and also the Warning message indicating the data of the sensors is also sent, as two separate notifications, through Email and SMS.



```
WQMS_Root> {
  "device" : "1C9FB9",
  "data" : "01031d04",
  "time" : "1544748866",
  "snr" : "17.21",
  "station" : "22F2",
  "avgSnr" : "19.88",
  "lat" : "43.0",
  "lng" : "-79.0",
  "rssi" : "-91.00",
  "seqNumber" : "1584"
}
7:54 PM

WaterQual>
{"UnSafe":"01031e03"}

WQMS_Root> {
  "device" : "1C9FB9",
  "data" : "01031e03",
  "time" : "1544749004",
  "snr" : "19.28",
  "station" : "22F2",
  "avgSnr" : "19.97",
  "lat" : "43.0",
  "lng" : "-79.0",
  "rssi" : "-90.00",
  "seqNumber" : "1585"
}
7:57 PM
```



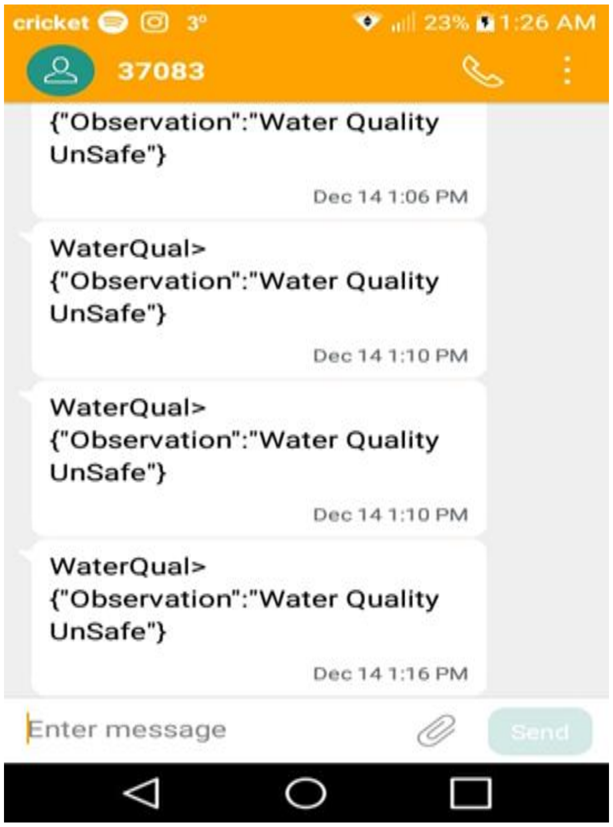
WaterQual <no-reply@sns.amazonaws.com>
to me
9:03 PM (1 minute ago)

```
{"UnSafe":"02031e04"}

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:061976764871:WQMSUnsafe:339f95bb-e3d3-42d6-b09a-53edfab77511&Endpoint=goutham
@buffalo.edu
```

WQMS_Root <no-reply@sns.amazonaws.com>
to me
9:03 PM (1 minute ago)

```
{
 "device" : "1C9FB9",
 "data" : "02031d03",
 "time" : "1544753022",
 "snr" : "11.29",
 "station" : "22F2",
 "avgSnr" : "19.45",
 "lat" : "43.0",
 "lng" : "-79.0",
 "rssi" : "-96.00",
 "seqNumber" : "1631"
}

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:061976764871:WQMSRoot:1ba52474-7862-4a7a-a5b7-ae1f1b8a6e0a&Endpoint=goutham
```

The following images represent the notifications received by accounts that are subscribed for the service.


