Development of A Container GPS Tracking System To Enhance Supply Chain Security

Sanskar Kumar[1], Russel Dmello[2], Sakshi Kaveri[3], Shreyas Nanaware[4], Freda Carvalho[5]
[1]*Department of Electronics and Telecommunication Engineering*
[1]*The Bombay Salesian Society's, Don Bosco Institute Of Technology, University of Mumbai, Mumbai India*
[1]*sanskarkumar2017@gmail.com,* [2]*russeldmello88@gmail.com, sakshikaveri2002@gmail.com,*[4]*shreyasnanaware2003@gmail.com*
[5]*freda@gmail.com*

***Abstract*—This paper intends to provide an innovative GPS tracking and anti-theft system for container shipments, addressing the growing concern about theft and smuggling. The suggested system uses GSM, GPS modules, and an STM32F302R8 microcontroller for better durability and cost effectiveness compared to existing alternatives. A simple current break sensor detects theft at- tempts, triggering alerts sent to the server via the GSM module. Real-time location monitoring is enabled through NMEA message format packets broadcast from the GPS module to the server, facilitating swift intervention to prevent theft. The system ensures accurate track- ing and effective theft prevention. This approach has the potential to greatly improve security in shipping operations, providing concrete benefits to society by protecting valuable cargo from unlawful activity.**

***Index Terms*—STM32 microcontroller, GSM, GPS, NMEA, asset management, tracking system, real time monitoring**

I. Introduction

In the span of 2021 to 2022, supply networks in North America had a 13% increase in cargo theft volume, with a 16% increase in value, a trend that has persisted into 2023. While electronic products remain extremely susceptible, thieves are increasingly targeting domestic items and food and beverages, showing that increased pressures are driving shifts in criminal behavior. With a stolen product value of $107 million in 2022, a considerable rise from the $68 million recorded in 2020, cargo theft is definitely a concern that requires increased mon- itoring. Allianz Commercial, a worldwide maritime insurance, has observed an increase in cargo theft cases in recent years, notably in transportation and logistics. [1] Cargo theft cases increased more than 57% in 2023 compared to the previous year, according to CargoNet, a Verisk firm. According to CargoNet, about $130 million in products were stolen in 2023, but because reporting cargo theft is not mandated, the figure is likely to be greater. Cargo theft refers to the theft of products at any step during the shipping process, and instances have reached record proportions. [2] The major aim is to keep the containers and their contents safe throughout travel. The items will be transported safely, and the tracking technology will help to track the position in real-time. If the customers know the precise locations of their containers, there will definitely be a trust factor from the client side due to the reliable mode of business. The below Figure 1.1 shows the statistics of cargo thefts in year 2023.

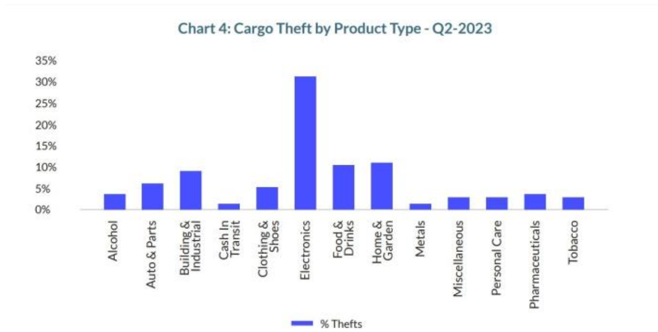


Fig. 1. Cargo Theft Statistics

A. Motivation

During the Covid-19 pandemic, when most modes of transportation were severely limited due to lockdowns and safety precautions, the movement of products grew more reliant on shipping containers, particularly for critical things like medical supplies, food, and other emergency goods. Shipping containers are really important for global trade, worth about $10 billion in 2024. But there are some significant challenges. One is ensuring that these containers are safe and secure. Imagine if someone stole or tampered with what's inside; that could be a major issue. So we're particularly working on keeping these containers safe and secure.That includes employing lock systems and technologies to ensure that everything stays where it is supposed to and is not tampered with

B. Scope of the Project

The project's goal is to develop a complete system capable of tracking several containers
simultaneously and individually. To achieve this, several key outcomes have been proposed.

