Low Power Real-Time GPS Tracking enabled with RTOS and Serverless Architecture

Sundar Anand
School of Electronics Engineering
Vellore Institute of Technology
Chennai, India
a.sundar2016@vitstudent.ac.in

Annie Johnson
School of Electronics Engineering
Vellore Institute of Technology
Chennai, India
annie.johnson2016@vitstudent.ac.in

Priyanka Mathikshara
School of Electronics Engineering
Vellore Institute of Technology
Chennai, India
priyankamathikshara.m2016@vitstudent.ac.in

Karthik R
School of Electronics Engineering
Vellore Institute of Technology
Chennai, India
karthik.r@vit.ac.in

***Abstract* –In this day and age everything is monitored and tracked. GPS tracking plays one of the most vital roles in our everyday life. Tracking entities is important in many fields including fleet management, personal security and even operations management. The proposed work details the low power analysis carried out on the built real-time GPS tracking system. Incorporating microservices and modern architecture, the system provides live tracking solution along with analyses of the travel history. The use of serverless architecture augments and provides an intelligible design for the system. Equipped with mobile and web application, this system delivers live tracking results to the end users. Customized geofences enable the email alert service to the user in case of a breach in the boundary.**

***Keywords* – GPS, RTOS, cloud computing, low power, QoS, microservices.**

I. Introduction

The concept of tracking makes our lives much easier. With exponentially advancing technologies including Internet of Things (IoT), we can enable simple everyday objects to send and receive data. In this way, these smart entities can communicate with each other, involving little or no human intervention. IoT solutions can be extended to the Global Positioning System (GPS) tracking domain. This paper discusses one such IoT solution that can be employed in GPS tracking. As an instance, vehicle tracking is chosen to explain the system.

A. Types of GPS systems

a) Classification based on transmission duration

Vehicle tracking devices are of two types: "Passive" and "Active". Passive devices are those devices that store the Global Positioning System (GPS) location, speed, vehicle details and the direction information temporarily. These systems are useful for work purposes interested in reducing the misuse of their vehicles. Passive devices store a predetermined amount of data which is permanent. They are smaller in size, easier to conceal and cheaper than active tracking devices. However, when it comes to real-time tracking applications, active trackers are preferred.

b) Classification based on data reception

Vehicle tracking can be done in three methods.
In the Automatic Vehicle Location (AVL) system, any vehicle can be tracked and monitored through the use of GPS and the Geographic Information System (GIS) [1]. However, the AVL system cannot get accurate satellite data in densely populated urban areas and indoors.
A terrestrial Radio Frequency (RF) network is used in the Assisted Global Positioning System (AGPS) system which improves the performance and accuracy of GPS receivers. Yet, the cost of sending a Short Message Service (SMS) here is a drawback.
Tags are devices used in Radio Frequency Identification (RFID) to carry out the process of retrieving data remotely and automatically [2-3]. The limitation of the RFID system is that it has a short range.

B. Cloud computing

To enhance the design and working of a system, efficient software and hardware must be incorporated. When it comes to the software, the system can be made with an easy jump start by leveraging many of the Cloud platform utilities. This created software can be made easily accessible at all times to the users with great performances around the world again by utilizing cloud Edge facilities. With respect to Hardware, Scalability and reliability are two of the most important characteristics of cloud.

Scalability refers to how the system handles resources with increasing amount of demand. This feature of the cloud allows to efficiently and instantly manage any increase in load by distributing it among various servers. There is no risk of hardware failure. The Random-Access Memory (RAM), Central Processing Unit (CPU) and storage are scalable and this is done automatically.
Cloud servers are very reliable. In case of any breakdown, the workload of the failed node is assigned to one of the other nodes automatically. This ensures zero downtime and maximum network uptime for the application.

C. Microservices

Microservices architecture involves splitting up the functionalities of any application into smaller modules that are independent of each other. It is defined in [4] as "the minimal independent process that interacts via messaging", and the microservice architecture as “a distributed application where all its modules are microservices”. This enhances the productivity and speed of the system. These loosely coupled modules are independent of each other. Failure in a certain component of the system will not affect the functioning of the other constituents of the system. Creating duplicate modules performing the same functions can be avoided by deploying this architecture. This greatly increases the efficiency of the system.