cricket 3° 23% 1:26 AM

37083

{"Observation":"Water Quality UnSafe"}

Dec 14 1:06 PM

WaterQual>
{"Observation":"Water Quality UnSafe"}

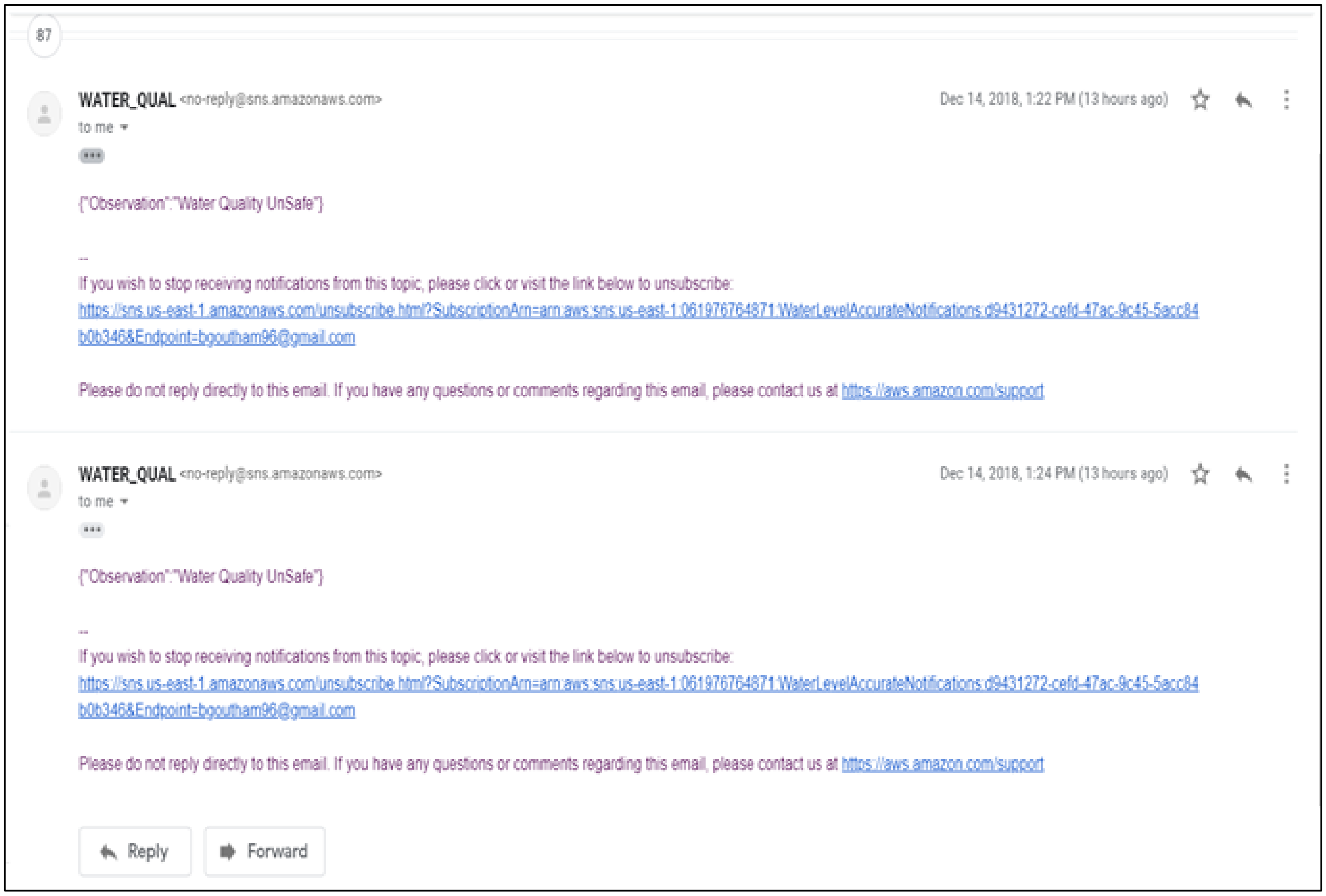
Dec 14 1:10 PM

WaterQual>
{"Observation":"Water Quality UnSafe"}

Dec 14 1:10 PM

WaterQual>
{"Observation":"Water Quality UnSafe"}

Dec 14 1:16 PM



WATER_QUAL <no-reply@sns.amazonaws.com>
to me

Dec 14, 2018, 1:22 PM (13 hours ago)

