Real Time Bus Tracking System Using Linkit One

A. Krishnamoorthy, V. Vijayarajan, G. Sivashanmugam, N. Visakeswaran

Abstract--- The public transport system is growing at a very rapid pace to meet the needs of the overgrowing population, each day thousands of buses are carrying millions of people to their destinations, there is an increased burden on the public transport system. The situations created difficulty in managing and tracking the public transport. Since there are a lot of buses travelling through same route and it creates immense amount of congestion and thereby the buses fail to reach their destinations at promises time, and this has become a major problem especially in cities, where most of the bus users are office going people, who need to keep up with time. With this kind of uncertainty in the transport system, people have to trade off with their time standing and waiting at the bus stop, for their bus to reach. It would be better if user is certain about when the bus is about to reach, investing his time in something productive rather than wasting it standing at the bus stop. In this paper, we use the power of user's smartphone combined with cloud computing to create a IOT system to provide certainty in the transport system. We use a very efficient and powerful IOT prototyping platform LinkIt ONE board to achieve this, integrate GPS, GPRS and other sensors are makes it stand out from other platforms such as Raspberry Pi and Arduino. We use the GPS and GPRS on the board to get the co-ordinates of the bus, these co-ordinates are sent to a Firebase cloud service and then co-ordinates are fetched in real-time to a React-Native mobile application installed on user's device. It was observed that, the proposed system brings down the waiting time of the user. This system can be used as a base to solve much more complex problems in transport system, like scheduling the buses and avoiding congestion.

Index Terms— real-time tracking, LinkIt One, IOT, cloud computing, react-native, Firebase

I. INTRODUCTION

In the way individuals move and travel opens a lot of opportunities for the frameworks is the principle issue which assume an undeniably important part. It is an exceptionally savvy mode of transport. Therefore, it is reason for overwhelming movement and roadwork and so on. The vast majority of the transports are deferred in time. At the transport end individuals need to sit tight for long time without knowing when the transport will arrive. Anyone who needs to utilize general society transportation framework, can't discover the season of entry of specific transport at the specific goal even at their homes and plan their take-off from home as needs be. Yet, because of unforeseen deferrals in rush hour gridlock clog the transport entry time can't be ensured. Our primary concentration is to give such a framework to remote client that will diminish waiting time for transport and will give him every single vital insight with respect to the landing/take-off time of the transport, its genuine area and expected holding up time. So, to discover the present area of a transport and the dynamic entry time a precise following framework is required.

GPS and GSM prove to be the core technologies when it comes to tracking for any form. The configuration and the setup of GSM and GPS in necessary. The GSM and GPS cannot work independently and require a base through which they can be integrated, we use LINKIT ONE board which has both of them integrated and does not need any additional modules for the working, which makes it stand ahead of other development platforms like Arduino and Raspberry pi. The GSM and GPRS together provide us with the current location of the bus, this raw data has to be sent to the user's device, therefore we need for middleware between the LINKIT ONE and the users Device i.e. the smart phone, this can be achieved in lot of ways, we can send the co-ordinates using SMS or send messages through internet on GSM, in this project we use advanced technology like cloud to transfer the co-ordinates to the user in a real-time fashion.

II. BUS-TRACKINGSTAGES

The model concentrates on three major stages: 1) Defining theCo-ordinate 2) Data push to cloud service 3) Fetching data to mobile device. The first step in any processing is to make sure the data is subjected to cleaning in order to remove noise. Step two involves actual task sending the co-ordinate data collected from the GPS to cloud service like Firebase.

Step three completes the delivery of the obtained co-ordinates of the bus to user device, this is achieved by fetching the data stored in the Firebase using API's and Sockets and displaying them in real time fashion on user's device.

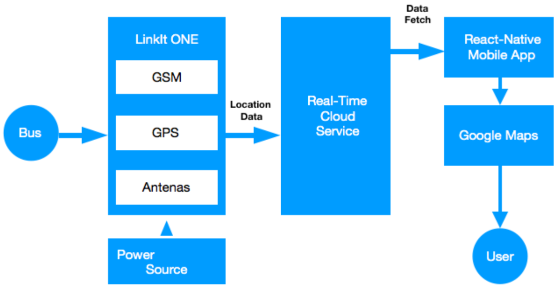


Fig 1: Architecture

Manuscript received February 01, 2019

A. Krishnamoorthy, Assistant Professor (Senior), School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Vellore, Tamil Nadu, India