- The project will include incorporating GPS devices into the tracking system.
  These devices will be mounted on each shipping container and will constantly collect location information. The acquired data will include latitude and longitude coordinates, allowing for accurate real-time information on the container's current location. This

connection will allow logistics managers and other stakeholders to monitor container movement throughout the shipping process, from origin to destination.

- GSM modules will add to GPS systems by allowing remote communication and data transmission over cellular networks.These modules will allow two-way communication between the tracking system and a central server or control center. They will allow the system to send alerts, receive commands, and transmit data, such as location updates and sensor readings, using standard mobile network infrastructure. This feature increases the system's versatility and allows for seamless connectivity, even in remote or inaccessible regions.
- In addition to GPS and GSM technology, the tracking system will incorporate sensor systems to enhance security and monitoring capabilities. The system includes a closed loop circuit system. It will continuously monitor the container's environment and detect any irregularities or unauthorized access attempts. When such events are detected, the system triggers an alert and notify specified persons or authorities, allowing for immediate response and intervention. By combining sensor systems, the tracking system improves theft prevention and cargo security while in transit.
- To ensure reliable and continuous functioning, the project intends on optimizing battery usage inside the tracking system. This optimization involves carrying out careful calculations to determine the power requirements of numerous components, such as GPS devices, GSM modules, microcontrollers. The sleep and wake cycles of the GSM module, and power-saving features of other components will be considered in battery optimization strategies.
- The tracking system will be connected to a database so we can see where something is going and check if someone tries to steal it.

The ultimate objective is to develop a viable GPS and GSM tracking system for shipping containers that provides continuous location visibility.

II. Methodology

The system begins by initializing the GPS module, which is necessary for the device to detect its accurate position. The initialization involves configuring the hardware connections and software settings to ensure proper operation. In addition, UART connection is established between the GPS module and the STM32 Microcontroller. UART is a common serial communication protocol used for transmitting data between devices. After establishing connectivity, the GPS module starts transmitting location data in NMEA. This data contains critical information such as latitude, longitude, altitude, time, and satellite status. The STM32 microcontroller then conducts the parsing process, carefully examining the NMEA words to extract relevant information, such as latitude and longitude coordinates. In addition, AT instructions are used to activate the GSM module. AT commands are standardized instructions for controlling and communicating with modems and other devices. Similar to the GPS setup, UART connection is established between the GSM module and the STM32 microcontroller. This allows data flow between the microcontroller and the GSM module. Once both the GPS and GSM modules are initialized and communication channels are established, the location data is transmitted to a mobile phone via the GSM module. This enables real-time tracking of the device's position. To ensure continuous and automated updates, the system is designed to repeat the data retrieval and transmission process hourly. This periodic procedure takes advantage of both GPS and GSM technologies to deliver efficient and dependable tracking functionality.

A. Software Used

Ansys HFSS (high-frequency structure Here is the list of softwares that have been utilized until this point of time :

1) STM32 Cube IDE : Designed for programming and developing applications for STM32 microcontrollers

2) Docklight : Docklight is a testing, analysis and simulation tool for serial communication protocols

3) KiCad : KiCad is an open-source electronic design automation (EDA) software suite for creating schematics and printed circuit boards (PCBs)

4) Putty : Putty is a free and open-source terminal emulator, serial console, and network file transfer application. It supports various network protocols like SSH, Telnet, rlogin, and raw socket connections.

5) Google Earth : Google Earth is a virtual globe, map, and geographical information program that allows users to explore the Earth's surface and view satellite imagery, maps, terrain, and 3D buildings. It enables users to navigate to different locations, view geographical features, landmarks, and even explore outer space. Google Earth is available as a web-based application, desktop software, and mobile app, providing an immersive and interactive way to discover and learn about our planet.

An important point to be noted is that all the above softwares used are open source and free of cost which means anyone can install it on their systems and work on it to develop applications free of cost.

