REAL TIME LOCATION TRACKER FOR CRITICAL HEALTH PATIENTS

VIT-AP University,

G-30, Inavolu,

Beside AP Secretariat Amaravati,

Andhra Pradesh 522237

MEMBERS INVOLVED:

Aishwarya – 20BCN7118

Vedanth – 20BCN7115

Yaswanth – 20BCI7331

Rakshan – 20BES7040

Mahindra – 20BES7026

Hemanth – 20BCN7168

UNDER THE GUIDANCE OF PROF

Dr. Kumar Debasis

Assistant Professor Sr. Grade 1

ABSTRACT

Internet of Things (IoT) is the technology that helps in communication between machines, circuits, and different types of devices. This feature has applications in the health care industry to benefit as sensors, actuators, and hardware support the technology behind IoT. In the health care sector, in case of an emergency, it is very crucial to know the exact location of the patient so that different critical health care services can be made available at the right time and place. This problem can be solved by using GPS coordinates. In this paper, an IoT device is made which locates the exact GPS coordinates of the patients to the server. Moreover, using the web interface on the server and Google Maps, doctors and hospital staff can track the exact location of the patient and serve him. This system made can also be used on wild animals, school students and transport services where the location is an essential parameter. The system is made using sensors like GPS Neo 6m, Arduino, GSM Sim800L. Here, different IoT platforms, Different IoT tools, sensors, different programming libraries, APIs usages, server hosting and needed AT commands with website interface are implemented and discussed in detail to make a low-cost GPS tracker.

CONTENTS

1. INTRODUCTION

IoT Technology is the layered technology architecture, where one-layer acts as a sensing data, the second layer can put/process the data in the needed form, the third layer can define different high-end protocols and interfacing logic which helps to send the data. The data is sent to the server where different channels on the server keep on receiving different data from different applications. The server processes the data, and the decision made on the data values is conveyed back to the user. To build any IoT based system, the developer has to understand the different layer phases of IoT. Multiple tools are available to strengthen the IoT architecture, are Database, external interfaces, visualizations, connectivity, and management plan are also discussed here. The IoT architecture with its different layers are explained. Various sensors are studied to make the proposed system. Out of these sensors, the sensors chosen are based on their efficiency, compactness and of course the cost. GPS Neo 6m is the sensor used to sense the GPS coordinates and GSM Sim800L sends the data over the cloud using the mobile network and acting as a gateway. Data is sensed and till it is being sent to the cloud, the interfacing is served by Arduino. By using IoT technology, Different sensors and APIs, many systems can be developed for different domains like health care , security, supply chain management to monitor, control track and analyze different physical entities.

2. BACKGROUND

The IoT based wireless system should be using a small number of sensors to maintain the Quality of Service (QoS) in terms of communication, routing, energy and processing. Because if a random faulty node is present in the system then the QoS degrades [11]. GPS tracker is used in many applications like military, vehicle industries, in transportation industries and health care industries. When such system is made using display units [12] and all it increases the system cost, processing load and reduces the battery life. When such GPS based tracking system is integrated with RFID, database systems and made it centralized [13], it can be used for security services to keep track of the systems with actively moving units. But such system is oversize, needs RFID, DB, centralized control and GPS trackers. GPS sensor gives effective coordinates describing our altitude, longitude and latitude position on the earth [14]. The basic aim of this paper is to make a GPS tracking system which can be used to track real time position of patients. The problem in other GPS sensors are they continuously monitor the patient which is not energy efficient and the other such available products are costly and proprietary. The GPS tracker should be flexible in terms of monitoring time to save the energy, it should take reading after a particular time interval only based on the systems need. System design should be as much as compact. To make the system compact and cost effective Arduino is used but whenever Arduino is interfaced with more than one sensor, the communication issue arises. Here that problem is solved and system is made up of such hardware and software which are easily available and cost effective.

3. PROBLEM DEFINITION AND OBJECTIVES

Consider an IoT based Health Care system, which tracks and reports the real-time location of the patient to the cloud. The response team can locate the user based on his live location and provide emergency services.