{"Observation":"Water Quality UnSafe"}

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:061976764871:WaterLevelAccurateNotifications:d9431272-cefd-47ac-9c45-5acc84b0b346&Endpoint=bgoutham96@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

WATER_QUAL <no-reply@sns.amazonaws.com>
to me

Dec 14, 2018, 1:24 PM (13 hours ago)

{"Observation":"Water Quality UnSafe"}

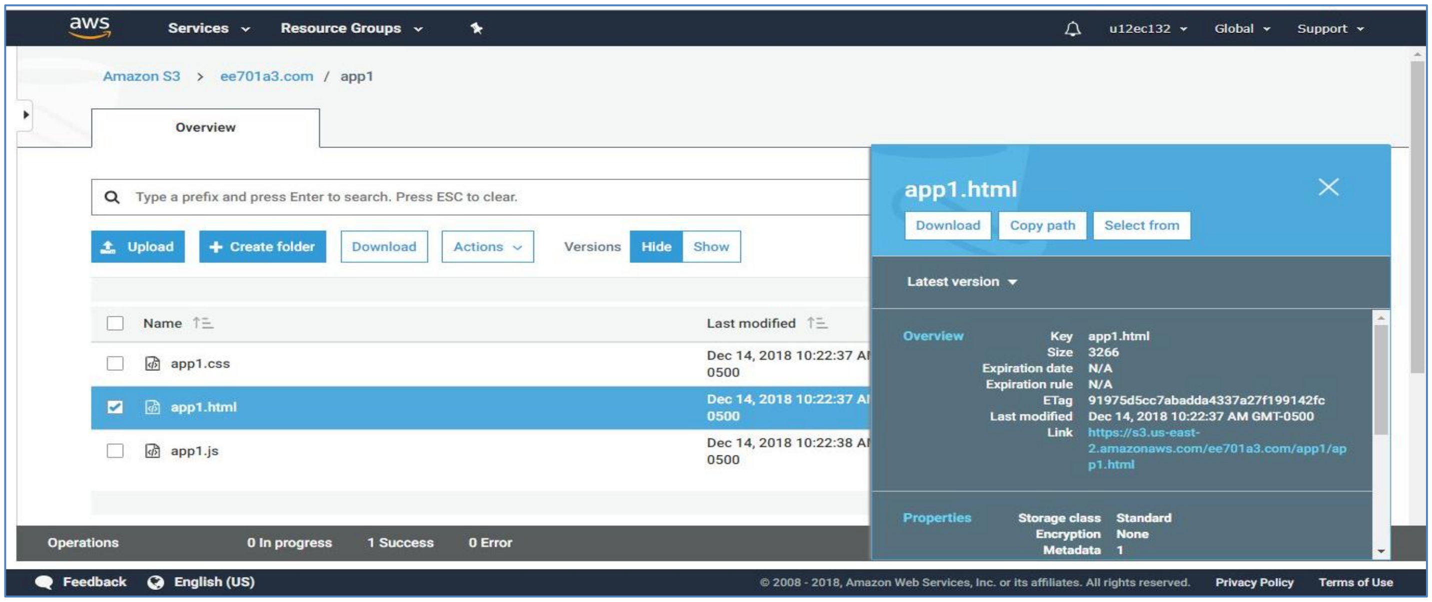
--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:061976764871:WaterLevelAccurateNotifications:d9431272-cefd-47ac-9c45-5acc84b0b346&Endpoint=bgoutham96@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

In addition to this, there is also a Webpage that has been hosted on a AWS S3 Service allowing the entire system to be highly scalable, secure and enhances data availability and performance.

FRONTEND (CLIENT-SIDE):

The frontend is designed to access details about the condition of water from multiple locations which has deployed this system from the client side. This has been designed using HTML, CSS and Javascript. The code is used to access data from Amazon DynamoDB and present it in the form of an interactive webpage. The webpage is hosted on Amazon S3.



Amazon S3 > ee701a3.com / app1

Type a prefix and press Enter to search. Press ESC to clear.

