IOT PROJECT REPORT

IOT BASED ASSET TRACKING SYSTEM

SARIT BURMAN
ROLL 180108039
Indian Institute of Technology , Guwahati

MAIN OBJECTIVE

To implement a IoT based Asset Tracking System that will allow to track the location of objects, goods, personnel within a building or facility.

IMPLEMENTED ATTRIBUTES

1) RF transmitters and receivers are used to communicate between the assets and the receiver in the rooms.

2) Tracker Circuit is small enough to be mounted on the entity and is battery powered.

3) Monitoring Circuits(RF receiver module connected to Arduino) placed in rooms, so that when an object enters the room or comes in 2-3 meter range of monitoring circuit, the monitoring system should transmit the AssetID of that transmitter circuit associated with that particular asset the to online system(via Wi-Fi module) where the Administrator will keep track of all Assets. (Online IoT System)

4) An Email Notification is sent by the Online IoT System to User's Mobile Device who wants to know the location of the Entity and the AssetID and the time of its arrival.

CONFIGURATION DIAGRAM

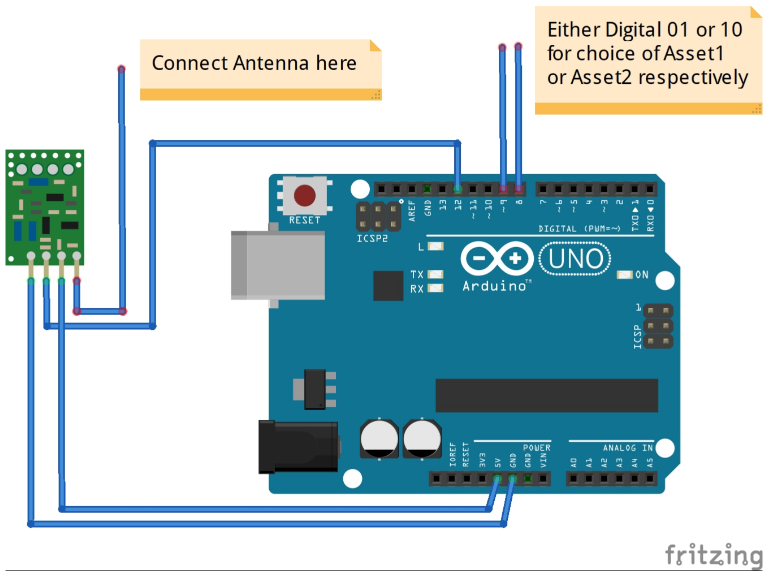

Connect Antenna here
Either Digital 01 or 10 for choice of Asset1 or Asset2 respectively
fritzing

Figure 1: Diagram of the Transmitter circuit(Asset)

