Asian Journal of Applied Science and Technology (AJAST)
(Open Access Quarterly International Journal) Volume 2, Issue 2, Pages 933-946, 2018

# Implementation of IoT Based Intelligent Transportation System

Arun.V[1] and Dr.M.Poongothai[2]

[1]PG Scholar, Department of Electronics and Communication Engineering, Coimbatore Institute of Technology, Coimbatore, Tamilnadu, India.
[2]Associate Professor, Department of Electronics and Communication Engineering, Coimbatore Institute of Technology, Coimbatore, Tamilnadu, India.

*Article Received: 27 January 2018* *Article Accepted: 23 February 2018* *Article Published: 25 May 2018*

A B S T R A C T

Rapid vehicular growth partnered with ever increasing population, rural to urban migration and economic upsurge has put immense amount of pressure on transportation infrastructure, especially on traffic management practices in urban India. Intelligent transportation system (ITS) is a well-known method to simplify, or nevertheless minimize traffic problems. Intelligent transportation system (ITS) refers to integrated application of communication, control and information processing technologies to the transportation infrastructure and vehicles. ITS is based on the collection, processing, integration and dissemination of information, based on static and real-time data obtained from sensors on-board vehicles. In this work, IoT based Intelligent Transportation System is proposed which aims to locate and navigate the position of CIT college bus. Also, it aims to provide user understandable through MIT app Inventor app, Google Map API and Thing speak cloud. The proposed systems is integrated with GPS and cloud technology together and transfer the real time live streaming data to cloud server. The GPS module is used to get geographic coordinates at regular time intervals and the vehicle location is updated to Thing-speak cloud IoT server using NodeMCU. Mobile application has been developed which is used to provide the details to the user regarding the bus location using his smart phone via Google map. Also, it allows the user to monitor the location of the college bus in anywhere around the world using mobile app /web browser. The proposed system has been implemented using Arduino IDE and MIT app Inventor.

Keywords: IoT, Vehicle Tracking, Google map API, Thingspeak Cloud and Intelligent Transportation System.

## 1. INTRODUCTION

ITS can be broadly defined as the use of technology for improving transportation systems. The major objectives of ITS is to evaluate, develop and analyze and integrated new technologies and concept to achieve traffic efficiency, improve environmental quality, save energy, conserve time, and enhance safety. ITS plays one of the major role in contributing towards smart city development. The Internet of Things connects objects, equipped with a communication unit and several sensors, to the Internet in order to achieve information-exchange, intellectualized identifying, positioning, monitoring and managing. The Internet of Things can logically be divided into a perception layer, a network layer, a service layer and an application layer. The perception layer senses, gathers information, and the network layer enables the connectivity between the different items making use of Internet technology, while the service layer offers services to the application or to the end user for further intelligent processing. Undoubtedly, this strong vision of Internet of Things could add new dimensions to Intelligent Transportation Systems and it will have a high impact on applications and services. However, there are many challenges such as real-time traffic management, seamless connectivity, vehicle location prediction, security and privacy, interoperability, communications, associated with Internet of Things that needs to be addressed.

## 2. LITERATURE REVIEW

*1. Bhavana and Godavarthi, "Design and Implementation of Vehicle Navigation System in Urban Environments using IoT", International Journal of Science and Research,vol.2, pp.56-62.*

This paper proposed [1] Design and Implementation of Vehicle Navigation System in Urban Environments using IoT. The working principle is comparison between the current vehicle paths already specified paths into the file system. In the raspberry pi's file system taken from vehicle owners through android phone using application. Hence the driver drive the vehicle only on the vehicle owner's specified path but if the driver drive in wrong path the message alert will be sent from this system to the vehicle owners mobile and also sent speaker alert to driver through audio jack .If the vehicles speed goes beyond the specified value of the speed, then warning message will be sent to owner mobile. This system also takes care of the traveller's safety by using gas leakage and temperature sensors.

*2. Dimil Jose,Santath Prasad and Sridar, "Intelligent Vehicle Monitoring using GPS and Cloud Computing" International Symposium on Big Data and Cloud Computing, pp.440-446*

This paper presented [4] Intelligent Vehicle Monitoring using GPS and Cloud Computing. In the system solution for accident to rising safety. There are also various sensors which relay information like fuel level, driver conditions and tire pressure. The vital information like the vehicle location, speed is gathered by the GPS which is fitted in the vehicle and transmitted in near-real-time via cellular or satellite communication to a centralized server maintained in the cloud network. This information is then available for the authorized users in real time and each licensed vehicle owner can access the data in cloud using a web portal anytime anywhere. This system thus provides an accurate positioning of the vehicle, speed, driver's condition and provides an intelligent monitoring of the vehicle remotely.