Name	Last modified
app1.css	Dec 14, 2018 10:22:37 AM 0500
app1.html	Dec 14, 2018 10:22:37 AM 0500
app1.js	Dec 14, 2018 10:22:38 AM 0500

Operations 0 In progress 1 Success 0 Error

app1.html

Overview

Key	app1.html
Size	3266
Expiration date	N/A
Expiration rule	N/A
ETag	91975d5cc7abadda4337a27f199142fc
Last modified	Dec 14, 2018 10:22:37 AM GMT-0500
Link	https://s3.us-east-2.amazonaws.com/ee701a3.com/app1/app1.html

Properties

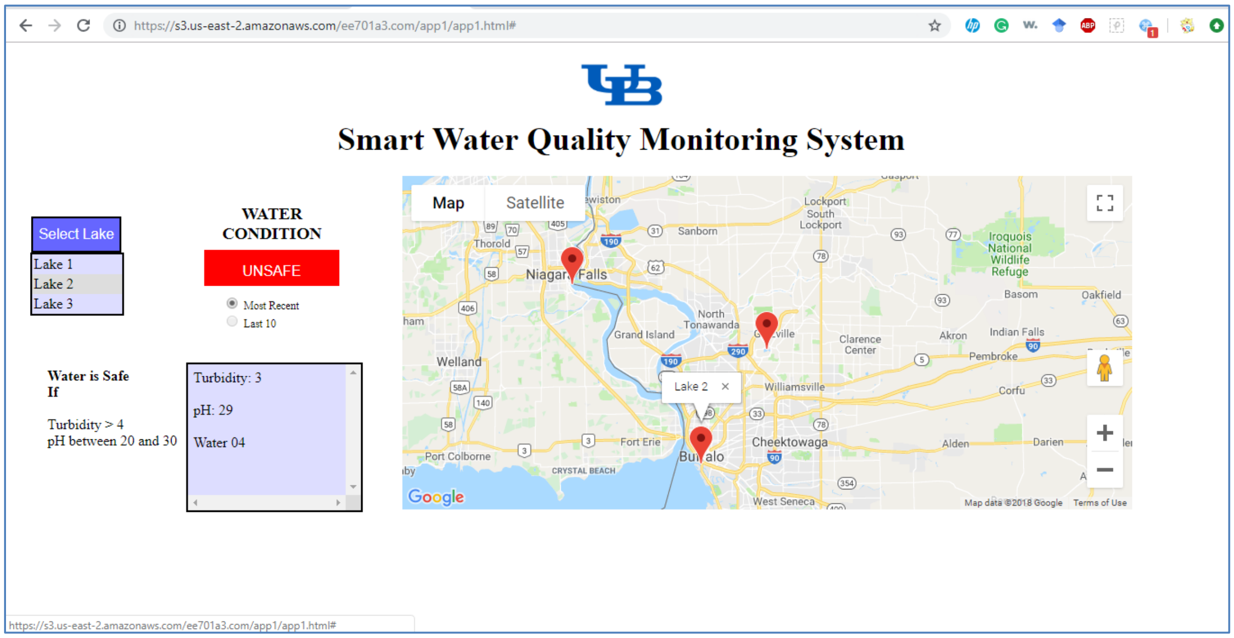
Storage class	Standard
Encryption	None
Metadata	1

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon S3(Simple Storage Service) is storage for the Internet. It is designed to make web-scale computing easier for developers. Amazon S3 has a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web.

First a bucket on S3 called ee701a3.com was created in which the three files of HTML, CSS and JavaScript have been added. A bucket is a container for objects stored in Amazon S3. Every object is contained in a bucket. Through the link associated with the HTML file, the webpage can be accessed.