D. Serverless architecture

One of the major constraints in developing any application is hosting the system on a server. Parameters such as RAM size, CPU speed and server architecture must be decided in advance. However, these parameters are highly dependent on the number of hits received by the website. Serverless architecture is a model in which the parameters are predicted by the cloud and depending on the concurrent users, the cloud dynamically assigns servers to the growing or falling demand. The greatest benefit of this model is that the servers need not to be managed by the developers. No server is physically owned by the developers and this makes it cost-efficient. Developers can make more applications in lesser time. Computing po
server resources are saved.

E. NoSQL

NoSQL caters to the limitations of traditional databases. It is used to store unstructured data. The use of NoSQL has also contributed to faster storage and retrieval of data. Data-intensive applications require a cost-effective and easier way to store the data. This is provided by NoSQL databases. NoSQL makes use of a flexible scale-out architecture instead of a monolithic architecture. It can handle large volumes of unstructured data.

F. Arm processor

The hardware that is used must perform the basic function of receiving the National Marine Electronics Association (NMEA) string and processing it to extract the position coordinates. The ARM processors have higher abilities than usual microcontrollers. They are less expensive as than most other processors. The overall power consumption of an ARM processor is much lesser than an ordinary microcontroller. The devices with ARM processor can have a better battery life than other processors. They perform one operation at a time and hence are reliable and hence faster.

G. Real-time operating system

A Real-Time Operating System (RTOS) processes real-time data as an when it comes. RTOS is used to schedule entering tasks simultaneously. RTOS provides task scheduling and multitasking and this ensures proper flow and event response in the application. RTOS has a shorter Interrupt Service Routine (ISR) and deterministic behaviour that ensures that the task is handled in a given period of time. RTOS also manages the hardware resources thus introducing scalability in task completion.

H. NMEA

NMEA strings are the most common sources of GPS coordinate information. However, there are many NMEA messages that are received and the latitude and longitude values need to be extracted from the message by using certain conditions [5]. Some NMEA sentences include, $GPGSA, $GPGGA, $GPRMC sentences. Of these various sentences the location coordinate data are in the $GPRMC and $GPGGA strings. Only the $GPRMC sentences are considered. To extract the location data, regular expressions were used. This process is called data filtering. The $GPRMC part of the sentence is searched for from among the various other sentences obtained using regular expressions and this contains the data we need.

II. EXISTING SYSTEM

A. Existing GPS tracking mechanism

In the existing systems, two microcontrollers like the combination of the Arduino UNO R3 and the Raspberry Pi Model B is used [6]. Many other processors such as PIC32, PIC18F4550, STC895C2 are commonly used in other vehicle tracking systems [7-10]. The GSM module SIM900, a SIM card and GPS receivers are the common hardware components that are used to receive the GPS coordinates and send notifications or messages along the network [11-15].

B. *Communication protocols*

Many of the existing systems focus on improving the hardware in their system. They use communication protocols such as Hyper Text Transfer Protocol (HTTP) to transfer the data packets across their network [16]. The information is transferred and displayed on the website using an Application Programming Interface (API) gateway. HTTP is an application level protocol and it makes use of the client-server model for delivering messages. There are many drawbacks of using HTTP. The information that is sent through the HTTP protocol is not encrypted and this means that the privacy of the user is threatened. The packet headers in HTTP are larger than the packet headers of other protocols. This provides security and quality assurance of the information being transferred at the cost of large overhead. Any information that is transferred through this protocol can expose the device to virtual threats. Information including the Internet Protocol (IP) address and the Operating System (OS) running on the device is available to anyone. The protocol is reliable although it is slower than most other protocols. This makes it less suitable for real-time applications such a GPS tracking system.

C. *Servers for websites*

The websites that are implemented in the existing GPS systems mainly used static servers [17]. These static servers use predefined parameters such as RAM size, Read Only Memory (ROM) size and CPU speed. These parameters are fixed and cannot be changed after the server is bought or rented. In case of failure of certain hardware resources on the server side then the application comes to a halt. This reduces the availability of the system. The system is not scalable and the overall performance of the system is reduced.