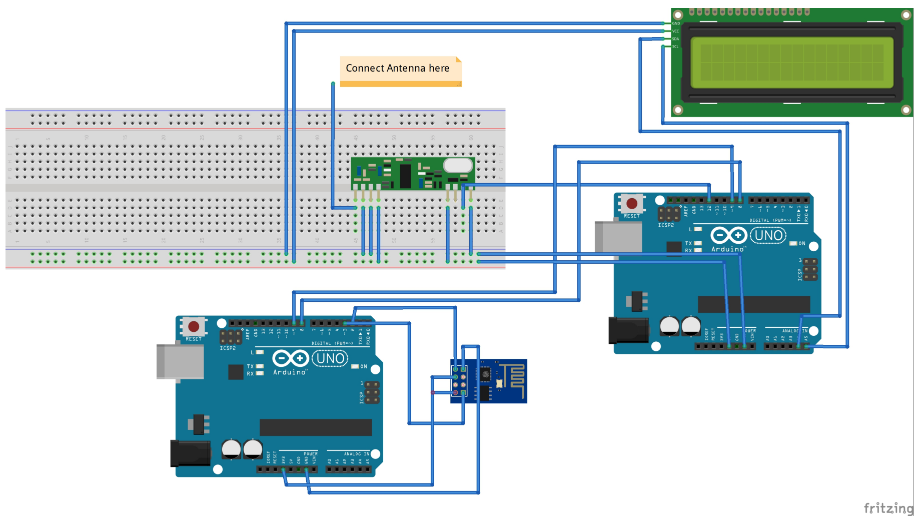


Figure 2: Diagram of the Receiver circuit

SAMPLE OUTPUTS

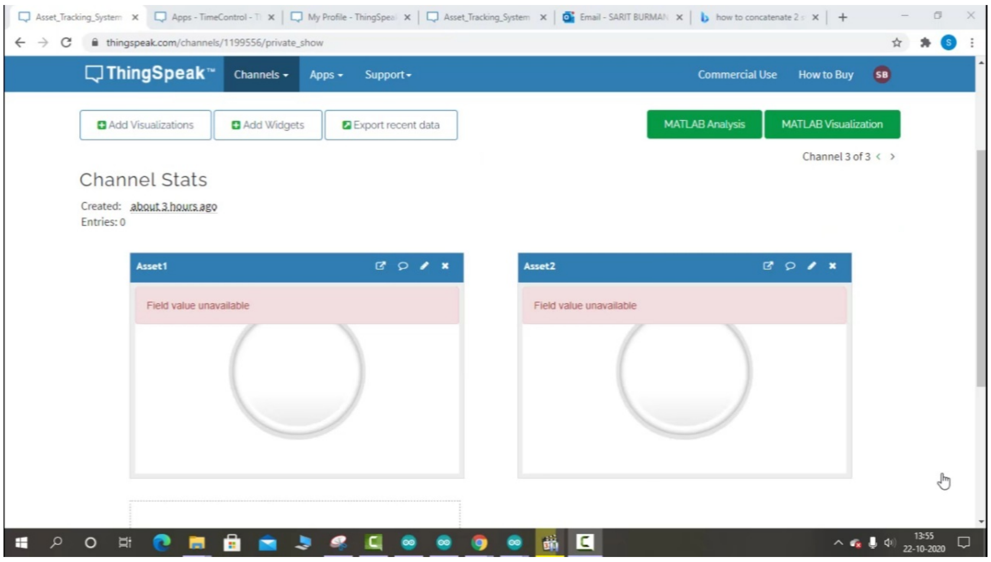
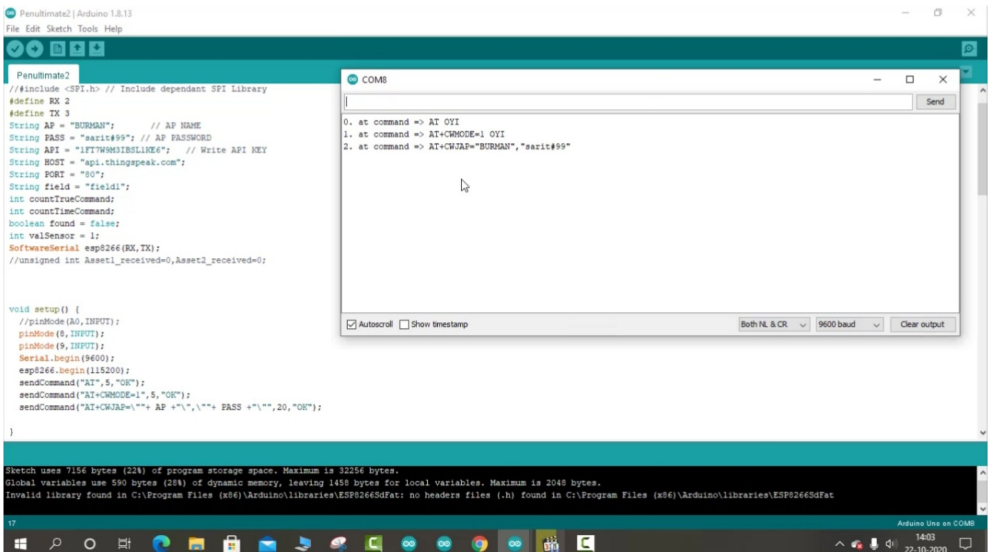