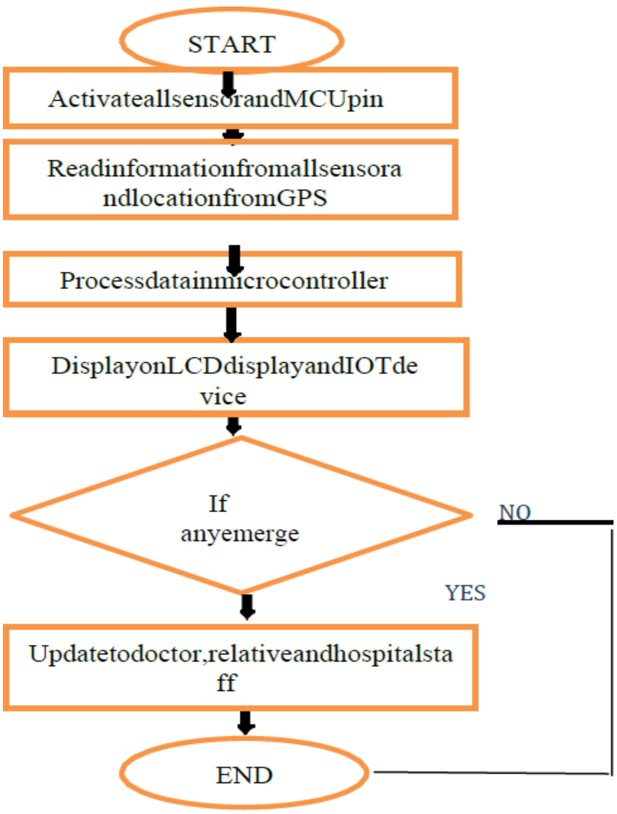
The initial tasks include:

1. Comprehending the technology
2. Development tools of IoT
3. Connectivity in IoT
4. Privacy and Security challenges in IoT
5. Cloud Services of IoT

4. METHODOLOGY/PROCEDURE

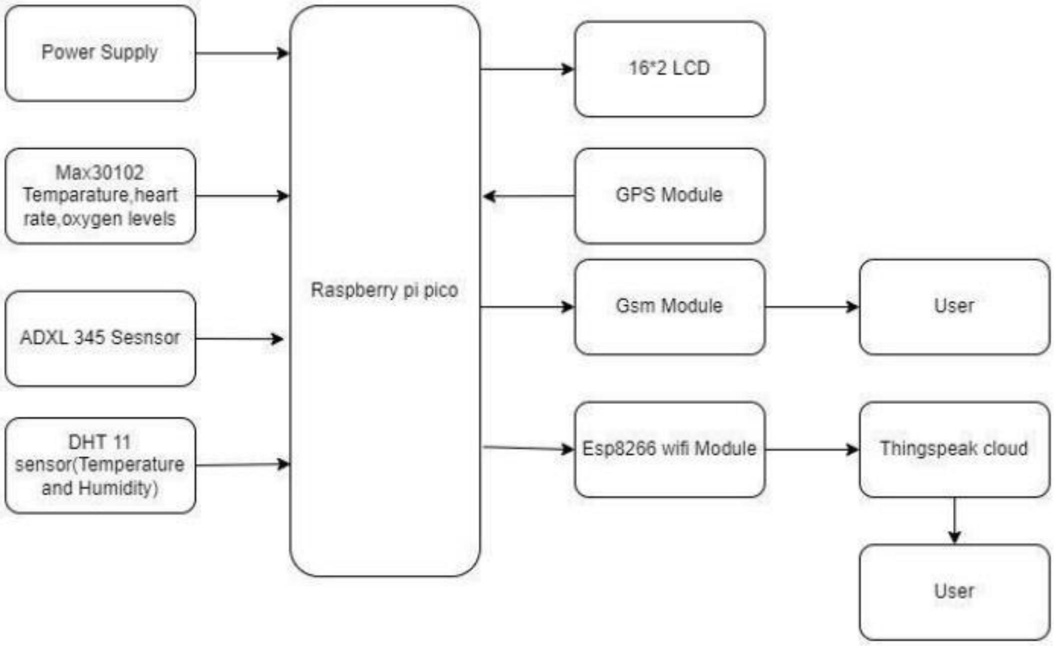
In the proposed system the temperature sensor, Humidity sensor, Heartbeat sensor and Body movement sensors are used. This all sensors are interfaced with micro controller. Once device will get power on, all sensor will get activated. Sensor will start the reading information from human body. Once any emergency will occur, that will be detected by sensor and micro controller. The micro controller used is Raspberry pi pico. The micro controller reads the coordinate for live location and it will send to IOT server.