From the client side, when this page is opened, there is an integrated map, which shows the locations where the module has been deployed in the form of markers. Clicking on the marker or selecting the waterbody from the list gives the details of the waterbody in terms of whether it is safe or not, it's turbidity, pH and water level measurements.



https://s3.us-east-2.amazonaws.com/ee701a3.com/app1/app1.html#

Smart Water Quality Monitoring System

Select Lake
Lake 1
Lake 2
Lake 3

WATER CONDITION

UNSAFE

Most Recent
Last 10

Water is Safe If

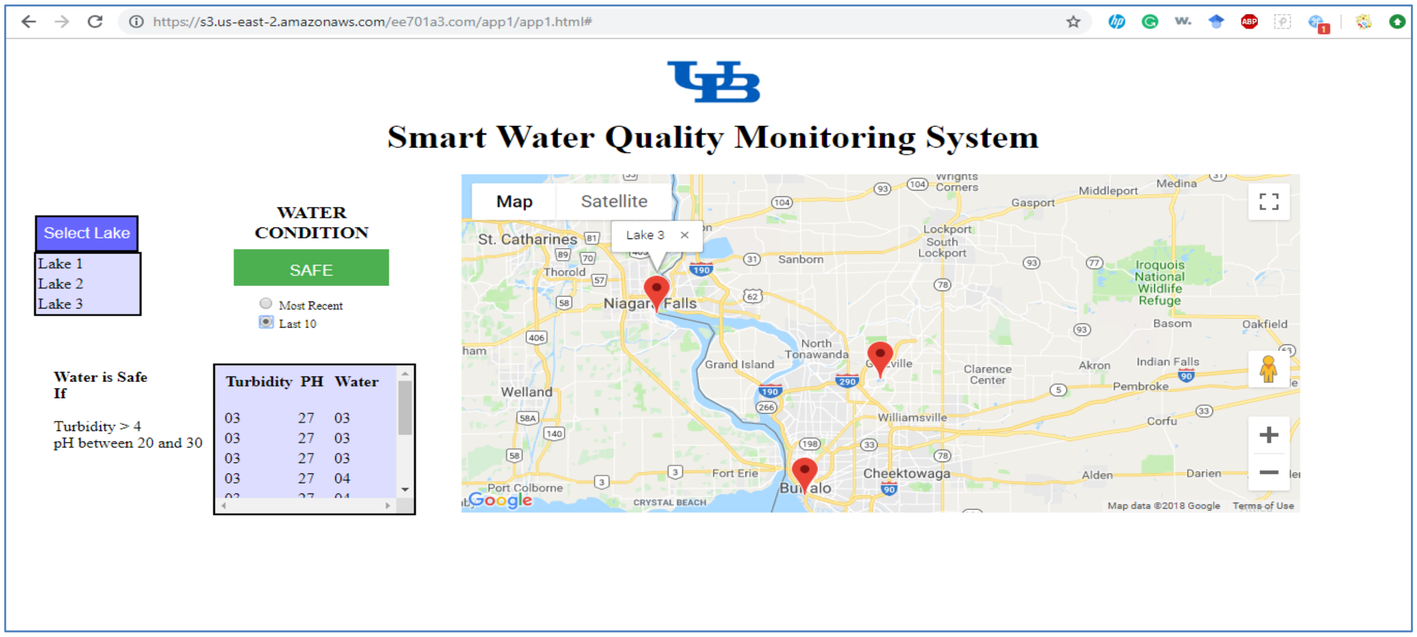
Turbidity > 4
pH between 20 and 30

Turbidity: 3

pH: 29

Water 04

If the quality of water is unsafe, there is a red button which says it is unsafe, likewise a green button when it is safe. Also, the last 10 readings can be accessed when the user clicks on that option.