Figure 3: Initially the cloud shows no asset has arrived



```
//#include <SPI.h> // Include dependant SPI Library
#define RX 2
#define TX 3
String AP = "BURMAN";       // AP NAME
String PASS = "sarit#99"; // AP PASSWORD
String API = "1FT7W9M3IBSL1KE6";   // Write API KEY
String HOST = "api.thingspeak.com";
String PORT = "80";
String field = "field1";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
int valSensor = 1;
SoftwareSerial esp8266(RX,TX);
//unsigned int Asset1_received=0,Asset2_received=0;



void setup() {
  //pinMode(A0,INPUT);
  pinMode(8,INPUT);
  pinMode(9,INPUT);
  Serial.begin(9600);
  esp8266.begin(115200);
  sendCommand("AT",5,"OK");
  sendCommand("AT+CWMODE=1",5,"OK");
  sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");

}
```

```
0. at command => AT OYI
1. at command => AT+CWMODE=1 OYI
2. at command => AT+CWJAP="BURMAN","sarit#99"
```

```
Sketch uses 7156 bytes (22%) of program storage space. Maximum is 32256 bytes.
Global variables use 590 bytes (28%) of dynamic memory, leaving 1458 bytes for local variables. Maximum is 2048 bytes.
Invalid library found in C:\Program Files (x86)\Arduino\libraries\ESP8266SdFat: no headers files (.h) found in C:\Program Files (x86)\Arduino\libraries\ESP8266SdFat
```

Figure 4: The receiver circuit has successfully connected to the WiFi

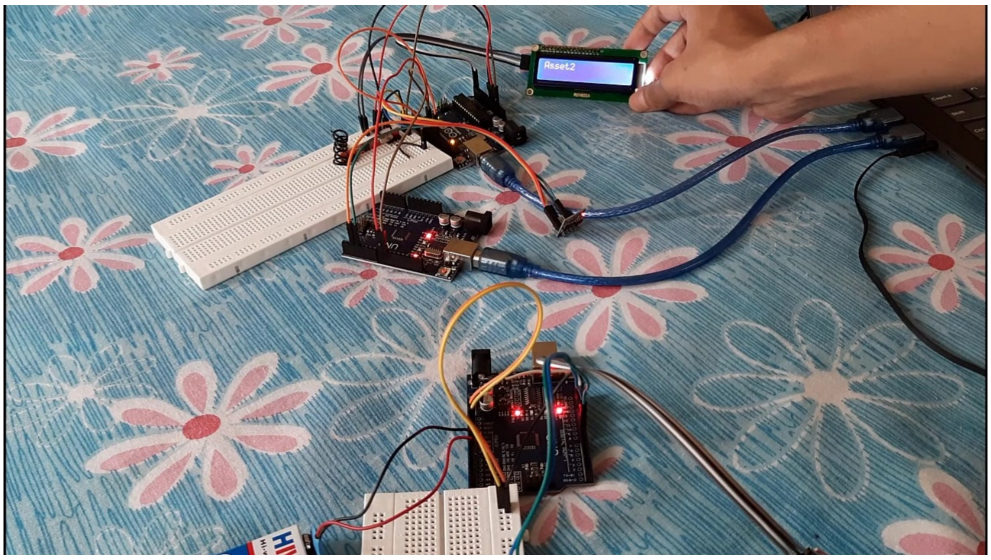
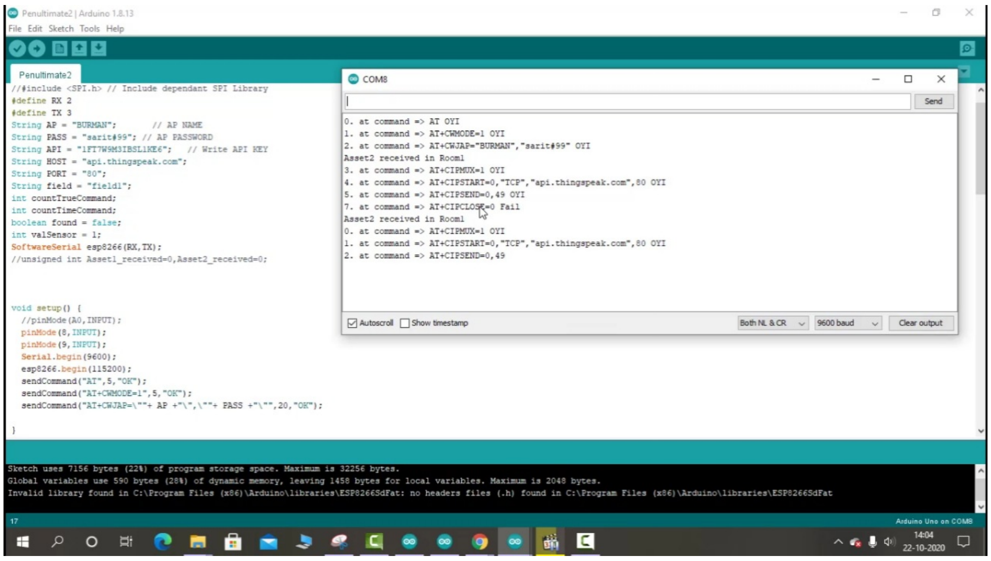