Flowchart:



Flowchart says when the device is turned on, all sensors and MCU pin is activated. The information is read from all sensors and the location is fetched from GPS. The data is processed in the microcontroller. The information read from the sensors and the location is displayed on LCD display and also updated in IOT cloud account. If there is any emergency based on the set parameters the information is updated to doctor, relative or hospital staff. If there is no emergency the process is ended.

Block diagram:



HARDWARE IMPLEMENTATION:

- ESP8266 wifi Module

ESP8266 is a WIFI SOC (system on a chip) chip device. It is a highly integrated chip designed to provide full internet connectivity in a small package. ESP8266 can be used as an external WIFI module. It provides wifi to micro controlled board.

- GPS Module

Global Positioning System (GPS) is a satellite- based system that uses satellites and ground stations to measure and compute its position on Earth. GPS is also known as Navigation System with Time and Ranging (NAVSTAR) GPS. GPS receiver needs to receive data from at least 4 satellites for accuracy purpose. GPS receiver does not transmit any information to the satellites. This GPS receiver is used in many applications like smartphones, Cabs, Medical, electronics circuit. Fleet management etc.

- GSM Module

It is a chip or circuit that will be used to establish communication between a mobile device or a computing machine and a GSM. Global System for Mobile communication (GSM) is digital cellular system used for mobile devices. It is an international standard for mobile which is widely used for long distance communication. There are various GSM modules. We used SIM800L here. It connects to the mobile network to receive calls and send and receive text messages.

- GSM Module

It is a chip or circuit that will be used to establish communication between a mobile device or a computing machine and a GSM. Global System for Mobile communication (GSM) is digital cellular system used for mobile devices. It is an international standard for mobile which is widely used for long distance communication. There are various GSM modules. We used SIM800L here. It connects to the mobile network to receive calls and send and receive text messages.

- LCD Display 16*2

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like any prototype, circuits, mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

- MAX30102 Sensor

MAX30102 sensor is used to detect blood oxygen and heart rate. First, infrared radiation is sent and reflected by hitting the finger, and then the amount of oxygen in the blood is determined by measuring the wave amplitude. Heart rate is also obtained by analysing the time series response of this radiation.