B. Hardware Used

1) Quectel M66 GSM module: The M66 GSM module is a compact and versatile cellular communication module. It provides GSM/GPRS connectivity, allowing devices to communicate over cellular networks. The module features support for voice, SMS, data transmission, and internet connectivity, making it suitable for applications such as remote monitoring, telemetry, tracking, and control systems. With its small form factor and low power consumption, the M66 module is widely used in embedded systems where reliable and efficient cellular communication is required.

2) NEO 6M GPS module : The NEO-6M GPS module is a compact and cost-effective global positioning system (GPS) module commonly used in navigation and tracking applications. It utilizes the NEO-6M GPS chipset, which provides accurate positioning information with support for multiple satellite navigation systems, including GPS, GLONASS, and Galileo. The module communicates with external devices such as microcontrollers or computers through a serial in- terface, typically UART (Universal Asynchronous Receiver-Transmitter), and provides real-time latitude, longitude, altitude, time, and satellite data.

3) STM32F302R8 mircrcontroller : The STM32F302R8 microcontroller is a member of the STM32F3 series of 32-bit ARM Cortex-M4 microcontrollers produced by STMicroelectronics. It features a high-performance core running at up to 72 MHz, with an extensive set of peripherals and interfaces, making it suitable for a wide range of embedded applications.

III. Implementation and Troubleshooting

To begin implementation, we initially tested the Quectel M66 GSM module, by linking its TX, RX, VCC i.e anywhere from 3.7 - 4.5V, and GND pins to a USB to TTL converter and then to the docklight. By adjusting the baud rate at 115200, selecting the suitable COM port and using AT commands, the GSM functionality was verified by acquiring the aˆOKaˆ response in return and also its operational integrity. Figure 4.1: AT commands response on Docklight Further, the GPS

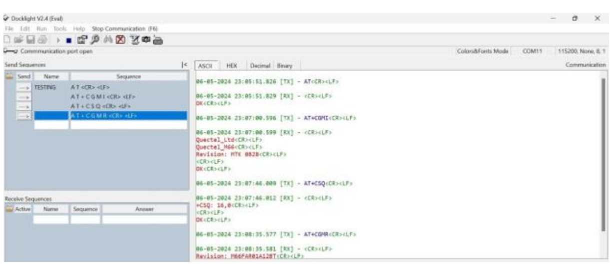
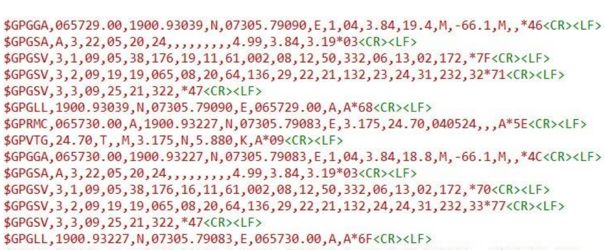


Fig. 2. AT commands response on Docklight

module was tested by linking its TX, RX, VCC i.e 3.3V, and GND pins to a USB to TTL converter, then to the docklight. By adjusting the baud rate at 9600 and selecting the suitable COM port, we observed NMEA sentences displayed on the docklight as illustrated below.



```
$GPGGA,065729.00,1900.93039,N,07305.79090,E,1,04,3.84,19.4,M,-66.1,M,,*46<CR><LF>
$GPGSA,A,3,22,05,20,24,,,,,,,,,4.99,3.84,3.19*03<CR><LF>
$GPGSV,3,1,09,05,38,176,19,11,61,002,08,12,50,332,06,13,02,172,*7F<CR><LF>
$GPGSV,3,2,09,19,19,065,08,20,64,136,29,22,21,132,23,24,31,232,32*71<CR><LF>
$GPGSV,3,3,09,25,21,322,*47<CR><LF>
$GPGLL,1900.93039,N,07305.79090,E,065729.00,A,A*68<CR><LF>
$GPRMC,065730.00,A,1900.93227,N,07305.79083,E,3.175,24.70,040524,,,A*5E<CR><LF>
$GPVTG,24.70,T,,M,3.175,N,5.880,K,A*09<CR><LF>
$GPGGA,065730.00,1900.93227,N,07305.79083,E,1,04,3.84,18.8,M,-66.1,M,,*4C<CR><LF>
$GPGSA,A,3,22,05,20,24,,,,,,,,,4.99,3.84,3.19*03<CR><LF>
$GPGSV,3,1,09,05,38,176,16,11,61,002,08,12,50,332,06,13,02,172,*70<CR><LF>
$GPGSV,3,2,09,19,19,065,08,20,64,136,29,22,21,132,24,24,31,232,33*77<CR><LF>
$GPGSV,3,3,09,25,21,322,*47<CR><LF>
$GPGLL,1900.93227,N,07305.79083,E,065730.00,A,A*6F<CR><LF>
$GPRMC,065731.00,A,1900.93327,N,07305.79080,E,3.441,7.73,040524,,,A*6D<CR><LF>
```

