

# Automation in supply chain management system using Internet of Things (IoT)

Aishwarya Raj Laxmi <sup>1</sup>, Ayaskanta Mishra <sup>2\*</sup>

<sup>1</sup>M.Tech Scholar, School of Electronics Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha, India

<sup>2</sup>Assistant Professor, School of Electronics Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha, India

\*Corresponding author E-mail: [ayaskanta.mishrafet@kiit.ac.in](mailto:ayaskanta.mishrafet@kiit.ac.in)

## Abstract

An automated Supply Chain Management (SCM) System includes the real-time GPS position tracking of transport fleets as well as a RFID based shipment tracking at the entry and exit point of the warehouse. This paper has proposed a system architecture and implementation of the same using Arduino and SIM808 based GPS position tracking system for constrained power application. A Raspberry Pi and RC522 RFID Reader is used for a MQTT based shipment tracking system at the warehouse. Integration of both the modules makes a comprehensive Automated Supply Chain Management System, which can be used by logistic companies to monitor and manage their logistics using the web based GUI. Internet of Things (IoT) platform is used for data communication from Sensors to cloud platforms HTTP POST and PHP is prior and CloudMQTT in later case.

**Keywords:** Supply Chain Management (SCM); IoT; Raspberry Pi; Arduino; HTTP; MQTT; RFID; IEEE 802.11 WLAN; GPS; SIM808.

## 1. Introduction

In this modern era, for any industry to be successful, there is a very important role of technology. Since the market is volatile so it becomes important to understand the customer base. As the forecasting of the demand by the customers and also fulfilling it has become complex so without proper coordination between various business partner it can be a tedious job. This has pushed the companies to address this gap through various software applications and utilizing the eminent data packet to meet the demand of smart world [1]. In order to address this challenge a review has been done to evaluate the feasibility of IoT in the area of supply chain management (SCM). In any business process before reaching the customer each commodity has to cross multiple business partners from manufacturer, suppliers, and distribution centers to retailers. After all this it reaches the customers. So it is vital for the business management system to provide the visibility of the product in each stage of supply chain to have right information about the identity, location and other information related to tracking of the object.

It is important for each company to be aware of, where the product is located physically within the supply chain by maintaining a delicate relation between the middle man and customer and also to keep a balance on cost and time. This study helped us to bring the need of supply chain management (SCM) in transportation [2].

IoT is a network of physical objects to exchange data with the manufacturer or the user used for integration, sensing and communication [3,4,5]. In IoT the word "things" or embedded devices are interconnected through public or private network [6]. It also refers to the unique id used for sensing, monitoring, self configuration for the implication of various applications influencing the business process for proper interaction between digital and real world [7-9]. With its successful applications in various domains, it also steps into smart logistic management and data analytics. We

see that, data management plays an important role for customer's satisfaction and at the same time maintain a balance between cost and time. This is done by RFID in place, which make the company aware of where is the product.

The main intention to develop a multimodality system is that it can be used for various applications. This system uses a wireless Raspberry Pi sensor. With this system all the sensor values are obtained. Arduino is more suitable for reduced power constraint application in IoT. Hence a comparative study is done as shown in Fig.1.

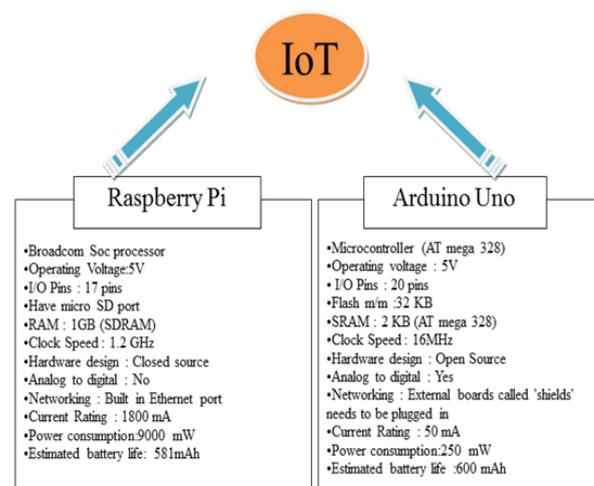


Fig. 1: Comparative Study between Raspberry Pi and Arduino UNO.

The comparative analysis of system specifications shows that Raspberry Pi is more suitable for Linux based MQTT application framework for RFID shipment tracking and Arduino will be for battery powered mobile real-time GPS position tracking module.

## 2. Methodology adopted

Supply Chain Management (SCM), is management of flow of goods and services that involves raw materials supply and storage, the process and the work for final goods from point of origin to point of consumption. Here all the entities are collaborated in order to enhance the overall benefits and satisfy the buyer's requests. The main aim of SCM is to increase product sales to end user and reducing both inventory and operating expenses [10]. So we can say that stock management, organization learning is important features of SCM [10]. It has been seen that E-Commerce is the electronic trade with goods and service. SCM is an efficient support for e-Commerce solutions.