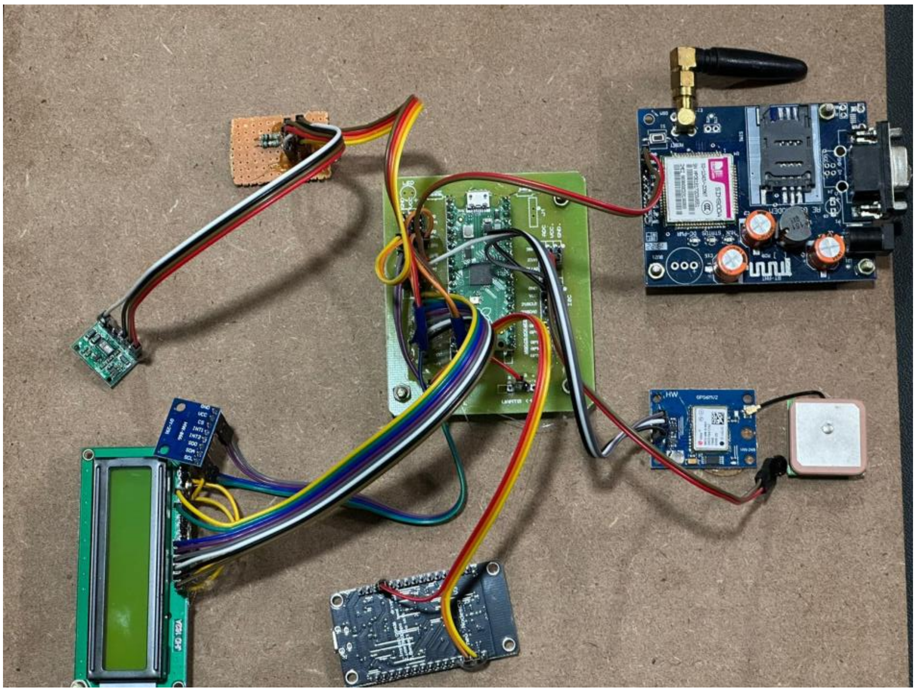
- DHT11 Sensor

DHT11 is a single wire digital humidity sensor, which gives relative humidity in percentage. DHT11 is a single wire digital humidity which provides humidity serially. DHT11 sensor provides humidity value in percentage in relative humidity (20 to 90% RH) DHT11 sensor uses resistive humidity measurement component.

5. RESULTS AND DISCUSSION

Temperature sensor, Heartbeat sensor, oxygen sensor are interfaced with pin of microcontroller. LCD display connected with digital GPIO pin. LCD interface required four data pin, RS pin and enable pin. Once microcontroller will start the execution of program, all sensor read signal from human body. IOT module is interfaced with TX and RX pin of microcontroller. GPS module also required TX and RX pin. GPS module working on serial communication. GSM module is used to send a message in the case of emergency. The GSM module doesn't require any internet connection to send the message. It is required to create an account in the IOT cloud where the result of the information read is stored. The results in IOT cloud is displayed in the form of graph. The message is sent to the doctor, relative and the hospital staff if the parameters matches the emergency conditions. The signals of the sensor is sent to microcontroller and there the signals are computed and displayed in the LCD display in digital form. The GSM module contains the numbers of the doctor, relative of the patient and hospital. The GPS module is used to know the location of the patient to make it easy for the hospital staff to give necessary medical aid as soon as possible. In this project, a system or device is being developed for real time location tracking for critical health patients. In this system different type of sensors are interfaced for different parameter of human body. GPS location finder is interfaced with microcontroller serial pin. All sensor will read the different parameter and signal from body. Based on that parameter of body, the sensor will send information to microcontroller. The patient's health is monitored and updated in the cloud account. The information can be accessed from anywhere and anytime. The device tracks the live location of the project in emergency. Not only the human's health monitoring and tracking is done, it is also done for animals.

PICTURE OF WORKING MODEL:



6. CONCLUSION AND FUTURE SCOPE

The prototype developed is simple to use. When the finger is placed, the sensor collects information and are displayedon LCD. The doctors can get basic medical report using this model in less time. The results can be viewed in cloud account to get the average results of the patients. After collecting the data, if the parameters indicate emergency then it updates the staff or guardian through SMS.

Future scope is to achieve benefits like:

- Improved Chronic Condition Management
- Reduced Emergent Situations and Readmissions
- Increased Revenue Streams
- Reduced Burden on Healthcare Systems
- Improved Patient Outcomes
- Better Quality of Care
- Increased Education
- Increased Patient Accountability

7. REFERENCES

[1] http://projects2u.com/embeddedsystems-major.html

[2] Bandana Mallick and Ajit Kumar Patro, Heart rate monitoring system using fingertip through Arduino and processing software

[3] Purnima and Puneet Singh, Zigbee and GSM Based Patient Health Monitoring System, International Conference on Electronics and Communication System, Vol. 05, 2014, PP 102- 103.