V. Vijayarajan*, Associate Professor, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Vellore, Tamil Nadu, India. (e-mail: vijayarajan.v@vit.ac.in)

G. Sivashanmugam, Assistant Professor (Senior), School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Vellore, Tamil Nadu, India

N. Visakeswaran, UG Student, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Vellore, Tamil Nadu, India.

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

A. *Initial setup*

The linkit one is a IOT prototyping microcontroller created and discharged by MediaTek and SeeedStudio. This was made for every one of the creators in this world for their wearables and IOT ventures. The unique thing about this chip is that it has Bluetooth, Wi-Fie, GPS and GSM officially introduce in a little bundle. It utilizes an indistinguishable programming dialect from Arduino making it simpler to construct and model ventures. The pinout is to some degree like Arduino so all Arduino shields are perfect with it. It likewise has a SD card space, sound jack and separate attachments for I2c and UART.

Before we can start tracking the bus, some initials setup needs to get done. The first step is to configure the hardware, that is LinkIt one board. In order to achieve this, Arduino IDE is downloaded, by default the Arduino IDE does not support the LinkIt One Board. As suggested in the documentation of the MediaTek website we need to download the OS specific USB COM port for the board. After this is done, the board can be connected to the and the IDE recognizes the board, now we install the board manager for the LinkIt one which downloads all the packages necessary for the board to function. In order for GPRS to function and send the co-ordinates to the cloud, we need a working SIM with edge service enables. Insert this SIM into the bottom of the board

Arduino IDE isn't sufficient to transfer a code to LinkIt one. You will require an extra module for IDE called the Linkit one programming improvement unit (SDK) which makes this board good with IDE. Without the SDK, it would not work with the board chose as Arduino Uno.

Afterfinishing introducing the SDK, at long last module your Linkit one board through a smaller scale USB link. The drivers for it would consequently be introduced. Ensure that the little switch close to the battery connector ought to be set to UART while another switch close to the SPI connector is set to SPI.

B. *Obtaining Co-ordinates*

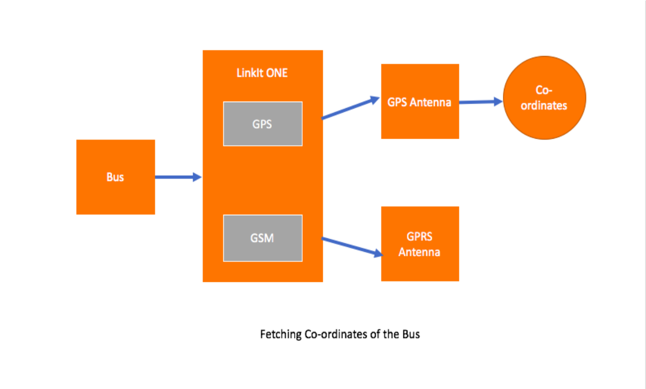
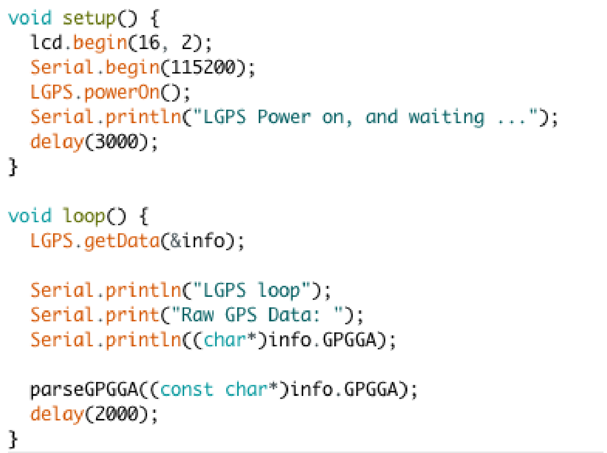


Fig 2: Co-ordinates Fetching using GPS.

After doing the initial setup for the board we are ready to obtain the GPS co-ordinates using the GPS modules present on the board. We write the code in Arduino IDE for fetching the current location of the bus and upload it to the board.