Another module for our work is Radio frequency identification (RFID) which is used to track the tag attached to the object using electromagnetic field in order to store the information. The passive tag used here collects energy from readers near to it. RFID is important to solve many business needs like tracking pallets, indication of products [11]. A contactless communication using a valid reader device is enabled through a radio link by sending its corresponding unique ID [11], [12]. RFID tags are mounted on locomotives to identify origin, destination of the commodities of being carried. Hence RFID offers advantage over manual system by allowing more efficient inventory and tracking [11].

The main protocol used here is MQTT protocol. MQTT stand for message queue telemetry and transport. MQTT is a light weight application layer protocol using TCP/UDP in transport layer which is working upon IP network for packet switching application. MQTT also works on AWS cloud for bilateral communication between things and people [13]. But here MQTT protocol works on two types of controls: Publish and Subscribe. In MQTT terminology broker is the server for MQTT protocol. Here broker gets all published and subscribed control messages from publishing devices to forward it to subscribed devices. The MQTT publish command is issued using mosquitto broker in python for Raspberry Pi application [14]. Using two network technologies, MQTT protocol is implemented as per this work. The IEEE 802.11 is more complex suitable for high data-rate and longer range wireless communication, hence it is suitable for tracking at places where power requirement is more where as when then the GPS coordinates just has to be obtained and requirement of power is less, the work can be done by Base Station Trans-receiver (BTS) sending the data to cloud and to front end.

Using the above modules the overall architecture has been planned such that, a vehicle having goods, when moves between the source and destination the GPS coordinates can be tracked to know the proper location of device at particular time stamp for transparency between the businessman and customers. The data accumulated get transmitted over IEEE 802.11 network and with the help of GPS tracker made to move to cloud for particular application. The overall system architecture has been shown in Fig.2.

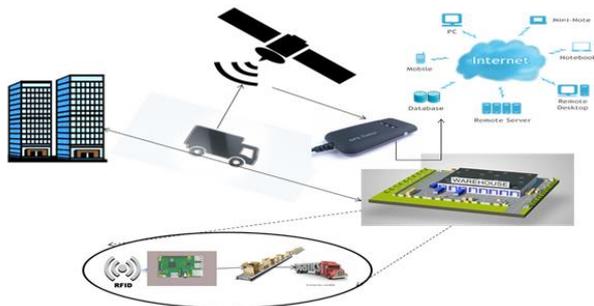


Fig. 2: System Architecture for Shipment Tracking.

### 2.1. Module for position tracking

We know in any business that involves dealer, supplier customer, there should be transparency maintained while transporting the goods so that there is effective utilization of time and total costing.

This can be done by keeping in track of the service of goods provided. Hence position tracking comes into picture. But tracking of the GPS values of a moving object requires high computational power which in turn can be costly for each member in the supply chain, so for proper utilization of money, time there can be implementation of IoT. Basically here when the sim808 module pins are connected to Arduino transmitter and receiver pin, an inter-connection between the two is built or setup for implementation to get the GPS coordinates. The SIM800 module is GPS and GSM module to transmit voice, information with low power consumption. The Arduino board is used as it requires low computational power and battery life is more. It is cost effective as tracking is done only for GPS coordinates. The information is sent through BTS to cloud MQTT/HTTP. The information could be seen at front end GUI, which is interlinked with database as information can be stored and used in future also as in Fig.3.

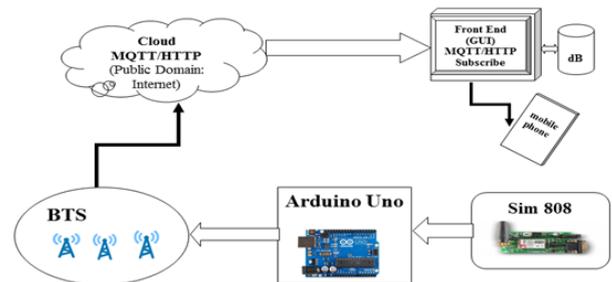


Fig. 3: Model for Position Tracking.

In order to get the GPS coordinates for a tracking system of a module using the Arduino Uno the basic algorithm has been explained in Fig.4.

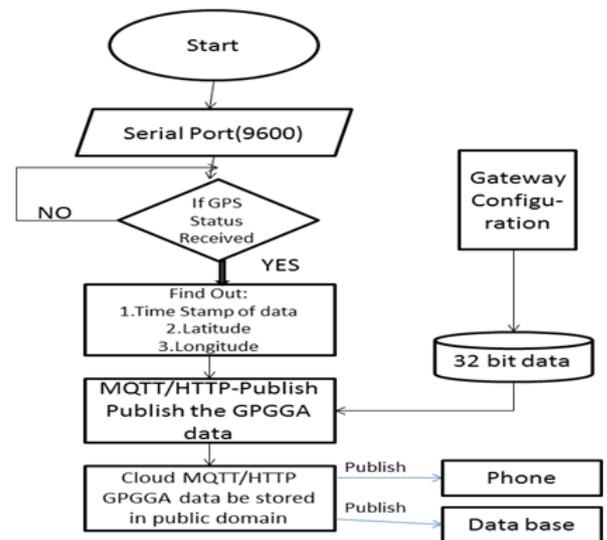


Fig. 4: Algorithm for Position Tracking.