8. CODES IN APPENDIX

```
#include <SoftwareSerial.h>
SoftwareSerial gsm_Serial(8,9);
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"
MAX30105 particleSensor;
const int rs = 10, en = 11, d4 = 12, d5 = 13, d6 = 14, d7 =15;
#include <LiquidCrystal.h>
int buz=3;
int pb=26;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int spo2=0,ssum=0,hsum=0;
int cnt=0;
float temperatureF;
const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
byte rates[RATE_SIZE]; //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred
float beatsPerMinute;
int beatAvg;
int kk=0;
int ab=0;
int xval,yval,sval;
#include <TinyGPS.h>
TinyGPS gps;
float flat=0, flon=0;
void read_gps()
{
   bool newData = false;
  unsigned long chars;
  unsigned short sentences, failed;
  for (unsigned long start = millis(); millis() - start < 1000;)
  {
    while (gsm_Serial.available())
    {
      char c = gsm_Serial.read();
      if (gps.encode(c))
        newData = true;
    }
  }

  if (newData)
  {

    unsigned long age;
```

```
    gps.f_get_position(&flat,&flon,&age);

  }
}
void setup()
{
    Serial1.begin(115200);
  Serial.begin(9600);
 lcd.begin(16, 2);
  pinMode(buz,OUTPUT);
//   sensors.begin();
    accel.begin();
  digitalWrite(buz,0);
  digitalWrite(buz,0);
  pinMode(pb,INPUT_PULLUP);
  lcd.print("   WELCOME");
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz speed
  {
    Serial.println("MAX30105 was not found. Please check wiring/power. ");
    while (1);
  }
  particleSensor.setup(); //Configure sensor with default settings
  particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate sensor is running
  particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
  particleSensor.enableDIETEMPRDY();
  delay(2000);
  wifi_init();
}

void loop()
{
  long irValue = particleSensor.getIR();

  if (checkForBeat(irValue) == true)
  {
    //We sensed a beat!
    long delta = millis() - lastBeat;
    lastBeat = millis();

    beatsPerMinute = 60 / (delta / 1000.0);

    if (beatsPerMinute < 255 && beatsPerMinute > 20)
    {
      rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the array
      rateSpot %= RATE_SIZE; //Wrap variable

      //Take average of readings
      beatAvg = 0;
      for (byte x = 0 ; x < RATE_SIZE ; x++)
        beatAvg += rates[x];
      beatAvg /= RATE_SIZE;
```

```
    }
   }

   //Serial.print("IR=");
   //Serial.print(irValue);
   //Serial.print(", BPM=");
  // Serial.print(beatsPerMinute);
   float temperatureF = particleSensor.readTemperatureF()-6; //Because I am a bad global citizen
   beatAvg=beatsPerMinute;
   if(beatAvg<40)
   spo2=0;
   else if(beatAvg>140)
   spo2=map(beatAvg,140,255,80,50);
   else if(beatAvg>=40 && beatAvg<60)
   spo2=map(beatAvg,40,60,70,100);
   else
   spo2=map(beatAvg,60,140,100,80);

   ssum=ssum+spo2;
   hsum=hsum+beatAvg;
   cnt=cnt+1;
   if(cnt==10)
   {
     cnt=0;
     spo2=ssum/10;
     beatAvg=hsum/10;
 ssum=0;
 hsum=0;
 if (irValue < 50000)
   {

   lcd.clear();
        lcd.print(" No finger?");
     Serial.println(" No finger?");

   beatAvg=0;
   spo2=0;
   temperatureF=0;
   }

 else
   {
   if(beatAvg>40)
   {
        sensors_event_t event;
     accel.getEvent(&event);
     int xval=event.acceleration.x;
     int yval=event.acceleration.y;
     int pbval=digitalRead(pb);
     // bt=((bt*10) + 4)/5 + 28;
       lcd.clear();
```

```
lcd.setCursor(0,0);
lcd.print("T:");
lcd.setCursor(2,0);
lcd.print(temperatureF);
lcd.setCursor(5,0);
lcd.print("H:");