*3. Thiyagarajan Manihatty Bojan and Uma maheswaran, "An Internet of Things Based Intelligent Transportation System , "IEEE International Vehicular Electronics and Safety , vol.1, pp.174-179.*

This paper proposed [2], An Internet of Things based Intelligent Transportation System. The system has three components the sensor system, monitoring system and the display system. The sensor system has Global Positioning System (GPS), Near Field Communication (NFC), Temperature and Humidity sensors, which are always connected with the internet via a GSM network to track the location, commuter and ambience inside the bus. The monitoring system is used to extract the raw data from the sensors database, convert it into a meaningful context, triggers some events with in the bus and provide information to the bus driver. The display system is used to show the context data (bus and travel related information) to all the commuters in the bus stop.

*4. Kamaraj, Radha and Priyanka , "Intelligent Transportation System using Integrated GPS optimized Reader", International Science Technology Engineering and Management, vol.2, pp.332-336.*

This paper [7], proposed Intelligent Transportation System using Integrated GPS Optimized Reader. This proposed a new solution to send quick intimation to their parents about the presence of students through school or college bus. By this way track the children and save them early if any mischief is happening. For this using an RFID reader, GPS tracker and GSM module along with an Arduino mega 2560 and its software. RFID readers can read the RFID tags of the students. GSM system is used to send their presence to their parents through SMS. To achieve this, a passive or active tracking device with GPS is used. Passive devices store GPS location, speed and time. Once the vehicle returns to a predestined point, the device is removed and the data downloaded to a computer for evaluation. By this way get arrival and departure locations of the students. Active devices also collect the same information, but usually transmit the data in near-real-time via cellular or satellite networks to a computer or data center for evaluation. It is used to track the location of a vehicle from organizing itself.

## 3. METHODOLOGY

Intelligent transportation system (ITS) refers to integrated application of communication, control and information processing technologies to the transportation Infrastructure and vehicles. ITS plays one of the major role in contributing towards smart city development. The proposed work aims to track the vehicle location through GPS data and also monitor the location in webpage through Google Map.

### *3.1 INTERNET OF THINGS*

The Internet of Things (IoT) is defined as a paradigm in which objects equipped with sensors, actuators, and processors communicate with each other to serve a meaningful purpose. More than 85% of systems are unconnected and do not share data with each other or the cloud. One such technology that facilitates the interconnection is the INTERNET OF THINGS. IoT also means making devices smarter, by connecting them online to acquire and process data.

Today IoT has touched every corner of the world. Internet of Things is simply an interaction between the physical and digital worlds. The digital world interacts with the physical world using a plethora of sensors and actuators. IoT devices are equipped with sensors, actuators and processors.

Sensors are used for interacting with the physical environment. The data collected by the sensors have to be stored and processed by the processor. The actuator performs the desired action depending on the value from the sensor. The storage and processing of data can be performed at that remote server. The pre-processing of data can be done at the sensor level. The communication between the IoT devices is wireless. Sensors, actuators, compute servers, and the communication network forms the core infrastructure of an IoT framework. The Internet of Things finds various applications in healthcare, fitness, education, entertainment, social life, energy conservation, environmental monitoring, home automation, and transport systems.

### *3.2 BLOCK DIAGRAM OF INTELLIGENT TRANSPORTATION SYSTEM*

The Fig.3.1 illustrates, the main component of the proposed system is E-tracker device with inbuilt GPS/GSM, ATmega microcontroller & SD card which is used to get geographic coordinates at regular time intervals. It is connected to Node MCU through serial connection. So that E-Tracker sends the GPS coordinates to Node MCU which is used to transmit the GPS coordinates and update the vehicle location to a Thing-speak cloud (IoT platform) database. Thing speak cloud continuously receives GPS coordinates every sec and plotted as graph. Google Map API is used to access the Google Map features and geographical location. Google Map API reads the GPS coordinates from the Thing speak Cloud and display the vehicle location on the Map in the mobile app or web browser