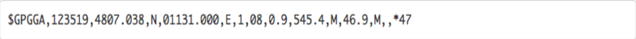
The GPS on the linkit is dealt with by a devoted chip (named MT3332). A GPS Antenna must be utilized with the goal that the chip can get signals from satellites, a huge number of miles away. The linkit SDK offers a superb working Arduino draw for utilizing this GPS highlight; all the grimy (code) work is dealt with by the LGPS library.



```
void setup() {
  lcd.begin(16, 2);
  Serial.begin(115200);
  LGPS.powerOn();
  Serial.println("LGPS Power on, and waiting ...");
  delay(3000);
}

void loop() {
  LGPS.getData(&info);

  Serial.println("LGPS loop");
  Serial.print("Raw GPS Data: ");
  Serial.println((char*)info.GPGGA);

  parseGPGGA((const char*)info.GPGGA);
  delay(2000);
}
```

Fig 3: Code for Configuring GPS.

Basically, the GPS chip on the linkit return a String message, an example of which is shown below:



```
$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47
```

Fig 4: Format of GPS data.

Presently, there's a variety of information in this string which is isolated by commas. We should break down each bit of information:

- GGA - Global Positioning System Fix Data
- 123519 - GPS fix taken at 12:35:19 UTC *
- 4807.038,N - Latitude 48 deg 07.038' N *
- 01131.000,E - Longitude 11 deg 31.000' E *
- 1 - Fix quality (0 means invalid, 1 means SPS GPS fix, other numbers refer to other advanced stuff we don't need to care about)
- 08 - Number of satellites being tracked *
- 0.9 - Horizontal dilution of position *
- 545.4,M -Altitude, Meters, above mean sea level *
- 46.9,M -Height of geoid (mean sea level) above WGS84 ellipsoid
- (empty field) - time in seconds since last DGPS update
- (empty field) - DGPS station ID number - (we would never use this so don't care)
- *47 - the checksum data, always begins with an asterisk

In most applications using the GPS, we would only have to care about some pieces of data.

C. *Data Push to Cloud*

There are various of options available in the market to provide cloud storage, but not all of them provide the real-time data access. In this paper, we use The Google Firebase, to achieve the real-time data access.

However, we need linkit one does not support the data transfer to firebase directly. We use the Google IOT cloud to achieve this. The Google IOT has integration support for the Firebase. The co-ordinate data obtained is first stored in the google cloud and from there it is sent to Firebase.

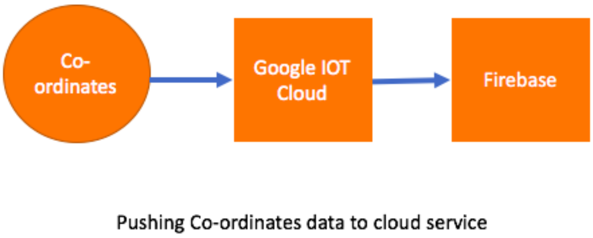


Fig 5: Data push to cloud.

D. Data Fetch to Mobile Application

For creating a mobile application, we are using a latest front-end technology that is, React-Native. React-native is a very powerful JavaScript framework which allows the developer to create a mobile app for both Android and IOS at the same time without having the need to write the code twice for both the platform separately.

The cloud service Firebase to which we pushed our co-ordinate data, provides us with the API's to access this data. Using React-Native we make API call to the firebase server where our data is stored. This way we get our data inJSON format. We parse this JSON data to get our co-ordinate information and give it as an input to the Google Maps API, which displays the position of the bus on a map.

The bus is constantly in movement and the co-ordinates keep changing in this case, we need something which keeps us updated with the co-ordinates. Firebase provides us with sockets which keeps us updated with the data. We integrate the sockets in react-native app to update the bus position on the map.

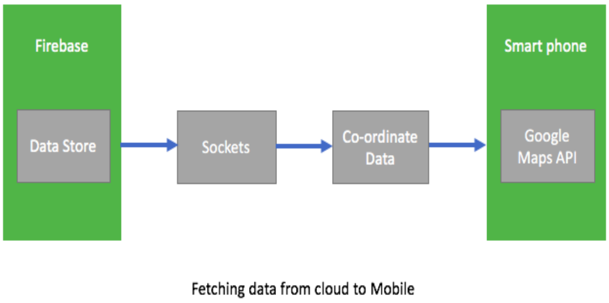


Fig 6: Data fetch form firebase to Mobile App.

III. RESULTS AND FINDINGS

After going through all the stages of development we are successfully able to implement the model. The system works as expected and provides the expected result. We are able to track the Bus in real time and help the user to manage his time efficiently.

The following figure [fig 7] displays the basic setup of the LinkIt one and the sensor antennas involved. With this setup, we can carry on with fetching the data from the GPS. The protruding things are the antennas and not the sensors themselves, the sensors are integrated on the board, this is a very unique feature of LinkIt One which make it stand out from other platforms. The SIM and microSD slots are also visible in the figure.