Fig. 3. NMEA sentences recorded on Docklight

Example of GPGGA GPS sentence:-
$GPGGA, 161229.487, 3723.2475, N, 12158.3416, W, 1, 07, 1.0, 9.0, M, , , , 0000*18

161229.487: Time of the fix in UTC.
3723.2475, N: Latitude of the fix (in degrees and minutes, with N indicating North).
12158.3416, W: Longitude of the fix (in degrees and minutes, with W indicating West).
1: Quality of the fix (1 = GPS fix).
07: Number of satellites being tracked.
1.0: Horizontal Dilution of Precision (HDOP).
9.0: Altitude of the fix in meters.
M: Units of altitude (meters).
(Empty fields): Additional information (may be empty).
0000: Differential GPS data age.
*18: Checksum for data integrity.

Example of GPRMC GPS sentence:-
$GPRMC, 161229.487, A, 3723.2475, N, 12158.3416, W, 0.13, 309.62, 120598, , *10

161229.487: Time of fix in UTC hours, minutes, and seconds.
A: Status A = active or V = void. This tells if the GPS has a valid fix at the moment.
3723.2475, N: Latitude 37 degrees 23.2475 minutes North.
12158.3416, W: Longitude 121 degrees 58.3416 minutes West.
0.13: Speed over the ground in knots.
309.62: Course over ground in degrees true.
120598: Date - 12th May 1998.
,: Magnetic variation direction (not provided in this sentence).
10: Checksum, used for error checking the data.

A ring buffer maintains continuous data flow by storing elements in a fixed-size buffer and overwriting the oldest ones when the buffer is full. Thus, the ring buffer was developed to verify UART communication, guaranteeing alignment between transmitted and received data. This establishes a groundwork for incorporating GPS functionality using this dependable communication method.

Sensing System : The STM32 board was utilized to generate an alert that displayed a message in response to a voltage drop at the RX pin. To demonstrate a prototype in which if no value is present at a pin, an alert message should be presented on the console at that time. The two pins were connected using jumper wires wherein one pin was providing the power supply to the next pin. As in when the connection of jumper wires was removed the output was displayed on the console. To use this sensing system prototype, the merging with the GSM module was done so that when the closed loop circuit used for sensing breaks the alert message will be received on the specified mobile phone whose number will be put in the program.

GSM+STM32 : Initially, the STM32F302R8 board was

Fig. 4. Verifying UART establishment on Docklight

selected using STM32CubeIDE.

Following the micrcontroller parameters were configured and it starts with establishing clock settings and GPIO pins, which sets up the hardware environment.The baud rate for GSM is 115200. The USART3 peripheral is initialized to allow UART connectivity, which is required for dealing with external modules such as GSM devices. Before entering the main loop, the system sends a sequence of AT instructions over UART to set up the GSM module for SMS transmission. These instructions contain fundamental AT commands for setup and configuration, like "AT" for checking if the GSM module responds, ``AT+CREG?''- provides
information about the registration status and access technology of the serving cell, and
``AT+CMGF=1''- this command sets the GSM modem in SMS Text Mode.