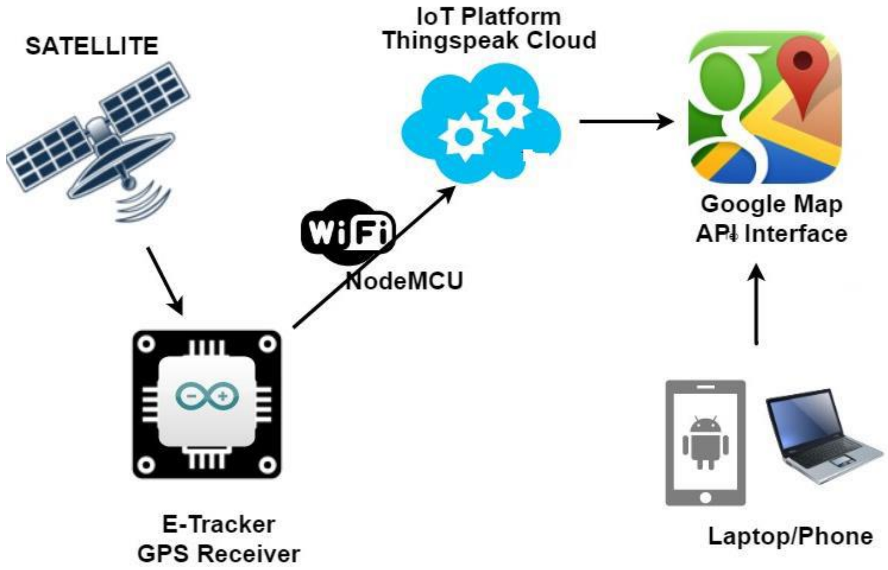


Fig.3.1 Proposed system architecture

### *3.3 FLOW CHART OF E-TRACKER*

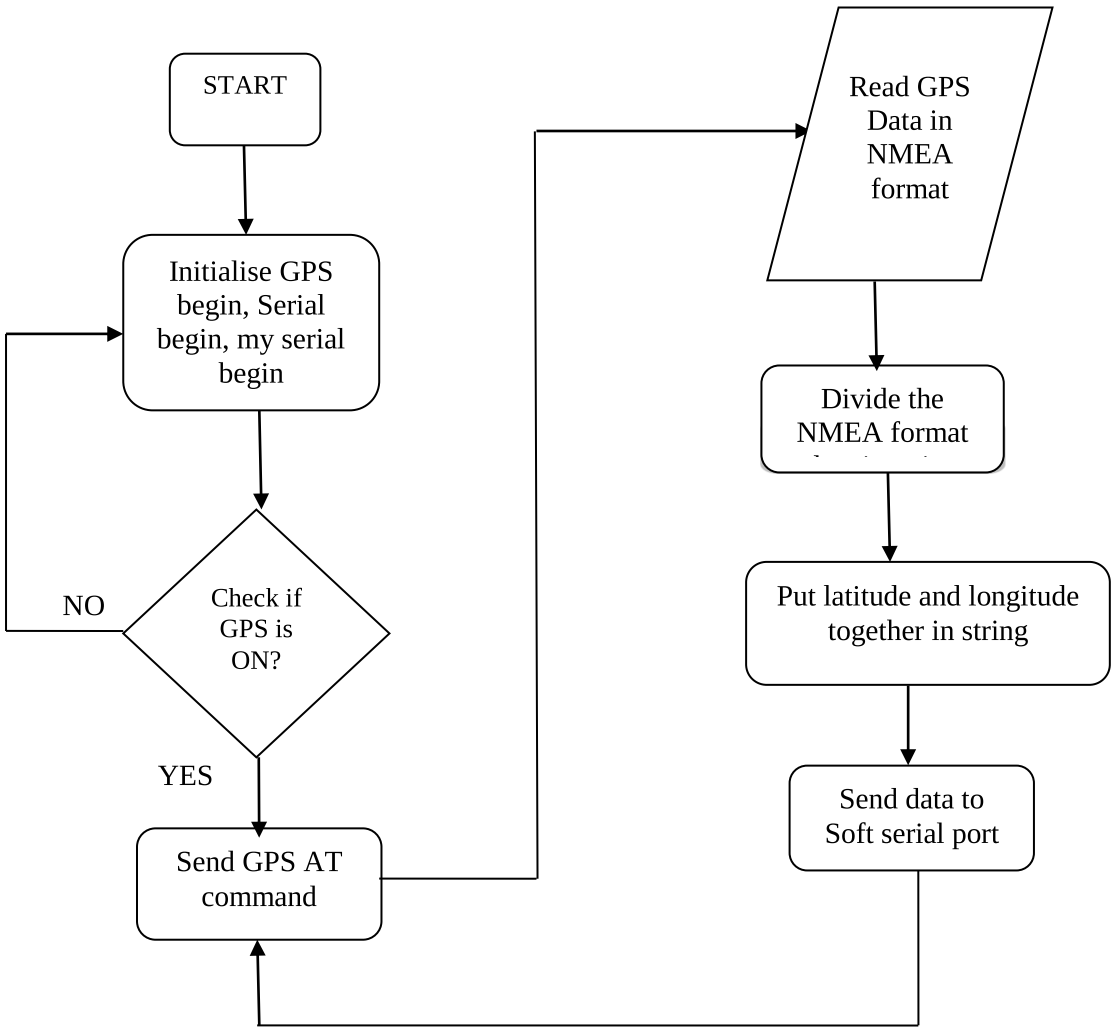


Fig.3.2 Flowchart of E-tracker coding

The above Fig.3.2 illustrates the sequence of activities carried out while updating the GPS coordinates on E-Tracker device. When the connection is established, the status of GPS is checked, if it is true E-Tracker gets the GPS coordinates (latitude and longitude) from the satellite through mobile network.

NodeMCU is connected to E-Tracker through soft serial connection .The obtained E-Tracker's GPS values send to NodeMCU. This process is continuously repeated until the connection is disabled.