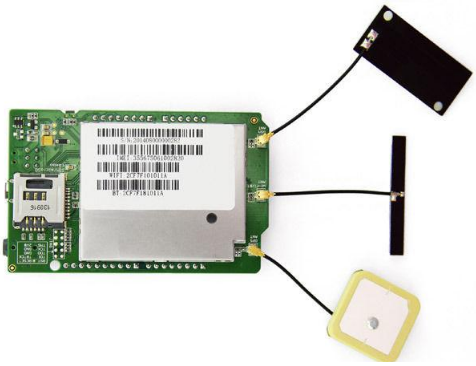
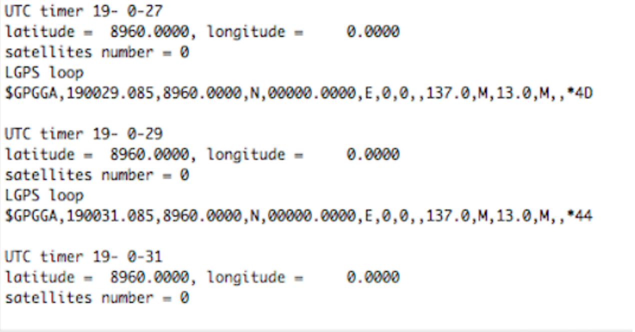


Fig 7: LinkIt ONE setup.

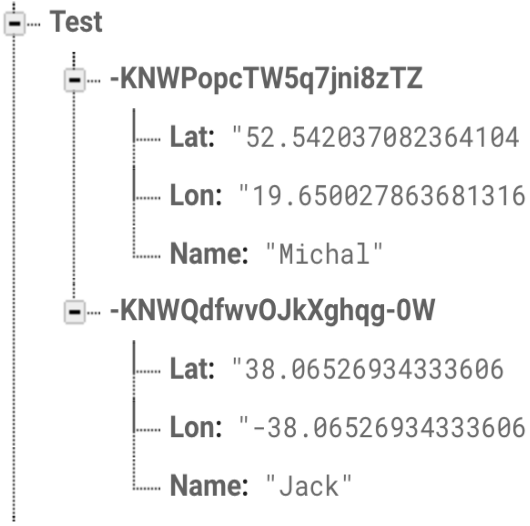
After we have uploaded the code to the board using Arduino IDE we get the result as displayed in the below figure [fig. 8], in the IDE console. We can see the co-ordinate data from the GPS being fetched.



```
UTC timer 19- 0-27
latitude =  8960.0000, longitude =     0.0000
satellites number = 0
LGPS loop
$GPGGA,190029.085,8960.0000,N,00000.0000,E,0,0,,137.0,M,13.0,M,,*4D

UTC timer 19- 0-29
latitude =  8960.0000, longitude =     0.0000
satellites number = 0
LGPS loop
$GPGGA,190031.085,8960.0000,N,00000.0000,E,0,0,,137.0,M,13.0,M,,*44

UTC timer 19- 0-31
latitude =  8960.0000, longitude =     0.0000
satellites number = 0
```

Fig 8: Output in Arduino data.

The Data is stored in Firebase an object format as displayed below. This object is passed as a response from the server when we request for location information from the mobile app. We can observe how the Latitude and Longitude data is stored in the object.



```
Test
  -KNWPopcTW5q7jni8zTZ
    Lat: "52.542037082364104
    Lon: "19.650027863681316
    Name: "Michal"
  -KNWQdfwvOJkXghqg-0W
    Lat: "38.06526934333606
    Lon: "-38.06526934333606
    Name: "Jack"
```

Fig 9: Data in Firebase.

This data from the firebase is integrated to the react-native mobile app using the API's and sockets to provide a result as displayed in the figure below [fig. 10], the user's mobile acts as client and fetches the data and provides that as input for the Google Maps API which them displays the result on a Map in real-time.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