Firstly the system is reset and made the software serial pins 6 & 7 to be the transmitter and receiver pin. Then the serial port has been set at 9600 to get the GPS status. If the GPS status is received the output is in a GPGGA format and from that we extract the time stamp, latitude, longitude of a given position at that point of time. It is a 32 bit address having Gateway configuration but if the status is not received the loop terminates. Once we get the data using MQTT/HTTP protocol the data is being published and sent to cloud MQTT/HTTP which is public domain internet. The data hence is published or made available in the front end or Graphical User Interface (GUI) and stored in mobile phones or database for further references. We see that in case of SIM 808, it doesn't support MQTT protocol so, seeing the prospect of this paper we have used HTTP instead. But if firmware update is done then the module SIM 808 will support MQTT protocol. Also the schematic diagram of basic connection is shown in Fig.5.

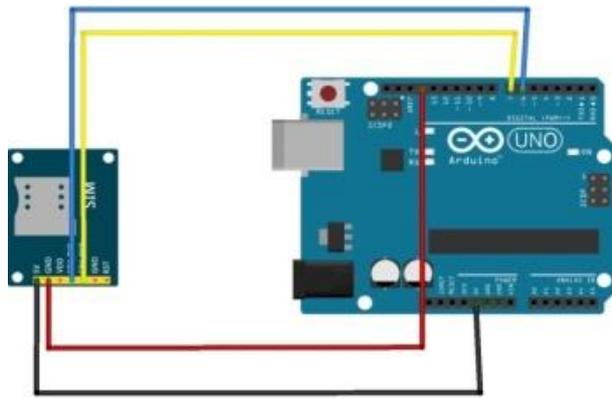


Fig. 5: Schematic Diagram of Modules for Position Tracking.

### 2.2. Module for shipment tracking

The proposed architecture has been designed for most of the real world IoT application with minor modification such that it suitable for generic IoT based application. Here python based gateway deice using Raspberry Pi module has been designed for the deployment of proposed architecture such that when the user sends the command, the module can be subscribed to the same topic and information can be published as the response of the topic and that command can be fetched by the gateway device. Here a high transmission power and high data rate is required so IEEE802.11 WLAN network becomes a good solution for the backbone of mentioned IoT deployment scenario [15]. Tracking of goods is important for maintaining transparency in the supply chain among its organizer, middle man and customer. In shipment tracking, the tracking is done in the warehouse during shipment in and out of goods having particular shipment and scanner id respectively. Also with the calculation of average transit time the transparency record can be maintained. The basic model of shipment tracking for reference has been explained in Fig.6.

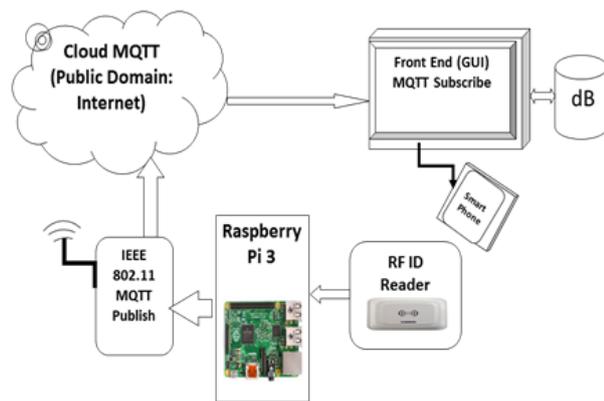


Fig. 6: Model for Shipment Tracking.

In order to read the information from the tag attached to the product being scanned RFID reader is used and that information is sent to Raspberry Pi module where the main processing occurs. Then the data is sent to MQTT published unit which is IEEE 802.11 WLAN network which publishes the information to MQTT broker that in turn receive all messages and filter them to be published to all subscribed clients such that the information could be seen in the front end GUI interlinked with database so that the information can be stored and used in future.

```
Mosquitto_pub -h cloudmqtt.com -p 1883 -u aishwarya -P xyz123 -m "shipment tracking" -t UID (goods)
```

For the basic algorithm explained as in Fig.7 it starts with initialization of SPI protocol. When the RFID reader reads the UID or Unique ID of each good being scanned the RFID tag information is obtained and read by SPI. If the unique ID obtained is present in the database then there is updating in the memory and the shipment is done and dispatched further such that the information can be stored in the database but if in case the UID is not present so it has to be added to the memory and there can be tracking of the shipment and the information is being stored in the database. After the above process the information is being analyzed in the data analytic system which reads all the protocols and analyses the information in such a way that it can be presented in front end GUI. Finally after all this process the goods are scanned over the conveyer belt and loaded to reach the desired destination. If the process at any point is hampered again this system goes back to reading the SPI protocol and process again continues.