### *3.4 FLOWCHART OF NODEMCU*

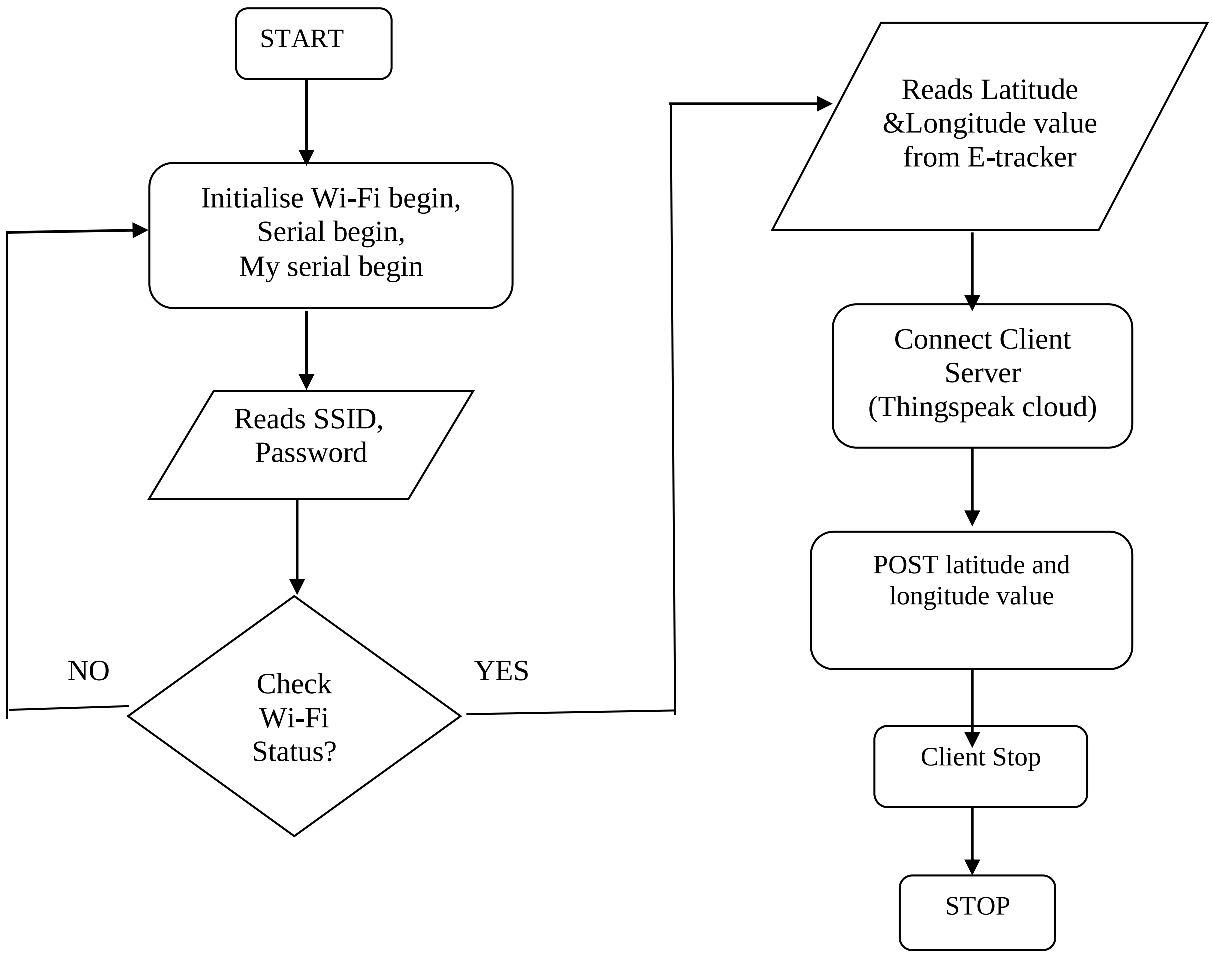


Fig.3.3 Flowchart of NodeMCU coding

The above Fig.3.3 illustrates the sequence of activities carried out by NodeMCU module. When the connection is established, NodeMCU's Wi-Fi reads SSID and Password in local network and gets the GPS coordinates from E-Tracker device through serial communication and the values are posted to Thingspeak channel through API key.

## 4. HARDWARE AND SOFTWARE ENVIRONMENT DESCRIPTION

### *4.1 E-Tracker Module*

The E-Tracker is an ATmega328 and SIM808 which provides the flexibility of arduino along with the added features of SIM808 which has GSM, GPS and bluetooth functionality as shown in Fig 4.1 and 4.2. SIM808 module is a complete Quad-Band GSM/GPRS module which combines GPS technology for satellite navigation. The compact design which integrated GPRS and GPS in a SMT package will significantly save both time and costs for customers to develop GPS enabled applications. Featuring an industry-standard interface and GPS function, it allows variable assets to be tracked seamlessly at any location and anytime with signal coverage. It has high GPS receive sensitivity with 22 tracking and 66 acquisition receiver channels.