Figure 5: After Asset2 arrives near the receiver circuit the LCD displays that it is detected



```
//#include <SPI.h> // Include dependant SPI Library
#define RX 2
#define TX 3
String AP = "BURMAN";       // AP NAME
String PASS = "sarit#99"; // AP PASSWORD
String API = "1FT7W9M3IBSL1KE6";   // Write API KEY
String HOST = "api.thingspeak.com";
String PORT = "80";
String field = "field1";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
int valSensor = 1;
SoftwareSerial esp8266(RX,TX);
//unsigned int Asset1_received=0,Asset2_received=0;



void setup() {
  //pinMode(A0,INPUT);
  pinMode(8,INPUT);
  pinMode(9,INPUT);
  Serial.begin(9600);
  esp8266.begin(115200);
  sendCommand("AT",5,"OK");
  sendCommand("AT+CWMODE=1",5,"OK");
  sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");

}
```

```
0. at command => AT OYI
1. at command => AT+CWMODE=1 OYI
2. at command => AT+CWJAP="BURMAN","sarit#99" OYI
Asset2 received in Room1
3. at command => AT+CIPMUX=1 OYI
4. at command => AT+CIPSTART=0,"TCP","api.thingspeak.com",80 OYI
5. at command => AT+CIPSEND=0,49 OYI
7. at command => AT+CIPCLOSE=0 Fail
Asset2 received in Room1
0. at command => AT+CIPMUX=1 OYI
1. at command => AT+CIPSTART=0,"TCP","api.thingspeak.com",80 OYI
2. at command => AT+CIPSEND=0,49
```

```
Sketch uses 7156 bytes (22%) of program storage space. Maximum is 32256 bytes.
Global variables use 590 bytes (28%) of dynamic memory, leaving 1458 bytes for local variables. Maximum is 2048 bytes.
Invalid library found in C:\Program Files (x86)\Arduino\libraries\ESP8266SdFat: no headers files (.h) found in C:\Program Files (x86)\Arduino\libraries\ESP8266SdFat
```