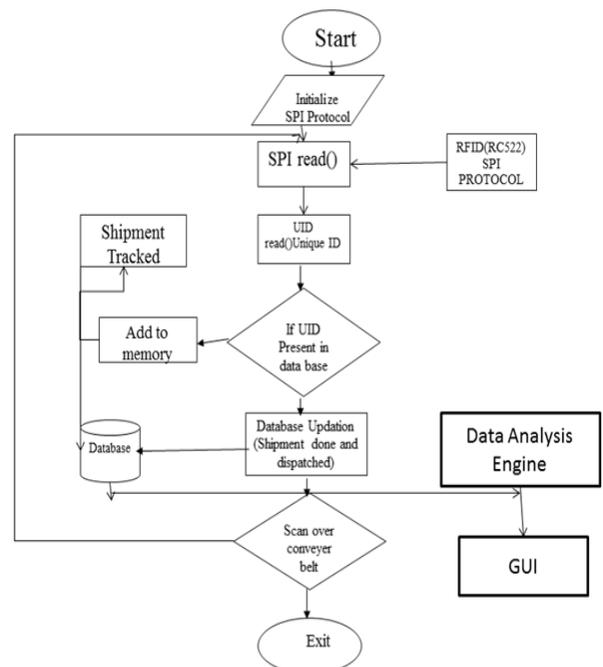


Fig. 7: Algorithm for Shipment Tracking.

For shipment tracking the MQTT client published/subscribed to MQTT broker using the Raspberry Pi has been explained in Fig 8.

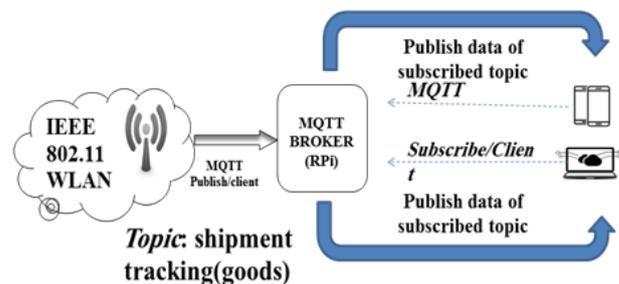


Fig. 8: MQTT Protocol MQTT\_Publish and MQTT\_Subscribe for Shipment Tracking.

An IEEE 802.11 WLAN network is used for the MQTT application broker/client architecture for availability of the data to the front end or GUI. In MQTT terminology the server for MQTT protocol is broker or is the channel between the two modules such that data can be published and subscribed simultaneously. The topic here is the tracking of goods such that the subscribed client will get the data immediately when then data is published. There can be multiple clients subscribing the same topic so that they can get a copy of received data from the MQTT broker. The process continues both way and tracking could be done easily. The Protocol Data Unit (PDU) encapsulation of MQTT Application data based on TCP/IP over an IEEE 802.11 network is shown in Fig.9.

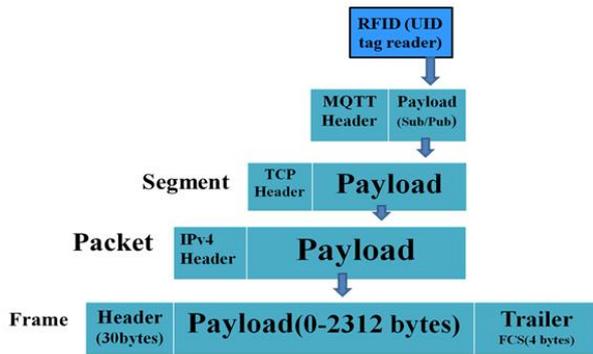


Fig. 9: Encapsulation of MQTT Protocol Using IEEE 802.11 Frame Format.

Fig.10 shows the basic schematic diagram for the connection between Raspberry Pi module and RC522 RFID reader done for shipment tracking in this paper.

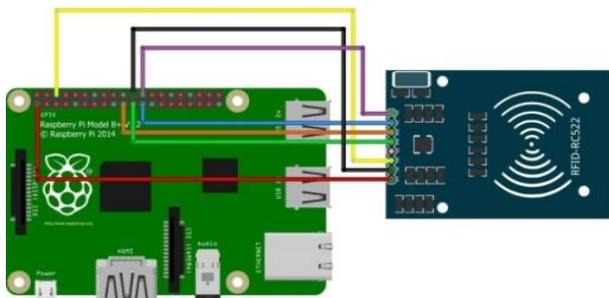


Fig. 10: Schematic Diagram of Modules for Shipment Tracking.

### 3. Results and discussion

The scope of the paper includes an implementation of proposed architecture as discussed in prior chapters.

#### 3.1. Position tracking SIM808 arduino UNO using google map API

The first module is the position tracking module for real-time GPS coordinate tracking of logistics. Fig.11 shows the prototype of GPS position tracking module implemented using a SIM808 module interfaced with Arduino UNO. The implementation of position tracking system is designed to give real-time GPS coordinates of the supply fleet, truck or any logistic on road or sea. The position tracking module is battery powered, thus a low power Arduino is proposed to implement the same to extend the battery life.