lcd.setCursor(7,0);
lcd.print(beatAvg);
lcd.setCursor(10,0);
lcd.print("s:");
lcd.setCursor(12,0);
lcd.print(spo2);
lcd.setCursor(0,1);
lcd.print("X:");
lcd.setCursor(2,1);
lcd.print(xval);
lcd.setCursor(5,1);
lcd.print("Y:");
lcd.setCursor(7,1);
lcd.print(yval);
 lcd.setCursor(11,1);
lcd.print("B:");
lcd.setCursor(13,1);
lcd.print(pbval);
  Serial.println("H:");
  Serial.println(beatAvg);
  Serial.println("S:");
  Serial.println(spo2);
 //     Serial.println("T:");
//        Serial.println(bt);
  if((spo2>30 && spo2<70) || pbval==0|| beatAvg>130 || temperatureF>100 || xval<-6 || xval>6 || yval>6 ||
yval<-6)
  {
    digitalWrite(buz,1);
  //  send_sms(1);
   upload_iot(beatAvg,temperatureF,spo2,xval,yval,pbval);
    delay(1000);
    digitalWrite(buz,0);
    ab=ab+1;
    if(ab>4 && kk==0)
    {
      kk=1;
    }
  }
  else
  {
    digitalWrite(buz,0);
    ab=0;
    kk=0;
  }
```

```
  }
  else
  {
lcd.clear();
      lcd.print("Reading..");


  beatAvg=0;
  spo2=0;
  temperatureF=0;
  }

  }

  }
  }
void send_sms(int sts)
{
gsm_Serial.println("Sending SMS...");
gsm_Serial.println("AT");
delay(1000);
gsm_Serial.println("ATE0");
delay(1000);
gsm_Serial.println("AT+CMGF=1");
delay(1000);
gsm_Serial.print("AT+CMGS=\"9494669293\"\r\n");// Replace x with mobile number
delay(1000);
if(sts==1)
gsm_Serial.println("SOS: Abnoraml Soldier Conditon : ");
//gsm_Serial.println("https://www.google.com/maps/search/?api=1&query=" + String(flat,6)+ "," +
String(flon,6));
gsm_Serial.println("https://www.google.com/maps/search/?api=1&query=" + String(16.4963)+ "," +
String(80.5007));
delay(100);
gsm_Serial.println((char)26);// ASCII code of CTRL+Z
delay(6000);
//gsm_Serial.println("AT");
//delay(1000);
//gsm_Serial.println("ATE0");
//delay(1000);
//gsm_Serial.println("AT+CMGF=1");
//delay(1000);
//gsm_Serial.print("AT+CMGS=\"9347232567\"\r\n");// Replace x with mobile number
//delay(1000);
//if(sts==1)
//gsm_Serial.println("SOS: Abnormal Soldier health condition At ");
////gsm_Serial.println("https://www.google.com/maps/search/?api=1&query=" + String(flat,6)+ "," +
String(flon,6));
//gsm_Serial.println("https://www.google.com/maps/search/?api=1&query=" + String(16.4963)+ "," +
String(80.5007));
//delay(100);
```

```
//gsm_Serial.println((char)26);// ASCII code of CTRL+Z
//delay(2000);
}
void wifi_init()
 {
   Serial1.println("AT+RST");
  delay(2000);
  Serial1.println("AT+CWMODE=1");
  delay(1000);
  Serial1.print("AT+CWJAP=");
  Serial1.write("");
  Serial1.print("SRC 24G"); // ssid/user name
  Serial1.write("");
  Serial1.write(',');
  Serial1.write("");
  Serial1.print("src@internet"); //password
  Serial1.write("");
  Serial1.println();
  delay(1000);
 }
void upload_iot(int x,int y,int z,int p,int q,int r)
{
  String cmd = "AT+CIPSTART=\"TCP\",\"";
  cmd += "184.106.153.149"; // api.thingspeak.com
  cmd += "\",80";
  Serial1.println(cmd);
  delay(1500);
 String getStr ="GET /update?api_key=WJ35MIZJXKIVQRUL&field1=";
  getStr += String(x);
  getStr +="&field2=";
  getStr += String(y);
   getStr +="&field3=";
  getStr += String(z);
   getStr +="&field4=";
  getStr += String(p);
   getStr +="&field5=";
  getStr += String(q);
    getStr +="&field6=";
  getStr += String(16.4963);
   getStr +="&field7=";
  getStr += String(80.5007);
    getStr +="&field8=";
  getStr += String(r);
  getStr += "\r\n\r\n";
  cmd = "AT+CIPSEND=";
  cmd += String(getStr.length());
  Serial1.println(cmd);
  delay(1500);
  Serial1.println(getStr);
  delay(1500);
 }
```