Figure 6: The data is being uploaded to the ThingSpeak server

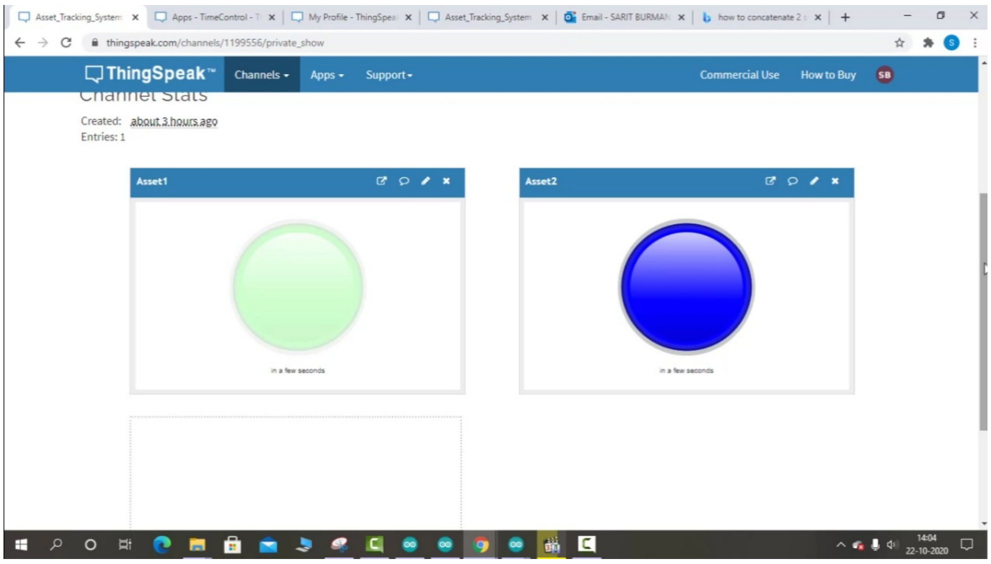
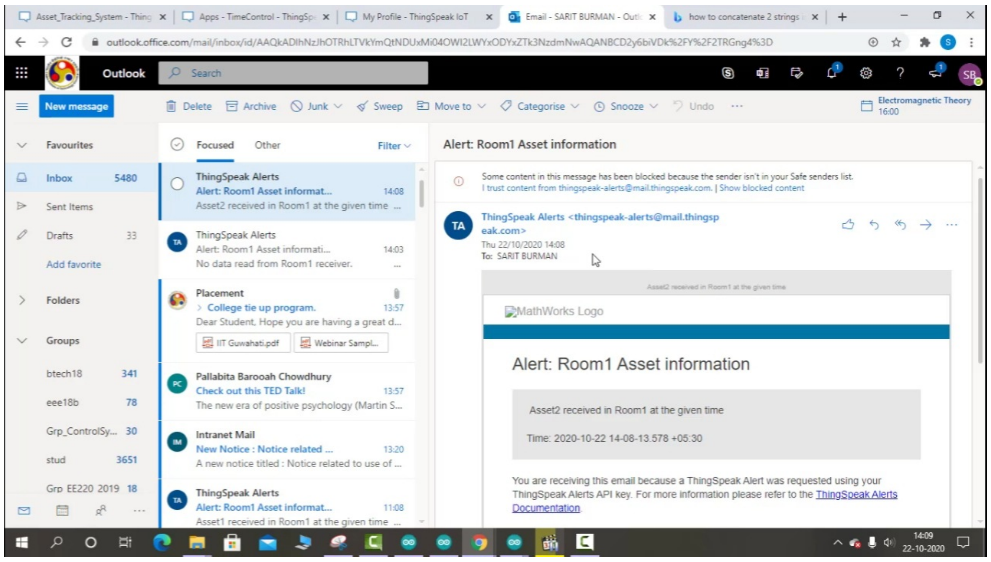


Figure 7: The cloud data reflects that Asset2 has arrived


Alert: Room1 Asset information

Some content in this message has been blocked because the sender isn't in your Safe senders list.
I trust content from thingspeak-alerts@mail.thingspeak.com. | Show blocked content

ThingSpeak Alerts <thingspeak-alerts@mail.thingspeak.com>
Thu 22/10/2020 14:08
To: SARIT BURMAN

Asset2 received in Room1 at the given time

Alert: Room1 Asset information

Asset2 received in Room1 at the given time

Time: 2020-10-22 14-08-13.578 +05:30

You are receiving this email because a ThingSpeak Alert was requested using your ThingSpeak Alerts API key. For more information please refer to the ThingSpeak Alerts Documentation.

