Design and Development for a Vehicle Tracking System

Tim Abe P. Andutan, Rosanna C. Ucat
College of Engineering, University of Southeastern Philippines, Davao City, Philippines

Abstract—**In recent years, the drastic increase in the number of vehicle thefts brings about at an alarming rate around the world. However, existing vehicle tracking devices have certain limitations including the lack of ability to determine if the vehicle is on the right route. To address this problem, this study focused on the design and development of a vehicle tracking prototype with route detection, emergency button, and STATUS command to monitor the current location of the vehicle. Arduino Mega 2560, SIM900 Global System for Mobile communication (GSM) module, and NEO-6M Global Positioning System (GPS) module were used to develop the prototype. The GPS module, push buttons, and SMS command served as an input. The Arduino Mega 2560 was programmed using an algorithm to determine if the device deviated from its route, detect if the emergency button was pressed, and if STATUS command was received. The system sends a SMS if the vehicle deviated from its path, emergency button is pressed, and a STATUS command from the operator is received. Results showed that after several trials the prototype was successful in performing its functional objectives. The prototype was only limited to the use of a prototyping grade GPS module. The GPS used a built-in antenna and took time to connect to satellites. It is recommended to use an industrial grade GPS module and connect an external antenna to improve signal strength.**

Keywords—Arduino; global positioning system; GPS; global system for mobile communications; GSM; vehicle tracking; route deviation detection

I. INTRODUCTION

As the number of automobiles in the road increases, the crime involving larceny or vehicle stealing also increases [1]. In recent years, the drastic increase in the number of vehicle thefts brings about at an alarming rate around the world [2]. The Interpol Statistics revealed that 4.2 million vehicles were reported to be stolen in 2008 from 149 countries [3]. In 2020, the National Insurance Crime Bureau (NICB) further revealed that the vehicle theft increased considerably which totaled up to 873,080 compared in 2019 [4].

Evidently, the scenario is also common in the Philippines with a highest record of 13.2 cases of vehicle theft per 100,000 population in 2014 [5]. The volume of vehicle theft has brought to a national concern which ranks third next to theft and robbery among all property crimes in the country [6]. Recently, the vehicle theft in the country was at 4.6 cases per 100,000 population in 2018 [5].

The application of this study could be significant to corporations such as the banking sector, logistics company, and food industry in improving vehicle security during transportation of confidential materials, and accomplishing projects to avoid unwanted intrusions, delay, and theft. This study can also benefit the vehicle operators in terms of theft recovery and improvement of performance in vehicle security, tracking, and monitoring. This study can serve as a basis or foundation to future researchers in relation to vehicle monitoring and tracking system.

To address the growing concern in vehicle tracking, the proponent intended to design and develop an easy-to-use vehicle tracking and monitoring system with route deviation detection and SOS capability. The study also aimed to use the nature of Global System Monitoring – Short Message Service (GSM-SMS) Technology to effectively send notifications and location details of the vehicle when the vehicle deviated from its path, if the button was pressed during an emergency situation, and “<STATUS>” command was requested to determine the current status of the vehicle.

II. LITERATURE REVIEW

Khin and Oo [7] created a vehicle tracking system that utilizes GPS technology to determine a vehicle’s location. The study used GPS and GSM technology, in which the embedded GPS module retrieves the vehicle's location. After which, the GPS continuously monitor the movement of the vehicle and plot the location using Google Map and displayed on a webpage. The vehicle’s position was also saved through MySQL databased using XAMPP platform.

Besides, Sharp, Cable, and Burns [8] used GPS technology for tracking the movements of the visitors in heritage sites. The research utilized GPS to understand the temporal and spatial distribution of the visitors’ movements. The findings were used to develop strategic opportunities to attract the visitors to the less visited areas of the heritage site and properly engage with the site.

The study of Shukla [9] incorporated GPS technology for landmine detection wherein GPS was attached to a rover and the rover was controlled using Dual Tone Multiple Frequency (DTMF). A landmine detection circuit was installed to the rover. When a landmine is detected, the rover returned the location from the GPS and directly sent to the operator and locate the landmine.

The recent studies mentioned above used GPS technology for tracking the movements of a robot, tourist, and vehicles. However, none of the studies utilized GPS to track the movement of a vehicle that identifies whether object of interest to be tracked deviated from its route or not. The studies also were not able to determine if the object being tracked arrived to the specific location based on the coordinates from the GPS module.

The study of Reddy, Krishna, Chaitanya, and M. Neeharika [10] used GSM technology in developing a security alarm based on door knock patterns. Piezoelectric element was used to detect the door knocking pattern to detect possible intruder. When a correct knocking pattern is detected, the door unlocks, otherwise, the prototype sends a text message to the home owner indicating that possible intruder is present.

Sakthivel et al. [11] established the same concept, where they used a Vehicle Security System using embedded and GSM Technology. They designed a security system for automobiles by installing the device in the vehicle. Sakthivel et al. [11] created a password system before starting the host vehicle, alerting the owner of unauthorized persons attempt to drive the vehicle.

The concept about the use of GSM technology as a medium for sending a notification to the concerned personnel was used in the study. The GSM technology had provided various application in security and emergency purposes. Unlike the studies mentioned above, the study utilized the GSM technology to send an emergency alert notification whenever the SOS button is pressed. GSM was also used to monitor the current status of the vehicle when the operator sends a "<STATUS>" command, which were absent to those studies.

III. Research Methodology

A. *Research Design*

The research design used in the study was developmental research approach, which aimed to the design and development of both hardware and software components to produce a desired system that tracks the vehicle, check if the vehicle deviated from its path, and retrieve data from the GPS including the speed and the coordinates, button-trigger for emergency purposes. The system utilized the Global System for Mobile Communication (GSM) to transmit the data to a mobile phone. This study was non-experimental, since there were no manipulations of the variables.