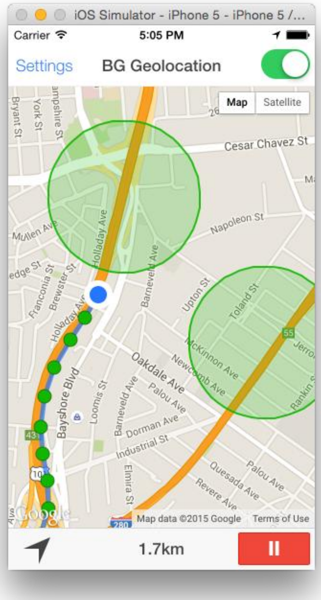


Fig 10: Tracking data on Mobile App

IV. CONCLUSION

The proposed system provides the exact location of the Bus to the user in a real-time fashion on his/her smartphone client that uses Google Maps API. The Firebase provides data in a very efficient manner, making it possible to achieve the real-time capability. This brings down waiting time of the user and substantially increases the efficiency of the user. Indian mobile network services may fail at some remote locations through which the bus passes, at this point he system fails dues to lack of internet service to push the data to cloud. The general evaluation result is system performs very well and solves the problem it addresses, given that the network connection is always strong.

FUTURE SCOPE

The system can be used as a base for many other complex related problems as scheduling and congestion management of public transport system. Furthermore, the setup can be used to track the efficiency of the buses, keeping a check on the efficiency of the employees and also tracking the bus in case of some accidents. Also, such a model can be used by the travel companies to allow the passenger to share the bus tracking details to their family members.

REFERENCES

1. Dr. Saylee Gharge, Manal Chhaya, Gaurav Chheda, Jitesh Deshpande, “Real time bus monitoring system using GPS,” An International Journal of Engineering Science and Technology, Vol. 2, Issue 3, June 2012.
2. Abid Khan, Ravi Mishra, “GPS-GSM based tracking system,” International Journal of Engineering Trends and Technology, Vol. 3, Issue 2, pp: 161-164, 2012.
3. A. Krishnamoorthy & V. Vijayarajan (2017) Energy aware routing technique based on Markov model in wireless sensornetwork, International Journal of Computers and Applications, DOI: 10.1080/1206212X.2017.1396423.
4. S. P. Manikandan, P. Balakrishnan, “An Efficient real time query system for public transportation service using Zigbee and RFID,” International Journal of Research in Communication Engineering, Vol. 2, No. 2, June 2012.
5. Swati Chandurkar, Sneha Mugade, Sanjana Sinha, Pooja Borkar, “Implementation of real time bus monitoring and passenger information system,” International Journal of Scientific and Research Publications, Vol. 3, Issue 5, May 2013.
6. Pankaj Verma, J. S. Bhatia, “Design and development of GPS-GSM based tracking system with Google map based monitoring,” International Journal of Computer Science, Engineering and Applications, Vol. 3, No.3, June 2013.
7. Madhu Manikya Kumar, K. Rajesekhar, K. Pavani, “Design of punctually enhanced bus transportation system using GSM and Zigbee,” International Journal of Research in Computer and Communication Technology, Vol. 2, Issue 12, December 2013.
8. R.Maruthi, C.Jayakumari “SMS based Bus Tracking System using Open Source Technologies,” International Journal of Computer Applications (0975 – 8887) Volume 86 – No 9, January 2014 College of Engineering,Chennai SSN College of Engineering, Chennai.
9. N.Vijayalashmy, V. Yamuna, G. Rupavani, A. Kannaki@VasanthaAzhagu,“ GNSS based bus monitoring and sending SMS to the passengers,” International Journal of Innovative Research in Computer and Application Engineering, Vol. 2, Special Issue 1, March 2014.
10. R. Manikandan, S. Niranjani, “Implementation on real time transportation information using GSM query response system,” Contemporary Engineering Sciences, Vol. 7, No.11, pp: 509-514, 2014.
11. G. Raja, D. NaveenKumar, G. Dhanateja, G. V. Karthik, Y. Vijay Kumar, “Bus Position monitoring system to facilitate the passengers,” International Journal of Engineering Science and Advanced Technology(IJESAT), Volume-3, Issue-3, pp: 132-135, 2014.
12. S. Eken, A. Sayar, “A Smart bus tracking system based on location aware services and QR codes,” IEEE International Symposium on Innovations in Intelligent and Applications Proceedings, pp: 299-309, 2014. [13] Vishal Bharte, Kaustubh Patil, Lalit Jadhav, Dhaval Joshi , “ Bus Monitoring System Using Polyline Algorithm ,” International Journal of Scientific and Research Publications, Volume 4, Issue 4, April 2014.
13. Shekhar Shinde, Vijaykumar Nagalwar, Nikhil Shinde, B.V. Pawar, “Design of E-City Bus Tracking System ,”Int. Journal of Engineering Research and Applications, ISSN : 2248-9622, Vol. 4, Issue 4(Version 9), April 2014, pp.114-117.