https://s3.us-east-2.amazonaws.com/ee701a3.com/app1/app1.html#

Smart Water Quality Monitoring System

Select Lake

Lake 1
Lake 2
Lake 3

WATER CONDITION

SAFE

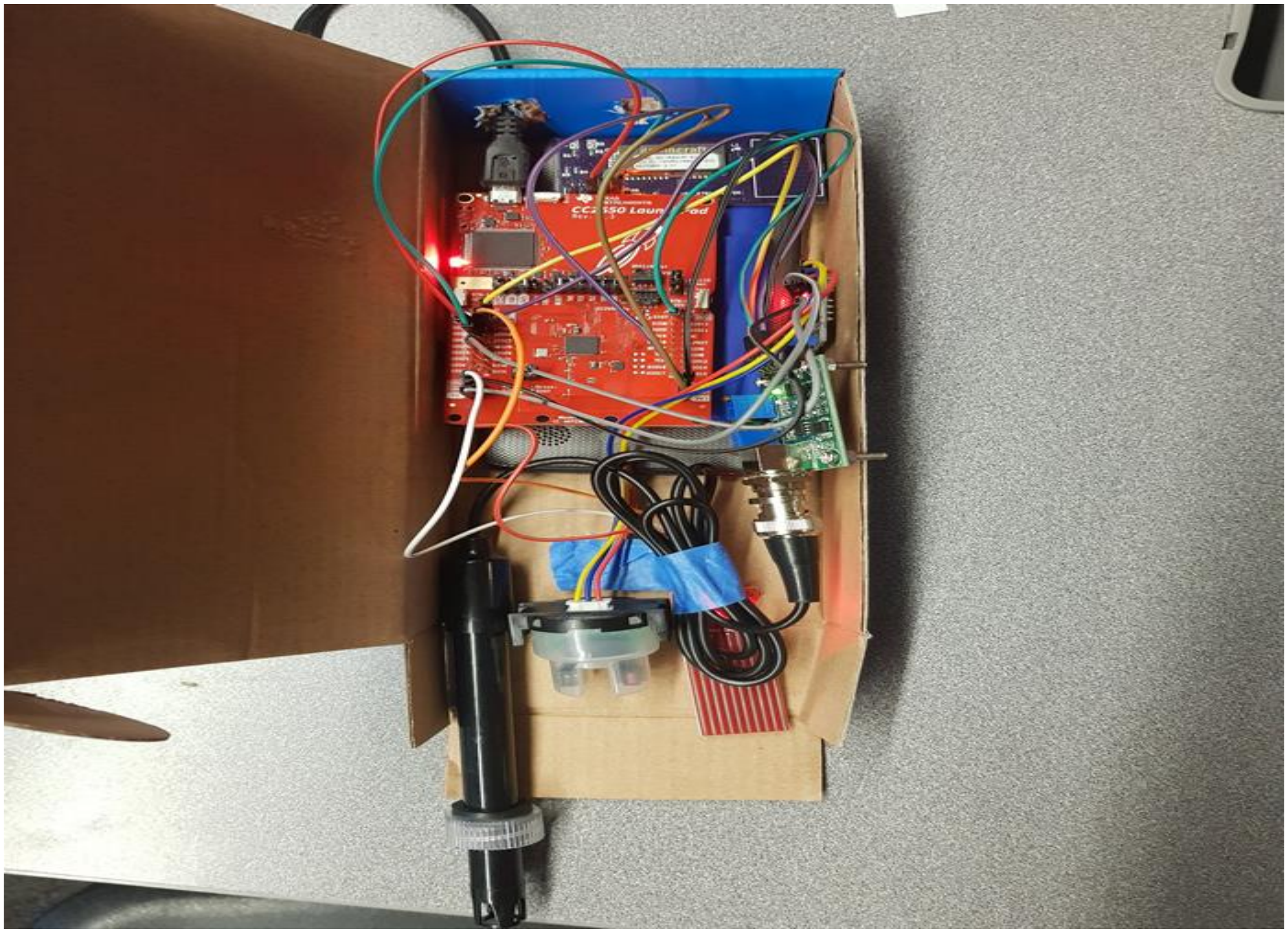