Fig. 11: GPS Position Tracking Module Using SIM808 and Arduino UNO Powered by MI 10000 mAh Power Bank.

#### 3.1.1. Process for filtering and extraction of GPS coordinates from NMEA format to required format using SIM808 AT command

\$GPGGA format of NMEA data is obtained using AT+CGPSOUT=2.

```
AT+CGPSPWR=1 /* Power on GPS
AT+CGPSOUT=2 /* GPGGA Format
```

Table.1 shows the detail \$GPGGA format of NMEA GPS data where the relevant data can be extracted using a data structure (in this case an array []) using C++ using the Arduino programming. Point to be noted, from the context of this work, we require the first three fields from the whole set of data available in \$GPGGA format namely time, latitude & longitude. The complete \$GPGGA format data is temporarily stored in a string and using the looping structure we extract the first three fields from its range of the whole string and store them into three separate variables. These separate variables are then pushed to HTTP server using the process mentioned in the chapter 3.1.2.

Table 1: Global Positioning System Fix Data

Name	Example Data	Description
Sentence Identifier	\$GPGGA	Global Positioning System Fix Data
Time	170834	17:08:34 Z
Latitude	4124.8963, N	41d 24.8963' N or 41d 24' 54" N
Longitude	08151.6838, W	81d 51.6838' W or 81d 51' 41" W
Fix Quality:		
- 0 = Invalid		
- 1 = GPS fix	1	Data is from a GPS fix
- 2 = DGPS fix		
Number of Satellites	05	5 Satellites are in view
Horizontal Dilution of Precision (HDOP)	1.5	Relative accuracy of horizontal position
Altitude	280.2, M	280.2 meters above mean sea level
Height of geo id above WGS84 ellipsoid	-34.0, M	-34.0 meters
Time since last DGPS update	blank	No last update
DGPS reference station id	blank	No station id
Checksum	*75	Used by program to check for transmission errors

#### 3.1.2. Process for sending data to HTTP server using AT command

Further to maximize the battery life the GPS coordinates are sent to the cloud on demand basis. This is achieved by implementing an algorithm in Arduino to detect and SMS received from a particular number, preferably the authorized company number to request for a Latitude, Longitude information. The algorithm also support a time based update of latitude and longitude information to the cloud using HTTP POST message by

```
AT+CIPSTART="TYPE", "domain", "port"
/* Stat a TCP Connection
AT+CIPSEND /*Send GPS data to Server
AT+CIPCLOSE /* Connection Termination
```

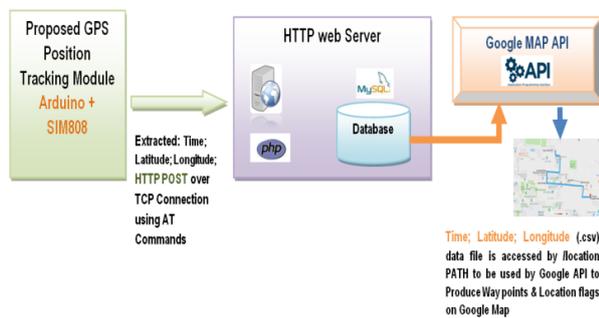
The data in the following format can be stored in .csv file at the server end. The data sent from SIM808 module is passed to a PHP Server. The php script will store the data as .CSV (Comma Separated Values) file. The latitude and longitude values are in degrees and decimal minutes (DMM): e.g. 41 24.2028, 2 10.4418 as per Google Map API.

Device\_ID, Timestamp, latitude, longitude

The .CSV file having the data can be passed to Google map API to show the coordinates (time, lat, long) on the Google Map. This can be visualized on any web based platform (Mobile, Desktop).

**3.1.3. Processes involved in the integration of RAW GPS data to Google API**

The GPS RAW data time, latitude, longitude is sent to a HTTP server using AT command from the Arduino and SIM808 based proposed position tracking module. The data is stored in the database of the server. The most simplified approach as proposed in this work is .csv file. The database implementation can also be achieved using a MySQL. The RAW GPS coordinate data along with its time stamp is then captured by calling a Google API using a handler using PHP scripting in the HTTP server running PHP at the backend as suggested in Fig.12.



**Fig. 12:** Process Involved in the Integration of RAW GPS Data to Google API.

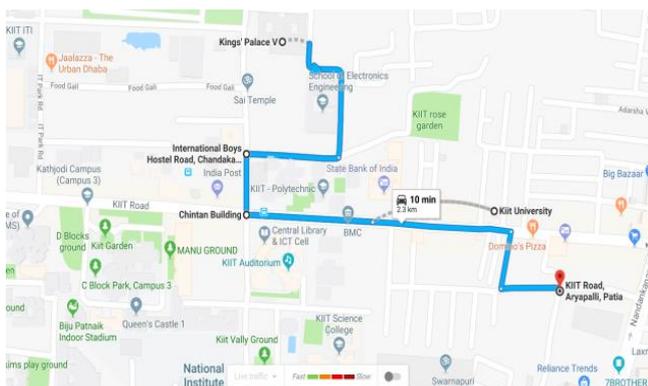
The Google API would fetch the data from .csv file as proposed and generate the vector and show it on the Google MAP for visualization on any GUI platform. The Google API can be programmed to show various desired output as positions vs. time or waypoints for tracking the positions of the desired logistics.