Figure 8: Email alert received from the IoT cloud

CODES

CODE FOR THE ARDUINO CONNECTED TO WIFI MODULE

```
#include <SoftwareSerial.h>
#define RX 2
#define TX 3
String AP = "BURMAN";       // AP NAME
String PASS = "sarit#99"; // AP PASSWORD
String API = "M02XG9N2CQKL6QCR";   // Write API KEY
String HOST = "api.thingspeak.com";
String PORT = "80";
String field = "field1";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
int valSensor = 1;
SoftwareSerial esp8266(RX,TX);
unsigned int Asset1_received=0,Asset2_received=0;



void setup() {
  //pinMode(A0,INPUT);
  pinMode(8,INPUT);
  pinMode(9,INPUT);
  Serial.begin(9600);
  esp8266.begin(115200);
  sendCommand("AT",5,"OK");
  sendCommand("AT+CWMODE=1",5,"OK");
  sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");

}

void loop() {
 valSensor = getSensorData();
 if(valSensor==0 || (Asset1_received==1 && valSensor==1) || (Asset2_received==1 && va
  delay(1000);
 else
 {
   Serial.print("Asset");
   Serial.print(valSensor);
   Serial.print(" received in Room1\n");
   if(valSensor==1)
    Asset1_received=1;
   else
    Asset2_received=1;
   String getData = "GET /update?api_key="+ API +"&"+ field +"="+String(valSensor);
   sendCommand("AT+CIPMUX=1",5,"OK");
   sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");
   sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");
   esp8266.println(getData);delay(1500);countTrueCommand++;
   sendCommand("AT+CIPCLOSE=0",5,"OK");
 }
}

int getSensorData(){

  unsigned int MSB=digitalRead(8);
  unsigned int LSB=digitalRead(9);
  //Serial.println("DigitalRead:");
  //Serial.println(MSB);
  //Serial.println(LSB);
  if(MSB==0 && LSB==1)
    return 1;
  else if(MSB==1 && LSB==0)
    return 2;
  else
    return 0;
}

void sendCommand(String command, int maxTime, char readReplay[]) {
  Serial.print(countTrueCommand);
  Serial.print(". at command => ");
  Serial.print(command);
  Serial.print(" ");
  while(countTimeCommand < (maxTime*1))
  {
    esp8266.println(command);//at+cipsend
    if(esp8266.find(readReplay))//ok
    {
      found = true;
      break;
    }

    countTimeCommand++;
  }

  if(found == true)
  {
    Serial.println("OYI");
    countTrueCommand++;
    countTimeCommand = 0;
  }

  if(found == false)
  {
    Serial.println("Fail");
    countTrueCommand = 0;
    countTimeCommand = 0;
  }

  found = false;
 }
```

CODE FOR THE ARDUINO CONNECTED TO THE RF RECEIVER

```
#include <VirtualWire.h>
#include <LiquidCrystal_I2C.h>
//#include <SoftwareSerial.h>
LiquidCrystal_I2C lcd(0x3F,2,1,0,4,5,6,7,3,POSITIVE);
const int datain = 12;
void setup()
{
    pinMode(8,OUTPUT);
    pinMode(9,OUTPUT);
    //Serial.begin(9600);
    vw_set_ptt_inverted(true);
    vw_set_rx_pin(datain);
    vw_setup(2000);
    vw_rx_start();
    lcd.begin(16,2);
    lcd.clear();
}
void loop()
{
    uint8_t buf[VW_MAX_MESSAGE_LEN];
    uint8_t buflen = VW_MAX_MESSAGE_LEN;

    if (vw_get_message(buf, &buflen))
    {
      if(buf[0]=='1')
      {
        //Serial.println("Asset1");
        lcd.clear();
        lcd.print("Asset1");
        digitalWrite(8,LOW);
        digitalWrite(9,HIGH);
      }
      if(buf[0]=='2')
      {
        //Serial.println("Asset2");
        lcd.clear();
        lcd.print("Asset2");
        digitalWrite(8,HIGH);
        digitalWrite(9,LOW);
      }
    }
}
```