( ) Most Recent
(●) Last 10

Water is Safe If

Turbidity > 4
pH between 20 and 30

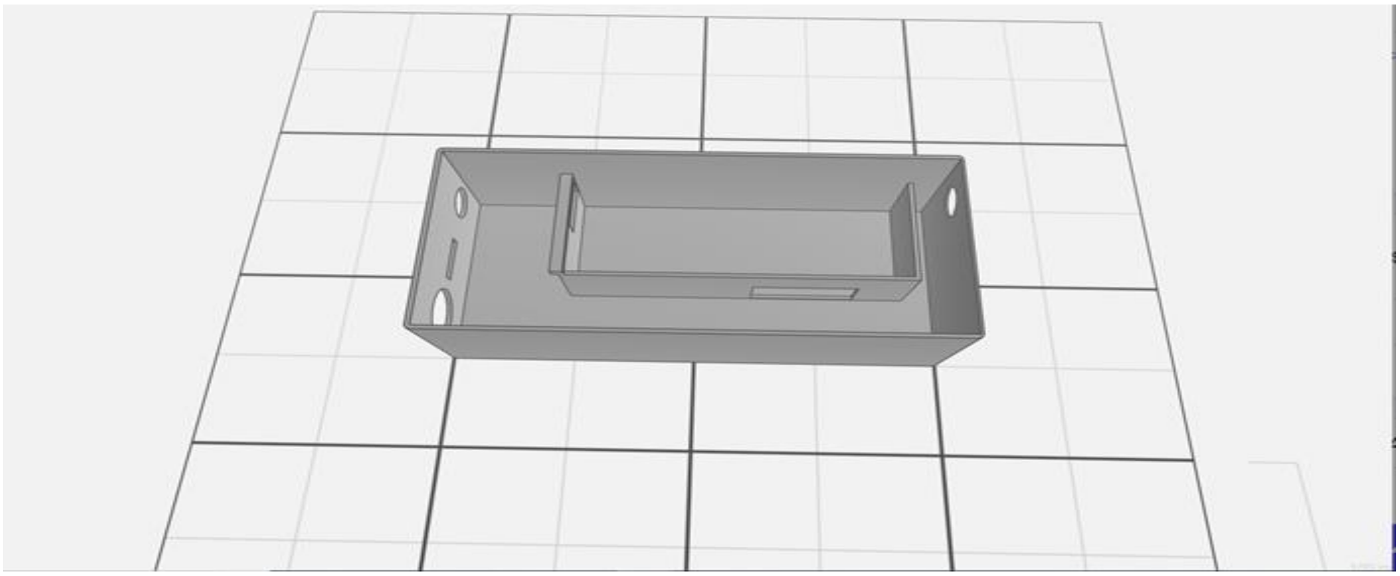
Turbidity	PH	Water
03	27	03
03	27	03
03	27	03
03	27	04