Table.2 shows the database (.csv) showing fields as discussed above. The latitude and longitudes stored in file can be passed to Google Map API.

**Table 2:** Position Database (.CSV) in Cloud

Device_ID	Time(GMT)	Latitude	Longitude
Truck-1	051333	44.96804	-94.42037
Ship-34	105311	44.33328	-89.13200
Truck-23	210827	33.75578	-116.35999
Ship-36	175634	33.84484	-116.54911

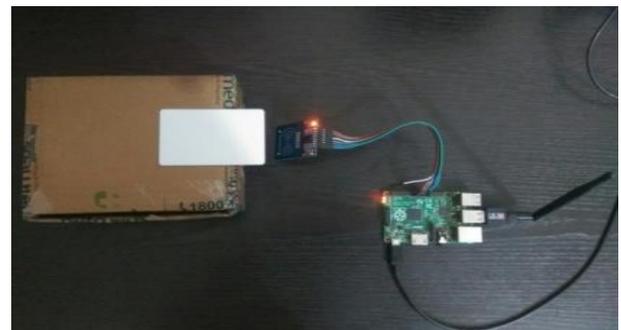
Fig.13 shows the way-point of a particular logistic for different time with different location. With the GUI the company can monitor the movement of its logistic in ease for better supply chain management.



**Fig. 13:** Google MAP View of GPS Coordinates for Different Time Way-Points for A Particular Device\_ID(Truck-1).

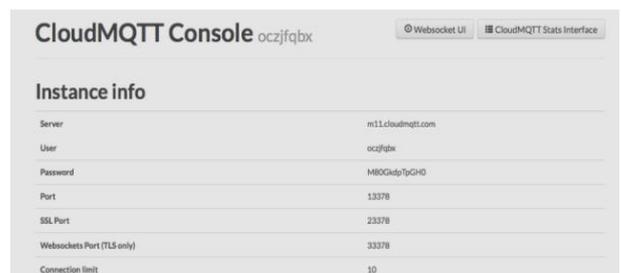
**3.2. RFID RC522 based shipment tracking with raspberry Pi using MQTT protocol**

The proposed second module for shipment tracking is an integral part of smart supply chain management system. The module includes RC522 RFID reader using the SPI protocol interfaced with Raspberry Pi model. The prototype of shipment tracking module is shown in Fig.14. Here the Raspberry Pi is equipped with IEEE802.11 WLAN adapter for connection to the cloud MQTT broker. In this case, there is integration of python code with mosquito MQTT client that is being operated on Raspberry Pi. In order to extract the UID of the passive RF tag which is attached to the shipment, python code is used. The RC522 RFID reader is being used to read the card and pass the UID and other information related to shipment to Raspberry Pi module. System call () method is done using python code and publishing the UID and time span to the cloud MQTT broker using mosquito MQTT client command is done.



**Fig. 14:** Shipment Tracking Module Using Raspberry Pi &RFID RC522.

Cloud MQTT console and instant information is shown in Fig.15 with the help of which user authentication, server address, port number can be received. The Raspberry Pi can publish the UID and time span to the cloud MQTT broker. Since the cloud MQTT broker is global and can be accessed from anywhere across the globe using simple internet connection, it is one of the major reasons or advantage of using it.



**Fig.15:** CloudMQTT Console and Instant Information.

A simple cross platform (Linux, Window) mosquito MQTT client can be used to retrieve the data from the cloud MQTT broker which can be installed at logistic company end. RFID topic can be subscribed by mosquito MQTT client and that topic is created in the cloud MQTT instant. The RFID tag information which are published to the broker are forwarded to the company MQTT client, and at the company end the data can be stored in a database. In this paper in order to get critical information for the automatic data generated various analytic tools has been used by this MQTT based.

The MQTT based RFID reader can be implemented at the entry and exist point of the warehouse by the logistic companies. When a shipment passes the entry gate or exist gate the passive RFID tags attached with the shipment reads the UID of that shipment that can be scanned by the RC522 RFID reader and can be send to cloud MQTT broker using MQTT published. The central monitoring system of the logistic company is connected to cloud MQTT broker as a subscribed client of particular topic. Here the warehouse location can be used as the topic names. The database of the

monitoring centre of the logistic company gets updated with what all shipment enter or exist a particular warehouse location using MQTT published and subscribed messages via cloud MQTT broker.

An example of set of automatic data is given in Table.3 that can be generated using different analytics tools to extract more mission critical information for real time shipment tracking.