***Features of E-Tracker***

i. Quad-band 850/900/1800/1900MHz
ii. GPRS multi-slot class12 connectivity: max. 85.6kbps(down-load/up-load)
iii. GPRS mobile station class B
iv. Controlled by AT command (3GPP TS 27.007, 27.005 and SIMCOM Enhanced AT commands)
v. Supports real time clock
vi. Supply voltage range 5V ~ 12V
vii. Integrated GPS/GNSS and supports A-GPS
viii. Supports 3.0V to 5.0V logic level
ix. Low power consumption, 1mA in sleep mode
x. Supports GPS NMEA protocol
xi. Standard SIM Card

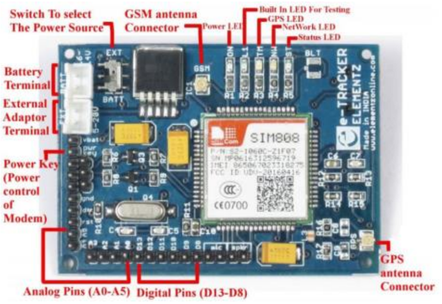


Fig.4.1 E-Tracker front view

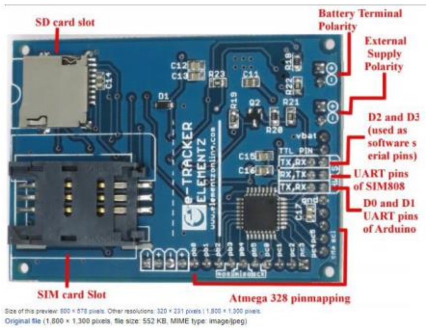


Fig.4.2 E-Tracker back view

### *4.2 NodeMCU Module*

NodeMCU is an open source IoT platform. It consists of firmware which runs on ESP8266. It is programmable, low cost, and smart Wi-Fi enabled.

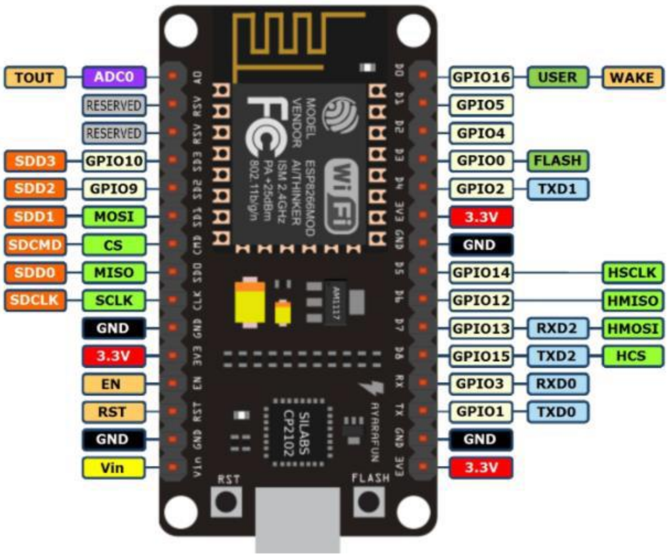


Fig 4.3 Pin description of Node MCU

NodeMCU has ESP-12 based serial Wi-Fi integrated on-board to provide GPIO, PWM, ADC, I2C and built-in USB-TTL serial as shown in Fig 4.3. NodeMCU has powerful on-board processing and storage capabilities that allows the sensors to be integrated with it.

***Specifications of NodeMCU***

i. Node MCU (Model: ESP8266-12E) is an open source IoT platform.
ii. It includes firmware which runs on the ESP8266 Wi-Fi SoC.
iii. Operating voltages: 3.3-3.6V
iv. Uses 802.11 b/g/n Wi-Fi standards
v. 10 GPIO, every GPIO can be PWM, I2C, 1-wire,
vi. Uses IPV4, HTTP, FTP, UDP & TCP network protocols
vii. Operating frequency range: 2.4-2.6 GHz
viii. Configured in both Android & iOS devices.

### *4.3 THINGSPEAK CLOUD PLATFORM*

Thing speak allows user to build applications around data collected by sensors. It offers near real-time data collection, data processing, and also simple visualizations for its users. Data is stored in so-called channels, which provides the user with a list of features. Each channel allows store up to 8 fields of data, using up to 255 alphanumeric characters each. There are also 4 dedicated fields for positional data, consisting of: description, latitude, longitude, and elevation. All incoming data is time and date stamped and receives a sequential ID. Once a channel has been created, data can be published by accessing the Thing speak API with a write key, a randomly created unique alphanumeric string used for authentication. Consequently, a read key is used to access channel data in case it is set to keep its data private (the default setting). Channels can also be made public in which case no read key is required. Essentially, things are objects that are given sensors to collect data. Data is sent and received via simple "Hypertext Transfer Protocol" (HTTP) POSTs, much like going to a web page and filling out a form. This communication happens through plaintext, JSON or XML. The data is then uploaded to the cloud and from there can be used for a variety of purposes. In turn, data (such as commands or choosing certain options) can be gathered and communicated to the cloud, which in turn sends these messages to the object.