<u>3D Prototype Design:</u>

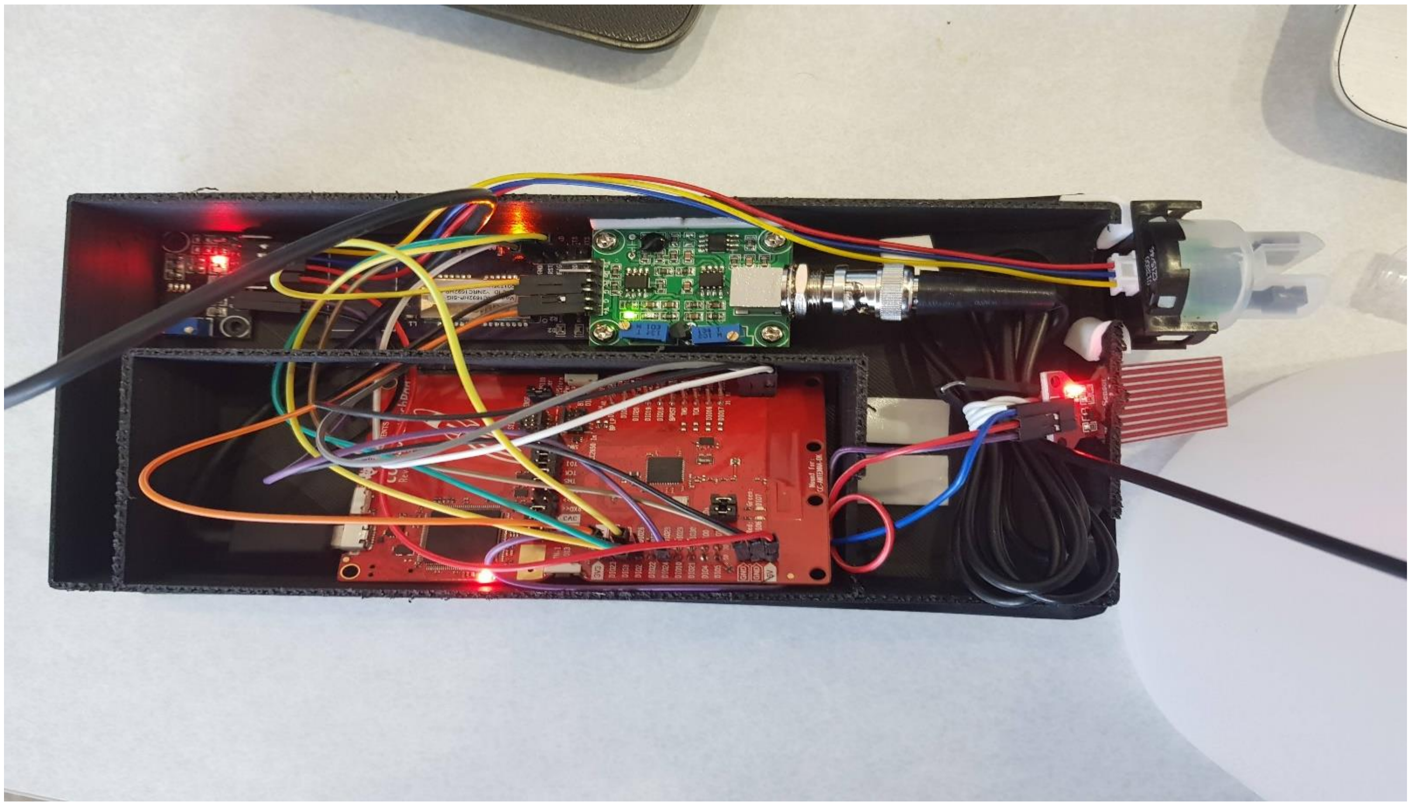


The preliminary prototype was designed using a cardboard box as the design base inside which all the sensors, modules and the LaunchPad were arranged. The appropriate connections along with a power source (Power Bank) were given and all the components were attached firmly to the base and the experiment was run to ensure the proper functioning of the device as a whole as well as the individual components.

Once the preliminary prototype was tested, the actual prototype was designed by accounting for each component and their place in the prototype. Necessary compartments were designed to isolate the power-source and the LaunchPad from rest of the components to provide additional protection from water. Additionally, separate compartment was designed to place the sigfox modules and the other interfacing modules for the sensors.



In addition to this, a cover resembling two rectangular prisms mounted on each other with different dimensions was also designed to provide an air-tight seal to the entire device once the components are fixed to the base with corresponding connections and interfaces.



CONCLUSION AND FUTURE WORK:

We have successfully designed and implemented an economical, scalable and efficient Internet of Things solution to track the water quality of large water bodies which can enable users to access the water quality information from multiple locations in real-time at the convenience of their mobile phone.

The future scope of this project includes:

- Developing specific algorithms to define the safety in terms of various defined activities. (e.g. Cattle Consumption, Human Consumption, Watersports, etc.) and integrating different types of sensors to characterize water-quality.
- Generating a compact PCB design for fabrication which will reduce the production cost of each device by up to 50%
- Increasing the number of devices to compare the variation of water quality from environmental perspective across different regions.

INSTRUCTION GUIDE FOR DEMONSTRATION:

1) If the module is intact with all the sensors connected, connect the power supply to the device. This can be done by a power bank or plugging it to the laptop using micro-USB cable.

2) The model has to be immersed in the test water carefully, making sure that water does not enter near the circuit and only the sensors are present in the water.

3) Leave it in this position for a minimum of 2-3 minutes for the sensor readings to be accurate.

4) The pH sensor has to be recalibrated every time there is significant change in the pH value to be measured.

5) The webpage for the demo can be accessed

https://s3.us-east-2.amazonaws.com/ee701a3.com/app1/app1.html