After startup, the system constantly checks the status of a jumper pin attached to a GPIO pin (PA1). If the jumper pin is identified in a low condition (indicating a door breach), the system generates an SMS message notifying a breach attempt. The `alertMsg[]` array stores the message "Alert: Door breach detected". To send the SMS message, the system first creates the ``AT+CMGS'' (the command sends an SMS message to a GSM phone) command and specifies the recipient's phone number. This command is formatted as a string and sent to the GSM module over UART. The system then waits for the ``>'' prompt, which indicates that the GSM module is prepared to accept the SMS message payload. Once the ``>'' prompt is
received, the system sends the message content contained in the `alertMsg[]` array. Finally, the system transmits the CTRL-Z character (ASCII code 0x1A) to mark the conclu- sion of the message. Delays are introduced throughout the process to guarantee correct timing for command execution and message transmission.

GPS+STM32: For this process also the initialization was done for a system based on a microcontroller, most likely an STM32F3xx series device by setting up the required configurations in the STM32Cube IDE. The baud rate for GPS is 9600. Specifically, it sets up the USART1 peripheral for UART connection with the GPS module. The system then enters an indefinite loop, always waiting for and processing GPS data.

Within the loop, the system uses custom methods to handle GPS data receiving and decoding. It uses the `Wait_for` function to wait for certain GPS data phrases, such as ``GGA'' and ``RMC''. When the system detects these phrases, it replicates the incoming data into the ``GGA'' and ``RMC'' character arrays. These words usually include important GPS information like position, time, and velocity. The line `Copy_upto("*", RMC/GGA);` likely invokes a custom function called `Copy_upto` with two arguments: a character pointer "" and a destination character array. The purpose of this function call appears to be to copy characters from a source buffer up to a specified delimiter (in this case, the asterisk character) into the destination array.

Once the data has been received and copied, the system will decode it using specialized
decoding routines such as `decodeGGA` and `decodeRMC`. These routines parse GPS data from
character arrays and extract important information into a structured manner that is saved in the `gpsData` structure. The `decodeGGA` function retrieves GPS fix information from a `GGA` phrase, such as latitude, longitude, altitude, fix quality, time, and number of satellites. Similarly, the `decodeRMC` function pulls date, speed, course data, and RMC data validity from the `RMC`
statement.

After getting the coordinates from the GPS module the parsing was done and extraction of only certain sections of the NMEA format was carried out .The retrieved data must then be transferred to the GSM module for transmission.This ring buffer was created using a UART1 connection within the STM32Cube IDE environment. The ring buffer stores the coordinates received from the GPS module via Docklight-Testing, analysis and simulation tool for serial communication protocols.The algorithm for reading GPS data and writing it to the buffer was included into the loop structure, guaranteeing that it runs continuously as long as fresh data is recognized.As data comes, it is stored in the buffer before being passed to the next level of processing.

IV. Troubleshooting

The project encountered its first obstacle when the UART Tx and Rx pins of the STM32 controller malfunctioned, necessitating a switch to a different UART port due to the multiple UARTs available on the controller. Moreover, upon each restart, the GSM module defaulted back to its default baud rate of 9600, despite being previously configured to a different rate. Indoors, the GPS module struggled to obtain a fix on the antenna, requiring placement in an open area. Additionally, there was a challenge in converting the latitude and longitude coordinates obtained from the GPS module. Compatibility issues arose between the GSM module and the high-frequency clock of the STM32 microcontroller. Despite the GPS module's slightly inaccurate location data, which deviated slightly from the actual location but still met project requirements, a USB to TTL converter was employed to troubleshoot communication errors, with analysis performed using Docklight software.

V. Results

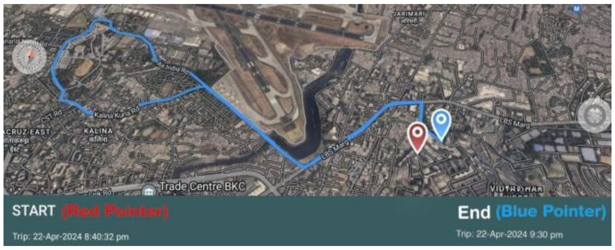


Fig. 5. Route

In Figure 5, Google Earth was used to map out the entire route, allowing for subsequent verification of the GPS coordinates' accuracy. The testing began at Holy Cross Church, highlighted in red, and concluded at the same endpoint, now marked in blue. The route included passing through the LBS area which is a high traffic zone, crossing the Mithi River to assess GPS reliability, and traveling alongside restricted zones such as Mumbai International Airport and Kalina Military Camp. Upon returning, the journey also takes place through Kalina's urban and residential areas, following the same route, crossing the Mithi River again, and navigating the congested LBS area.