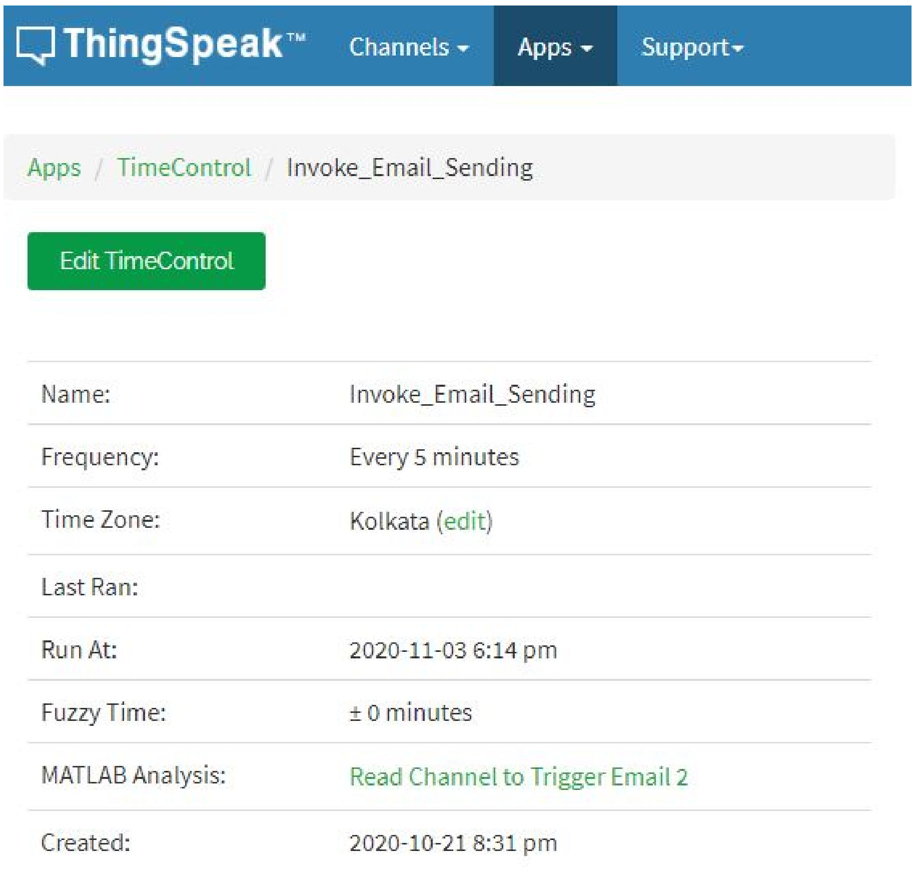
CODE FOR THE ARDUINO CONNECTED TO THE RF TRANSMITTER

```
#include <VirtualWire.h>
char *data;
void setup()
{
  pinMode(8,INPUT);
  pinMode(9,INPUT);
  vw_set_ptt_inverted(true);
  vw_set_tx_pin(12);
  vw_setup(2000);
}

void loop()
{
  unsigned int MSB=digitalRead(8);
  unsigned int LSB=digitalRead(9);
  if(MSB==0 && LSB==1)
    data="1";
  else if(MSB==1 && LSB==0)
    data="2";
  else
    data="0";
  vw_send((uint8_t *)data, strlen(data));
  vw_wait_tx();
  delay(2000);
}
```

CODE FOR MATLAB ANALYSIS IN THINGSPEAK

```
channelID = 1199556;

% Provide the ThingSpeak alerts API key.  All alerts API keys start with TAK.
alertApiKey = 'TAKYCEJ08GA6GFATXK5W8';

% Set the address for the HTTTP call
alertUrl="https://api.thingspeak.com/alerts/send";

% webwrite uses weboptions to add required headers.  Alerts needs a ThingSpeak-Alerts
options = weboptions("HeaderFields", ["ThingSpeak-Alerts-API-Key", alertApiKey ]);

% Set the email subject.
alertSubject = sprintf("Room1 Asset information");

% Read the recent data.
[data,time] = thingSpeakRead(channelID);

% Check to make sure the data was read correctly from the channel.
if isempty(data)
    alertBody = ' No data read from Room1 receiver. ';
else
    % Set the outgoing message
    if (data == 1)
        alertBody = ' Asset1 received in Room1 at the given time';
    elseif (data == 2)
        alertBody = ' Asset2 received in Room1 at the given time';
    end
end

 % Catch errors so the MATLAB code does not disable a TimeControl if it fails
try
    webwrite(alertUrl , "body", alertBody, "subject", alertSubject, options);
catch someException
    fprintf("Failed to send alert: %s\n", someException.message);
end
```