### *4.4 GOOGLE MAP API*

The Google Map API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures. It can also use API calls to add markers, polygons, and overlays to a basic map, and to change the user's view of a particular map area. These objects provide additional information for map locations, and allow user interaction with the map.

### *4.5 ARDUINO IDE*

Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage over other systems:

i. **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50

ii. **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

iii. **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works

iv. **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based.

v. **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

**4.6 *MIT APP INVENTOR***

App Inventor can be develop applications for android phones using a web browser and either a connected phone or an on-screen phone emulator. The MIT App inventor servers store the work and help it keep track of projects. There are two screens in MIT App inventor.

i. **Design View Screen:** Select placing design elements and the components for app as shown in Fig.4.4.

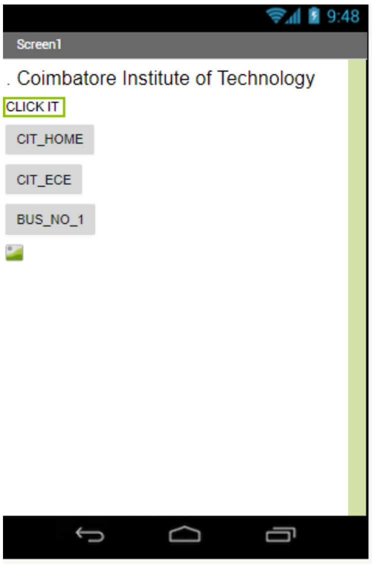


Fig.4.4 MIT App designing Page

ii. **Blocks Editor Screen:** Assemble program blocks that specify the components should behave. It assembles programs visually, fitting pieces together like pieces of a puzzle as shown in Fig.4.5.

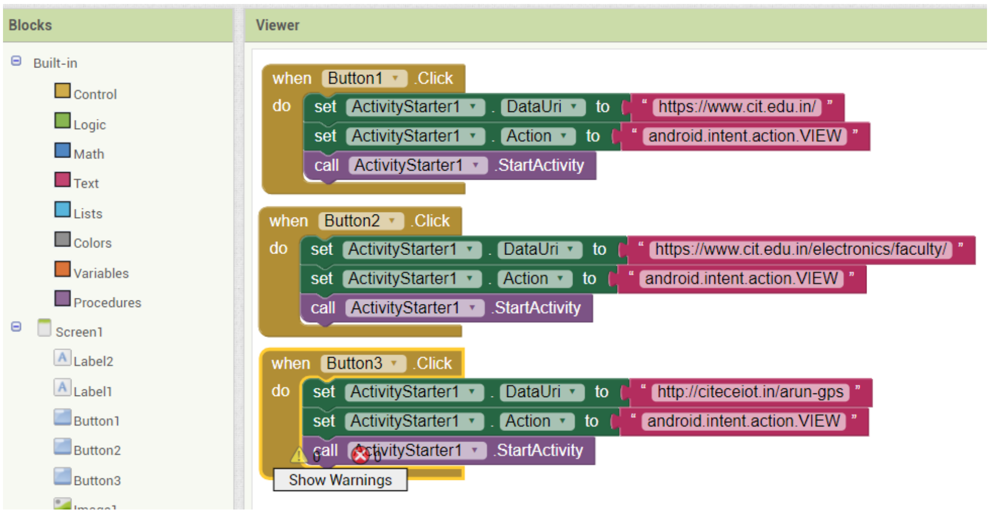


Fig.4.5 MIT App Block page

The above Fig.4.11 illustrates the Block Editor Screen in MIT App inventor. Whenever button is clicked corresponding web page can be loaded. Each button have specific block can be designed. The App Inventor development environment is supported for Mac OS X, GNU/Linux, and Windows operating systems, and several popular android phone models. Applications created with App Inventor can be installed on any android phone.

## 5. RESULT

### *5.1 GPS COORDINATES OF E-TRACKER*

```
CGNSINF: 1,1,20180204115416.000,11.019590,76.965555,453.300,0.72,222.5,1,,
Sending AT+CGNSPWR=0

OK
Got Correct Reply
Sending AT+CGNSPWR=1

OK
Got Correct Reply
Sending AT+CGNSINF

+CGNSINF: 1,1,20180204115449.000,11.019565,76.965550,439.100,1.78,39.0,1,,
Sending AT+CGNSPWR=0

OK
Got Correct Reply
Sending AT+CGNSPWR=1


+CGNSINF: 0,,,,,,,,,,,,,,,,,,,,

Sending AT+CGNSINF
LATITUDE:    11.019542
LONGITUDE:   76.965448

Sending AT+CGNSINF
LATITUDE:    11.019543
LONGITUDE:   76.965465
```

Fig.5.1 GPS coordinates received by E-Tracker using Arduino IDE

The E-Tracker module gets the latitude and longitude coordination from the satellite in real time and continuously send AT command in order to gets their GPS coordinates. Fig.5.1 shows the GPS coordinates received by E-Tracker using Arduino IDE**.**