D. *Power analysis*

The power analysis on most of the existing systems are done by altering the hardware components used [9]. Thus, the system does not try to improve power consumption by altering the software side of the application. Low power is achieved here by changing the clock frequency and turning off the GPS module whenever it is not being used. The GPS module uses cold start mechanism and therefore takes a longer time to initialize and it makes the system slower. This however slows the system down because it takes longer for the GPS to initialize after it is turned off every time. This system compromises on the speed and performance of the application while achieving low power.

III. PROPOSED SYSTEM

The key aspects that were tackled through this work comprise real-time tracking, a low power system, automatic theft notification through the use of geofences which is internet free and a cloud platform to ensure secure and enhanced services.

A. *System Implementation*

The microcontroller that is used is an Ultra-low-power STM32L4 Series MCUs based on Arm® Cortex®-M4 core with 1 Mbyte of Flash memory and 128 Kbytes of Static RAM (SRAM), in LQFP100 package [18]. The hardware system implementation was built on the *B-L475E-IOT01A* development board. The board has an inbuilt Inventek ISM43362-M3G-L44 Wi-Fi module in it. The microcontroller used in the proposed system is operated in 48MHz clock and it also has Low Power UART (LPUART). Other hardware components that were used include a GPS module, a Wireless Fidelity (Wi-Fi) module, an internet dongle, a power adaptor and an antenna.

The implemented prototype initialises the internet connectivity system. The Wi-Fi module is initialised and the specified Service Set Identifier (SSID) is connected via the FreeRTOS. The initialisation process also includes connecting the board to the cloud by uniquely identifying through the unique ARN number, and verifying the keys and certificates.

The GPS module once received a fetch would start storing the NMEA string character in the Direct Memory Access (DMA) buffer. This data is cleaned up and sent to the cloud for further processing. Processing at the edge is reduced as it would enhance the battery life of the system. The received NMEA string is parsed and the unprocessed coordinates are sent to the cloud. The process flow of the system is shown in *Fig 1.*

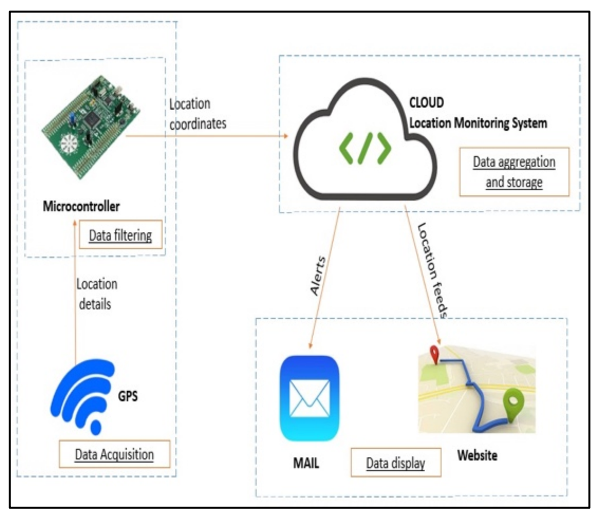

Fig.1 Block Diagram of the proposed system

The LPUART is used to obtain the data from the GPS module facilitating low power. In addition to that the microcontroller is put to sleep for 10 seconds between every data acquisition routine.

The microservices technology with serverless architecture is used in this system. After the Wi-Fi module publishes the latitude and longitude details on a Message Queuing Telemetry

Transport (MQTT) topic, the message is sent to the broker. The broker forwards all these messages on any topic to all the clients that are subscribed to that topic.

As seen in *Fig 2.* The IoT service platform triggers the microservice-1. The microservice-1 processes the received data from the microcontroller and checks whether the location value is valid or not and it adds the date and time to the coordinate values. It then stores the data in a NoSQL database-1. Any change in database-1 triggers microservice-2, which will compare with the previously received pair of coordinates and conclude on violation of the geofence. This result will be stored in NoSQL database-2. A change in database-2 will trigger microservice-3 and this goes on to send the respective email notification to the concerned user. All these internet-free processes require no user interactions.

For this system, a password-protected website is designed using Hypertext Mark-up Language (HTML) and JavaScript (JS). This website calls an API using Asynchronous JavaScript and AJAX XML to trigger a program called microservice-4. Microservice-4 gets the location from database-1 and feeds it to the website. The website now plots the coordinates using google maps API and then plots the route [17,19]. This website is hosted in an Apache-Tomcat server. This map automatically updates every 2 seconds without refreshing the entire page. This is done without disturbing the user interaction with the website. The system also enables the user to view the history of routes. The user can choose to enable watch mode and disable email notifications if the user is moving out of the geofence with their knowledge.