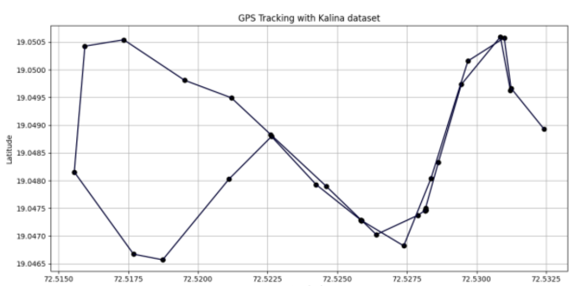


Fig. 6. Graph

Figure 6 Plotting those coordinates in Google Colab using Python and the Matplotlib module, a graph was obtained, mirroring their route, confirming GPS accuracy. Comparing their tracked path to Google Maps, a high level of consistency was found, validating the system's accuracy in tracing their journey from point A to point B.

Figure 7 shows the accuracy of the GPS module.

In the precision test of the GPS module at Don Bosco College Playground, a 5-meter path was chosen for measure- ment using Google Earth. Walking this path with the GPS

Fig. 7. Measurement of path

module produced latitude-longitude coordinates, which when analyzed on Google Earth, showed a traveled distance of roughly 18 meters, significantly more than the expected 5 meters. Despite this divergence, the GPS module consistently generated distinct coordinates for each point along the path.

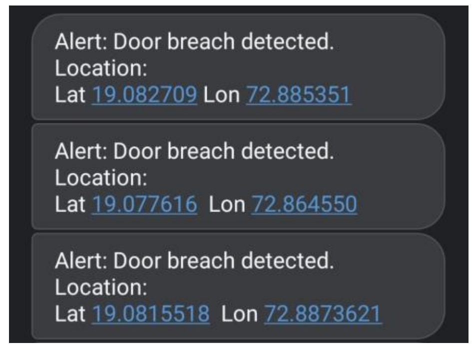


Fig. 8. Alert SMS

Further, our tracking system is incorporated with an alert sensing mechanism to promptly alert us of any intrusion during container tracking. Upon detection, the system sends real-time alert messages containing latitude and longitude coordinates to the vehicle's driver for immediate response, while simultaneously logging this information in the database so that the company which owns the cargo can conduct further inspection and take necessary actions. The above Figure 8 represents the alert SMS.

References

[1] Abdullahi Abdirahman, Abdirahman Hashi, Ubaid Dahir, Mohamed Elmi, and Octavio Rodriguez. Enhancing vehicle tracking through sms: A cost-effective approach integrating gps and gsm. *SSRG International Journal of Electronics and Communication Engineering*, 10(9), 2023.

[2] Andrea Miller. Cargo theft spiked over 57% in 2023 vs. 2022, new data shows, 2024. [Accessed: (April 29, 2024)].

[3] Rahul Khanna. Cargo theft: An increasing concern, 2023. [Accessed: (April 29, 2024)].

[4] Thin Thin Htwe and Kyaw Kyaw Hlaing. Arduino based tracking system using gps and gsm. *International Journal for Advance Research and Development*, 4(8):11–15, 2019.

[5] Mr Saif Iqlas, Mr Owais Ahmed, Mr Mohammed Ahmed Khan, and Shilpa Mangshetty. Vehicle tracking system and theft detection.

[6] Bilkisu Jimada-Ojuolape, Oluwafunto Spencer, and Monsurat Omolara Balogun. Development of a gsm based vehicle demobilizer and tracking system. *ABUAD Journal of Engineering Research and Development*, 6(2):167–175, 2023.

[7] Kamran Khan and Shuang Wang. Vehicle security system based on stm 32 micro-controller using gsm and gps module. *International Core Journal of Engineering*, 7(4):135–140, 2021.