**Table 3:** Automated Real-Time Data Available for Smart Logistic Management Using Real-Time Shipment Tracking

Parameters	Values
Average time in transit from Warehouse 'A' to 'B':	3 Days 2 hrs. 23 Minutes
Availability of TYPE-X shipment at Warehouse 'A':	150 Numbers
Shortest transit path for TYPE-X shipment:	Nearest next available Warehouse 'C' in terms of average transit time
Fast moving transit path:	Warehouse 'A' to 'C'
Most demand shipment type:	TYPE: E
Average waiting time in warehouse for a TYPE-Y shipment	2 months 11 days 3 hrs.

Table.4 shows the tabulated data collected as the result of shipment tracking at a particular warehouse. With this automation tracking of the shipment at various warehouse location can be done by the logistic company without much human intervention which in turn can cut down the operational cost, reliability and availability of real time data about their logistic by the usage of RFID based shipment tracking system.

**Table 4:** Summary of Collected Data

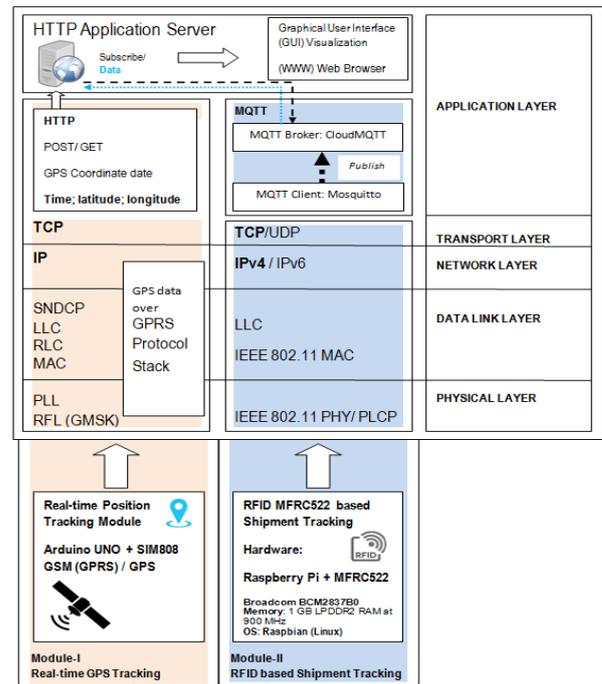
		RF ID Tracking					
Scanner ID	Shipment ID	Time Stamp (In)	Last time stamp	Time in transit	Time Stamp (Out)	Time in warehouse	
3DFF	7123	10:03:45	09:03:45	01:00:00	11:03:45	01:00:00	
3DFF	3239	11:56:45	10:56:45	01:00:00	22:56:45	11:00:00	
3DFF	8976	13:23:45	07:23:45	06:00:00	21:23:45	08:00:00	
3DFF	1975	14:45:45	8:45:45	06:00:00	20:45:45	06:00:00	
				Average time in Transit	3:45:00	07:00:00	

**3.3. Integration of position tracking & shipment tracking and implementation framework**

As discussed in detail in the prior chapters 3.1 and 3.2, this work has suggested two modules for low power real-time GPS position tracking and RFID based Shipment tracking using light weight MQTT protocol. Fig. 16 shows the proposed integration framework for both the modules. This integration strategy will help both the modules to be compatible to a common platform to harness the best of both the system and produce data and their visualization over a IP based network with the web based front end , which can be independent of any operating systems (OS).

The two proposed systems are designed with its own set of attributes. The real-time GPS tracking module is implemented mainly by keep the constrained power application for battery powered scenario. This module uses a low power Arduino UNO with a SIM808, which is scheduled on-demand to optimize the battery life of the tracking device. The second module is designed for a different scenario, i.e. keeping protocol overhead in mind. The RFID based shipment tracking module is designed using a Raspberry Pi based on System on Chip (SoC) unlike the first one, based on Arduino a Microcontroller firm factor. The Raspberry Pi powered module is significantly higher power but gives an added advantage of a complete Linux platform for complex application

support. The Raspberry Pi based RFID shipment tracking is designed by harnessing the design advantage of Linux OS. The MQTT mosquitto client is installed on the Raspberry Pi to send the RFID passive electronics tag information to the Cloud MQTT broker in internet. This RFID based shipment tracking module does not has any power constraint as it can be installed at the entry and exit point of any warehouse to track shipments and this is supposed to be a wall-outlet powered device. As both the module has different set of hardware, software and network protocols for implementation best suited for their individual applications, it is needless to say an integration framework is needed to combine both the module to work collectively to give more comprehensive set of data useful for both real-time positions tracking as well as RFID based shipment tracking. The end results will be more useful for the logistic companies to get a comprehensive tracking data on a single platform. This platform can produce graphical data for visualization and monitoring of logistics on a web based Graphical User interface (GUI) which can be platform independent.