TIMECONTROL IN THINGSPEAK WEBSITE



Name:	Invoke_Email_Sending
Frequency:	Every 5 minutes
Time Zone:	Kolkata (edit)
Last Ran:	
Run At:	2020-11-03 6:14 pm
Fuzzy Time:	± 0 minutes
MATLAB Analysis:	Read Channel to Trigger Email 2
Created:	2020-10-21 8:31 pm

Figure 9: Time Control triggering the MATLAB code every 5 mins

USER MANUAL

System overview

- The model of the asset tracking system consists of 2 main parts- (a) the asset-detector or the RF trackers that are supposed to be placed in various rooms in which the assets are to be placed, and (b) the assets themselves.

- For the purpose of demonstration and due to limited hardware, I have used only a single arduino board with a RF transmitter to behave as an asset. But it has been provided with 2 select lines whose values if 2'b01, the asset acts as Asset1 and if it is 2'b10, the asset is Asset2. The select lines can be manually changed to get the experiment done for 2 assets.

- The asset-detector circuit consists of an RF receiver connected to an Arduino which is further connected to a 16×2 LCD and another Arduino board. Whenever an asset(the RF transmitter circuit) comes near the RF receiver, it sends out a signal which contains an unique code which identifies the particular asset. The receiver circuit then displays the received asset's ID on the LCD.

- The RF receiver circuit further connects to another Arduino board which has a ESP8266 WiFi module connected to it. The function of this part of the circuit is to take the data(the received asset's ID) and upload it to a IoT cloud via WiFi connection.

- The IoT cloud I used is ThingsSpeak by Mathworks. It has facility to receive data in multiple fields, but I used only a single field which is basically the assetID. In my model, there are only 2 possible values for the assetID, namely 1 and 2. But the technique can be directly extended to more than 2 assets to any number of assets.

- The receiver circuit connects to the ThingsSpeak website via the Write API key of the particular channel "AssetTrackingSystem" that I created in the website and uploads the data to the field1(named AssetID field) of the channel.

- The uploaded data is visible in many desirable forms such as line graph, lamp indicator etc. I used the lamp indicator since it would be apt in the context of visualizing if a particular asset has arrived or not. The lamp for Asset1 is OFF when the Asset1 has not arrived yet and turns ON as soon as it arrives. Similarly for Asset2 and more if we have other assets. In my case I have only 2 lamp indicators.

- I used some MATLAB code for creating a program which will send the data of any newly arrived asset to the administrator via Email. Also, I added a TimeControl which will periodically trigger the MATLAB program for generating the user alert.

Steps to use

- For using the system, we have to set up the circuit as I have shown in the diagrams attached.

- Then, we have to program the 2 arduinos in the receiver circuit with the respective codes provided . Also, we need to program the transmitter arduino.

- Initially we need to power ON the receiver circuit by connecting the 2 arduinos to the PC. Open the serial monitor for the Arduino that is connected to the WiFi module.

- We should notice that the Arduino connects to the WiFi and then does nothing.

- Then, Power the transmitter Arduino with a 9 V battery and set the select lines to 2'b01.

- The receiver LCD should immediately display Asset1. The serial monitor now shows that the system is connecting to the ThingsSpeak server and uploading the data.

- After data is successfully uploaded, the same should be reflected in the ThingsSpeak website.

- The server then sends an Email alert to the given email that Asset1 is received.

DEMO VIDEO LINK

https://youtu.be/uTaMmNux4do