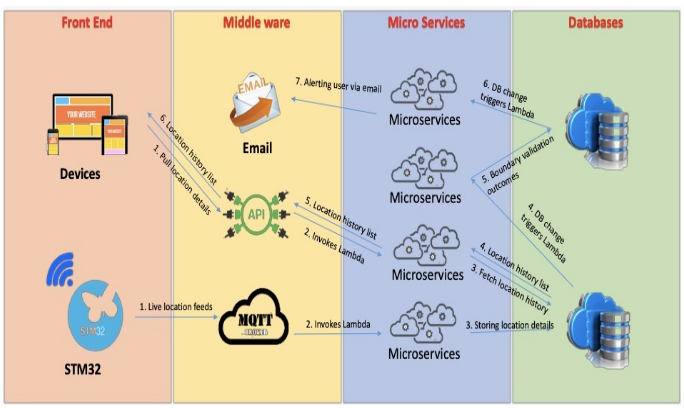

Fig.2 Flow Diagram of the proposed system

B. FreeRTOS

The proposed system uses the *'FreeRTOS'*. It is an operating system for microcontroller that enables it to be scheduled effectively and function in real time, it is specifically used in IOT based devices enabling them to be connected to the cloud securely. It is free of cost and can be used with existing Hardware Abstraction Layer (HAL) libraries of the microcontroller.

C. MQTT communication protocol

The proposed system uses the MQTT application layer protocol to communicate with the cloud. It is a lightweight machine to machine protocol extensively used in IoT based applications, hence can function in low bandwidth environments effectively. It has relatively less space and time complexity aiding the low power feature of the proposed system. It also facilitates the Quality Of Service (QoS) feature that would let us choose the degree of reliability and throughput.

D. Serverless

The proposed system uses serverless modern architecture as the GPS tracking system is quite lightweight and does not demand the requirement of a whole stand-alone server. Serverless architecture hence poses to be a less expensive alternative for the proposed application in the aspects of maintenance and operation. The GPS tracking system requires a reliable architecture to avoid loss of data since route would become misleading and inaccurate in case of server breakdown. Also, since the tracking system is vulnerable to potential expansion in terms of computation and storage in the cloud, having a serverless architecture would prove to be a better alternative.

E. Microservice

The proposed system uses microservice based software development technique. Since the proposed GPS tracking system has multiple modules such as the data filtering, storage, aggregation, notification alerts to name a few, having modularity in the server-side program would prove to reduce plenty of overheads. The core advantage of the modularity is during break down. Consistency demanding systems like GPS tracking is suggested to function using microservices since, in case of modularity break down, separate parts of the system would still be functional. Hence enabling easier and quick debugging.

F. Low power

The proposed system achieves low power extensively by the use of sleep modes in the microcontroller and various other methods including, low power UART, reducing computation at the edge and using MQTT protocol. In addition, the GPS module is also set to standby mode to reduce the power usage.

IV. RESULTS AND DISCUSSION

The existing system is incompetent and is capable of potential upgrades. Most of the existing GPS tracking systems use static stand-alone servers which are expensive to operate and maintain [17]. They are prone to break down and hence loss of performance and data. Breakdown of the stand-alone server hardware would freeze the whole tracking system, hence making it dis-functional. Lack of modularity in the server-side program organisation also poses to be a severe issue as breakdown of small part would corrupt the whole system. GPS tracking system demands scalability as the data is burgeoning, most of the existing systems do not address this aspect. GPS tracking system would be used in various internet bandwidth environments hence the existing systems that demand a lot of bandwidth would be inefficient and misleading due to loss of data in areas of low bandwidth [16]. Power consumption is one of the major concerns of the GPS tracking system. GPS tracking has to function seamlessly without the requirement to refill the power source as it would be generally fixed in remote and less accessed areas [9]. The GPS system has to be secure with the coordinate data as it could lead to vulnerability of safety. The advantages of the proposed system over the existing system include,

A. Low Power

The proposed system is proven to work with reduced power consumption from *Fig 3.* For a hardware implemented with the specifications as mentioned earlier the system would run on 2 Alkaline AA LR6 batteries for 2 months, 1 day and 8 hours. This would ease the setup of power supply from the side of the end user. There are 3 run phases and 1 sleep phase during initialisation and subsequently 2 run phases and 1 sleep phase in the ratio of 1:10 respectively, providing greater power saving. The proposed system also puts the GPS module to standby and gives it a warm start during the run phase.