### *5.2 GPS COORDINATE OF NODEMCU*

```
Connecting to
jrmtech
.........
WiFi connected
a :11.019563,76.965467,0,0

s1 :11.019563
s2 :76.965467
s3 :0
s4 :0
LATITUDE: 11.019563 LONGITUDE: 76.965467Waiting...
Next data :
a :11.019563,76.965467,0,0

s1 :11.019563
s2 :76.965467
s3 :0
s4 :0
LATITUDE: 11.019563 LONGITUDE: 76.965467Waiting...
Next data :
a :11.019563,76.965465,0,0
11.019563,76.965465,0,0

s1 :11.019563
s2 :76.965465
s3 :0
s4 :0
LATITUDE: 11.019563 LONGITUDE: 76.965465Waiting...
```

Fig.5.2 GPS coordinates received by NodeMCU using Arduino IDE

The NodeMCU gets the GPS coordinates from the E-tracker module through serial connection and it is connected to Wi-Fi network and send the GPS coordinates to Thingspeak cloud platform using ESP826Wi-Fi Module as shown in Fig.5.2.

### *5.3 THINGSPEAK LATITUDE AND LONGITUDE COORDINATES*

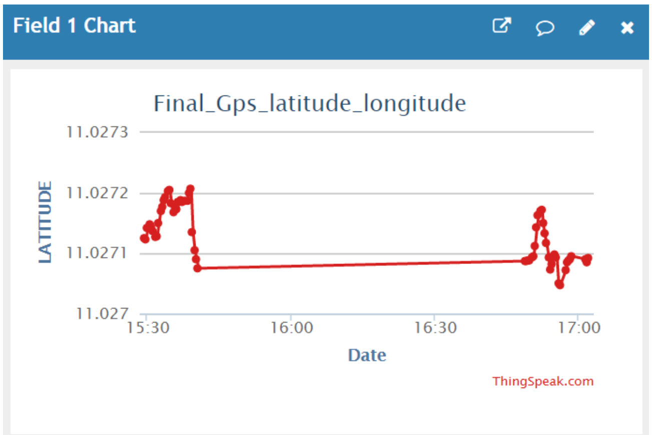


Fig.5.3 Latitude coordination received Thingspeak cloud

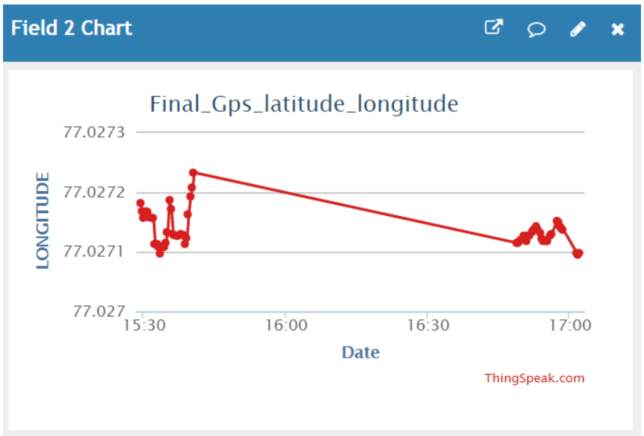


Fig.5.4 Longitude coordination received Thingspeak cloud

.

| | A | B | C | D |
|---|---|---|---|---|
| 1 | created_at | entry_id | field1 | field2 |
| 2 | 2018-03-23 08:48:10 UTC | 1 | 11.0196 | 76.9655 |
| 3 | 2018-03-23 08:48:35 UTC | 2 | 11.0196 | 76.9655 |
| 4 | 2018-03-23 08:48:56 UTC | 3 | 11.0196 | 76.9655 |
| 5 | 2018-03-23 08:49:16 UTC | 4 | 11.0196 | 76.9655 |
| 6 | 2018-03-23 08:49:36 UTC | 5 | 11.0196 | 76.9655 |
| 7 | 2018-03-23 08:49:58 UTC | 6 | 11.0196 | 76.9655 |
| 8 | 2018-03-23 08:50:20 UTC | 7 | 11.0196 | 76.9655 |
| 9 | 2018-03-23 08:50:40 UTC | 8 | 11.0196 | 76.9655 |
| 10 | 2018-03-23 08:51:01 UTC | 9 | 11.0196 | 76.9655 |
| 11 | 2018-03-23 08:51:20 UTC | 10 | 11.0196 | 76.9655 |
| 12 | 2018-03-23 09:15:24 UTC | 11 | 11.0195 | 76.9654 |
| 13 | 2018-03-23 09:15:48 UTC | 12 | 11.0195 | 76.9654 |
| 14 | 2018-03-23 09:16:11 UTC | 13 | 11.0195 | 76.9654 |
| 15 | 2018-03-23 09:16:32 UTC | 14 | 11.0195 | 76.9654 |
| 16 | 2018-03-23 09:16:51 UTC | 15 | 11.0195 | 76.9654 |
| 17 | 2018-03-23 09:17:12 UTC | 16 | 11.0195 | 76.9654 |

Fig. 5.5 Thingspeak CSV file

Fig.5.3 shows the visualization of latitude coordinate values are processed and sent to the Thingspeak server in field 1.The sampled values are sent to the server every second and values get plotted every 15 seconds .The X axis gives the date and time of real time and Y axis corresponding latitude coordination are plotted.