**Fig. 16:** Integration Framework for Both the Proposed Modules.

Needless to say this integration framework will be advantageous as the output of the data can be obtained from any device such as a Windows based PC x86 x64, Linux, Macintosh or even mobile platforms like Android OS. This is possible by putting the data visualization front-end implemented over a HTTP platform, which can be rendered using any generic web browser.

**4. Conclusion**

The proposed system for automation in Supply Chain Management is implemented using prototyping development platforms like Arduino for SIM808 based position tracking and Raspberry Pi for RFID based shipment tracking. Both the proposed modules are integrated using the software framework to provide a comprehensive system for automation in SCM. Both the modules designed and developed from the scope of this paper are primarily for proof of concept. This design can be further taken forward to the next level of product development by implementing the same proposed system architecture using industry standard embedded systems and the cloud analytics platforms. The research work can be further expanded by developing a data analytics engine customized for SCM application.

## Acknowledgement

The prototypes are developed in Wireless Communication & Networking Lab, School of Electronics Engineering, KIIT Deemed to be University. The schematic diagram of Arduino/ Raspberry Pi is designed using Fritzing software. The software like python 3.0, php, Apache HTTP Web Server used in the scope of this work is as per their respective GNU General Public License agreements. Mosquitto MQTT client and Cloud MQTT hold their own individual registered license and this work acknowledge and comply with all their individual rights.

## References

- [1] S.Kalaivanan and S.Manoharan, "Monitoring and controlling of smart Homes using lot and low power Wireless Technology," Indian Journal of Science and Technology, Vol. 9, October 2016. <https://doi.org/10.17485/ijst/2016/v9i31/92701>.
- [2] V. Ganesan, U.S Lavanya, G. Maragatham, "A study of IoT in SCM and its nodes in multimodal business process", Indian Journal of Science and Technology, Vol. 9, ISSN 0974 -5645, June 2016.
- [3] Narayan Sharma, NirmanSingha, TanmoyDutta, "Smart bin implementation for smart cities", International Journal of Scientific & Engineering Research, Volume 6, November-2016.
- [4] Joon-SooJeong, Oakyoung Han, Yen-You You, "A design characteristics of smart healthcare system as the IoT application", Indian Journal of Science and Technology, Vol.9, ISSN 0974-5645, October 2016.
- [5] Amit Joshi, Gurpreet Singh, Gagandeep Singh, "Internet of Things: A Beginners' Précis and Future Scope", Indian Journal of Science and Technology, dec. 2016. ISSN 0974 -5645.
- [6] Atzori L., Iera A., Morabito G., "The internet of things: A survey", Computer Network, Elsevier, Volume 54, Issue 15, October 2010, Pages 2787-2805. <https://doi.org/10.1016/j.comnet.2010.05.010>.
- [7] Hyun- GiHong, "Analysis of business framework for Internet of Things", Indian Journal of Science and Technology, Vol.9, ISSN 0974 -5645 October 2016.
- [8] S. Chen, H. Xu, D. Liu, B. Hu and H. Wang, "A vision of IoT: applications, challenges, and opportunities with China perspective", in IEEE Internet of Things Journal, Vol. 1, no. 4, pp. 349-359, Aug. 2014.
- [9] GandeveBayuSatrya, HafuTasewReda, Kim Jin Woo, PhilipTobiantoDaely, Soo Young Shin and SeogChae, "IoT and Public Weather Data Based Monitoring & Control Software Development for Variable Color Temperature LED Street Lights," International Journal on Advanced Science, Engineering and Information Technology, Vol. 7, no. 2, pp 366-372, 2017. <https://doi.org/10.18517/ijaseit.7.2.1578>.
- [10] BurraKaruna Kumar, Srinath Narayanan, DegaNagaraju "Supply chain coordination models: A literature review", Indian Journal of Science and Technology, Vol. 9, ISSN 0974 -5645, October 2016.
- [11] C.Balasubramaniam, D.Manivannan, "IoT enabled air quality monitoring system (AQMS) using Raspberry Pi", Indian Journal of Science and Technology, Vol. 9, ISSN 0974 -5645, October 2016.
- [12] VS Reddy, MA Quadir, "Smart Live Tracking Garbage Collection using RFID", International Journal for Research in Applied Science & Engineering Technology, Volume5, October2017.
- [13] T. Manjula, U. Sreenivasulu, S.JaweedHussain, "A dynamic Raspberry Pi sense HAT multimodality alerting system by using AWS IoT", Indian Journal of Science and Technology, Vol. 9, ISSN 0974 -5645, October 2016.
- [14] Ding Yi, Fan Binwen, Kong Xiaoming and Ma Qianqian, "Design and implementation of mobile health monitoring system based on MQTT protocol," 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, 2016, pp. 1679-1682.
- [15] Ayaskanta Mishra "Design and deployment of MQTT based HeT-NeT using IEEE 802.15.4 and IEEE 802.11 for Internet of Things", International Journal for Research in Applied Science & Engineering Technology, Vol.5, ISSN: 2321-9653, November-2017, pp. 1616-1625.