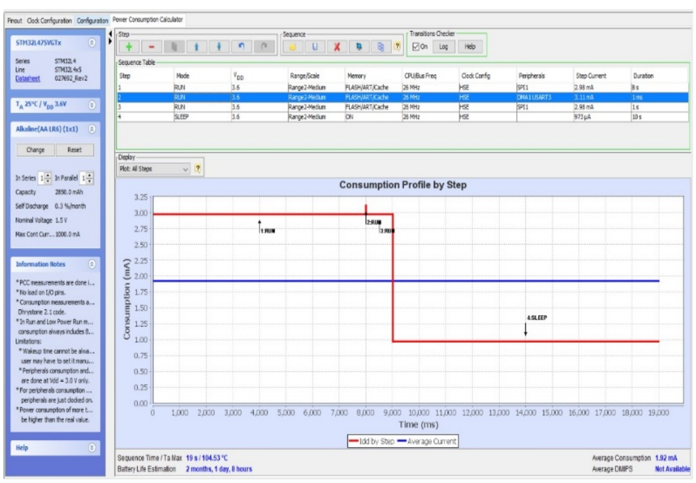

Fig.3 Low power analysis simulation.

B. Reliable

The proposed system is proven to be reliable in the aspects of being break down free unlike most of the existing systems. The aspect of modularity enables reliability as breakdown or damage of a single module would not affect the overall functionality as it does in the existing system. The proposed system would work in areas of low bandwidth too, hence unlike the existing system there would be no loss of information due to insufficient bandwidth. Quality of Service feature of the MQTT protocol enables us to decide on the level of required reliability.

C. Scalable

The serverless architecture in the proposed system unlike the existing static server-based systems can be expanded on demand dynamically in both performance and storage aspects.

D. Secure

The proposed system has various security aspects included in it as a part of the RTOS. Various certificates and keys are used to abstract the cloud from the hardware layer and provide security. Hence spam data cannot be inputted as well as the data in the database cannot be accessed without proper permissions.

E. Improved Performance

The proposed system has certainly improved performance over the existing system due to the scheduling efficiency of RTOS. The user can easily configure the SSID and the MQTT specifications including the topic, ARN of the cloud, QoS etc.

V. CONCLUSIONS

An augmented and effective low power, real-time GPS tracking and automatic theft notification system has been implemented as shown in *Fig 4.* by the use of modern architecture and ARM core processor. However, further changes can be introduced in this field.

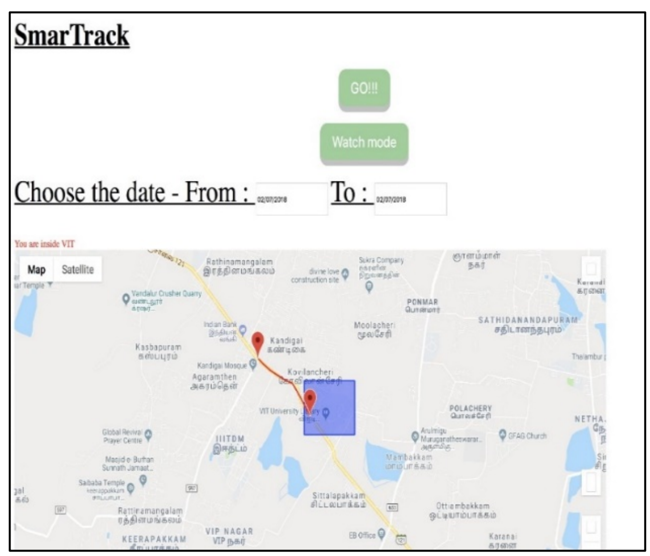

Fig.4 Propose system implementation

VI. REFERENCES

[1] M. Dey, M. A. Arif and M. A. Mahmud, "Anti-theft protection of vehicle by GSM & GPS with fingerprint verification," *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Cox's Bazar, 2017, pp. 916-920.