Fig.5.4 shows the visualization of longitude coordinate values are processed and sent to the Thingspeak server in field 2.The sampled values are sent to the server every second and values get plotted every 15 seconds .The X axis gives the date and time of real time and Y axis corresponding longitude coordination are plotted.

Fig.5.5 illustrates the Thingspeak CSV file. It can be automatically generated when field 1(latitude) and filed 2(longitude) values are uploaded in Thingspeak channel corresponding date and time will be generated.


Fig 5.6 Thingspeak and Google Map Output

Fig.5.6 illustrates the Google Map API generate the Google Map webpage on web browser. Their GPS coordinates are received from Thingspeak cloud and plotted as a location.


Fig.5.7 Bus monitoring Webpage design

Fig.5.7 shows the design Bus Monitoring webpage using HTML code .It can be used to monitor the location of the vehicles by clicking the BUS number automatically link goes to corresponding Google Map webpage.

**6. CONCUSION AND FUTURE WORK**

In recent years, the Internet of Things has developed very rapidly and has become a development trend to improve living conditions in the world as a whole. One of the feasible areas to cover with this new paradigm in the field of smart cities is the monitoring of college transport. Considering the above, an Intelligent transport system is implemented for college bus monitoring. It can be implemented in any vehicle and usable by anyone or any "thing" using the paradigm of the Internet of Things in order to provide greater security to the users of this medium of transport. The proposed system is tested to track the exact location of a moving or stationary vehicle in real-time. This system allows the user to monitor the location of the bus in anywhere around the world using mobile app and web browser. The web interface is used to transmit the location and vehicle status information to the Thingspeak cloud server. Also, Smartphone application has been developed which is used to provide the details to the user regarding the bus location using his smart phone via Google map.

The Internet of Things applied to the monitoring of the college transport will play an important role in the management of urban transport in the cities of the future. So, as future work, It consider integrating wireless communication protocols to the mobile propagation node, to add other sensors to it such as proximity sensors at the college transportation stops, in the passenger seat and integrate them to the Internet of things. It will also be necessary to create models to analyze data that provides the monitoring system using the Big Data approach.

## 7. ACKNOWLEDGMENT

The authors wish to thank the respondents who have spent their precious time and patience for participating in this project.

## REFERENCES

[1]. Bhavana and Godavarthi (2017), "Design and Implementation of Vehicle Navigation System in Urban Environments using IoT " , International Journal of Science and Research (IJSR), vol.2, pp.56-62.

[2]. Kamaraj , Radha and Priyanka (2016), "Intelligent Transportation System Using Integrated GPS Optimized Reader", International Science Technology Engineering and Management, vol.2, pp.332-336.

[3]. Elavarasi and Kuppusamy (2016), "The Smart Transportation using IoT and Intelligent Transportation System in GPS Localization", International Journal of Innovative Research in Computer and Communication Engineering, vol.4, pp.11197-11203.

[4]. Dimil Jose, Santath Prasad and Sridar V.G (2015), "Intelligent Vehicle Monitoring Using GPS and Cloud Computing", International Symposium On Big Data and Cloud Computing, pp.440-446.

[5]. SeokJu Lee, Girma Tewolde and Jaerock Kwon (2014), "Design and Implementation of Vehicle Tracking System using GPS/GSM/GPRS Technology and smartphone Application", IEEE World Forum of Internet of Things (WF-IoT), vol.2, pp.353-358.

[6]. Thiyagarajan Manihatty Bojan and Uma maheswaran (2014), "An Internet of Things based Intelligent Transportation System", IEEE International Vehicular Electronics and safety, vol.1, pp.174-179

[7]. Pham Hoang Dat, Micheal Driebug and Nguyan chi Cuong (2013), "Development of Vehicle Tracking System using GPS and GSM Modem", IEEE on open system, pp.89-94.

[8]. Tareq Binjammaz and Ali Al-Bayatti (2013), "GPS Integrity Monitoring for an Intelligent Transportation System", IEEE Transactions on Intelligent Transportation System, vol.2, pp.97-102

[9]. Shaojun Feng and Choi Look Law (2013), "Assisted GPS and its Impact on Navigation in Intelligent Transportation Systems", International Conference on Intelligent Transportation System, vol.2, pp.926-931.

[10]. Bharat kumar and Sunil babu (2013), "An Intelligent Traffic and Vehicle Monitoring System using Internet of Things Architecture", International Journal of Science and Research, vol.5, pp.853-857.