[2] M. M. Rahman, J. R. Mou, K. Tara and M. I. Sarkar, "Real-time Google map and Arduino based vehicle tracking system," *2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, Rajshahi, 2016, pp. 1-4.

[3] M. Geetha, T. Priyadarshini, B. Sangeetha and S. Sanjana, "Anti-theft and tracking mechanism for vehicles using GSM and GPS," *2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM)*, Chennai, 2017, pp. 252-255.

[4] Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina. *"Microservices: yesterday, today, and tomorrow. "*2016.

[5] P. V. Mistary and R. H. Chile, "Real time Vehicle tracking system based on ARM7 GPS and GSM technology," *2015 Annual IEEE India Conference (INDICON)*, New Delhi, 2015, pp. 1-6.

[6] K. Shruthi, P. Ramaprasad, R. Ray, M. A. Naik and S. Pansari, "Design of an anti-theft vehicle tracking system with a smartphone application," *2015 International Conference on Information Processing (ICIP)*, Pune, 2015, pp. 755-760.

[7] N. Kiruthiga, L. latha, S. Thangasamy, "Real Time Biometrics based Vehicle Security System with GPS and GSM Technology", *Procedia Computer Science*, Volume 47, 2015, pp. 471-479, 2015.

[8] Branko M. Vukelic, Rade Stancic, Stevica G. Graovac "Microcontroller Based Implementation of an *Integrated Navigation System for Ground Vehicles ", 1st IFAC Workshop on Advances in Control and Automation Theory for Transportation Applications September 16-17, 2013. Istanbul, Turkey*, Volume 46, Issue 25, 2013, pp 139-144, 2013.

[9] M. S. Amin, M. A. S. Bhuiyan, M. B. I. Reaz and S. S. Nasir, "GPS and Map matching based vehicle accident detection system," *2013 IEEE Student Conference on Research and Developement*, Putrajaya, 2013, pp. 520-523.

[10] W. Aiping, "The Design of Anti-theft Device for Vehicle Based on GSM," *2017 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, Changsha, 2017, pp. 384-386.

[11] M. S. Uddin, M. M. Ahmed, J. B. Alam and M. Islam, "Smart anti-theft vehicle tracking system for Bangladesh based on Internet of Things," *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, Dhaka, 2017, pp. 624-628.

[12] M. A. Fera, R. Aswini, M. Santhiya, K. R. G. Deepa and M. Thangaprabha, "HEAL-health monitoring in emergency vehicles with their authentication by RFID and location tracking by GPS," *2015 Seventh International Conference on Advanced Computing (ICoAC)*, Chennai, 2015, pp. 1-6.

[13] M. T. Kamisan, A. A. Aziz, W. R. W. Ahmad and N. Khairudin, "UiTM campus bus tracking system using Arduino based and smartphone application," *2017 IEEE 15th Student Conference on Research and Development (SCOReD)*, Putrajaya, 2017, pp. 137-141.

[14] N. J. Gor and A. Jeyakumar, "Voice controlled motorized wheelchair with real time location monitoring," *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Chennai, 2017, pp. 1163-1167.

[15] P. Jyothi and G. Harish, "Design and implementation of real time vehicle monitoring, tracking and controlling system," *2016 International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, 2016, pp. 1-4.

[16] N. Mangla, G. Sivananda, A. Kashyap and Vinutha, "A GPS-GSM predicated vehicle tracking system, monitored in a mobile app based on Google Maps," *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Chennai, 2017, pp. 2916-2919.

[17] N. S. Sehwani, S. R. Sangle and Y. N. Vadhavkar, "Real time automobile tracking system with an automated security algorithm," *2017 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, 2017, pp. 2125-2129.

[18] ST microelectronics, *"Discovery kit for IoT node, multi-channel communication with STM32L4"*, B-L475E-IOT01A datasheet, March. 2018.

[19] T. Monawar, S. B. Mahmud and A. Hira, "Anti-theft vehicle tracking and regaining system with automatic police notifying using Haversine formula," *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, Dhaka, 2017, pp. 775-779.