Waterfall Method was used in this project; waterfall method is a sequential design process in which progress is seen as flowing straight downwards through a series of phases. Waterfall model was utilized since the proponents should complete each phase in order to proceed to the succeeding phase. Overlapping of the phases is not possible.

Fig. 1 showed the process started from the procurement of the materials. When the materials were already procured, the hardware, which includes the circuitry and the 3D printing were done. The programming phase under the software development commenced to test the functionality of the system based on the objectives of the study. When the prototype was already built, the proponents tested the prototype to various locations with the vehicle having the prototype attached to it. Data analysis was then carried out to discover the findings of the data obtained from the prototype sampling.

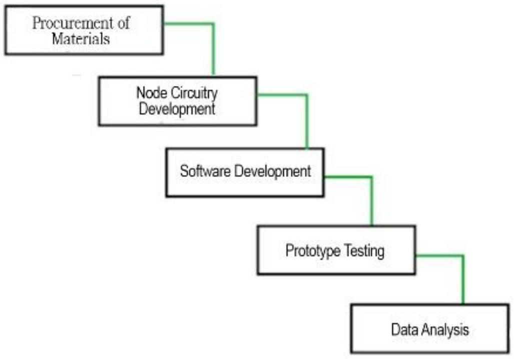


Fig. 1. Flow Diagram of the Study (Waterfall Method).

1) Data sampling: The proponent selected fourteen (14) distinct locations in Davao City. Four (4) locations were selected as the deviation route. While ten (10) locations were selected as the route. The prototype was put inside the vehicle and the vehicle travelled along the pre-determined route. The proponents then recorded if the prototype was able to record each location where the device traversed. Also, it was recorded if the prototype was able to detect the deviation route and if it successfully notified the user about the deviation. The SOS feature was also tested by pressing the SOS button repeatedly. Lastly, the "<STATUS>" command was also tested if the prototype was able to reply to the user the current location (coordinates), the current speed (kph), and the link to Google Maps.

The data for the deviation detection functionality was collected by recording whenever the coordinates from the GPS module were inside a specific area. If the coordinates were recorded in a specific area, the program then checked if that area was a deviation area. A success remark was recorded, which denotes that the prototype was able to identify the area correctly, and if successfully sent a SMS notification whenever the coordinates were inside the deviation area. Otherwise, no message was received.

The status command functionality was tested by giving ten alternating requests. Five requests for no status, and another five status requests. When the user sent a "<STATUS>" command, the prototype sent a SMS notification to the user. When no status request was sent, the prototype did not send any message. These two criteria were used to give a success remark. However, when the user sent a "<STATUS>" command to the prototype and no reply was received, or if the user did not request any status from the prototype but received an SMS, a failing remark was given.

Lastly, the same logic was applied to evaluate the SOS command. Whenever the driver pressed the emergency button on the device, the prototype sent an emergency notification on the dispatcher and a success remark was given. However, when the driver did not press the emergency button, and the prototype sent an emergency notification to the dispatcher, then, a failing remark was recorded.

B. Research Procedure

The research procedure focuses on the five major areas in the study, which included the materials used in the study, node circuitry development, software development, prototype testing, and data analysis. Procurement of materials discussed what materials were used in developing the study including material description and supplier. The node circuitry development discussed how the hardware counterpart of the prototype was built including the circuitry and the enclosure of the device, while the software development tackled the method on how the device was programmed. The prototype testing explained the method on which the data were collected. Lastly, the data analysis showed the tools to analyze the data and arrive with a comprehensive conclusion.

1) Materials: Prior to the development of the project, the following materials were procured, which included the Arduino Mega 2560 Rev 3, which served as the motherboard of the prototype. The SIM900 GSM module, was used to manage the SMS notification between the device and the receiver (driver and dispatcher). Ublox Neo 6M GPS module was used as GPS module that communicated to the satellites and acquired the coordinates. Stranded wires were used to manage the connection between the motherboard and other components, 5V 2A DC power supply to power the entire prototype and breadboard for testing the cricuitry. The Arduino Mega 2560 Rev 3, SIM900 GSM module, and Ublox Neo 6M were procured from Makerlabs Electronics in Santa Cruz, Manila through Lazada Philippines, while the remaining components were procured in Davao Times Trading Incorporated in Palma Gil Street, Davao City.

2) Node circuitry development: The node of the data-gathering transceiver was developed. The following materials were integrated to the Arduino Mega 2560 board, which processed the input from the sensors and transmitted the data gathered to a smartphone via GSM. For the casing, 3D printed enclosure was designed using SketchUp: 3D Design Software 2018. The enclosure was made of Acrylonitrile butadiene styrene (ABS) polymer.

Fig. 2 showed the top view of the 3D design of the enclosure, the object on the left was the cover while, the object on the right was the body of the enclosure where the electronic components were attached.

Fig. 3 showed the right view of the body wherein the name of the prototype was engraved. Fig. 4 is the left view of the enclosure, Fig. 4 showed the thickness of the cover.

On the other hand, Fig. 5 showed the schematic diagram of the prototype. The prototype used Arduino mega 2560 as its main processor where the SOS button is connected to the digital pin 2 and ground pin, the GSM module RX pin was connected to the TX 16 pin while the TX pin was connected to the RX 17 pin of the Arduino mega. The GPS module RX pin is connected to TX 18 while the TX pin was connected to RX 19. The GSM and the GPS used a 5V logic levels from the Arduino.

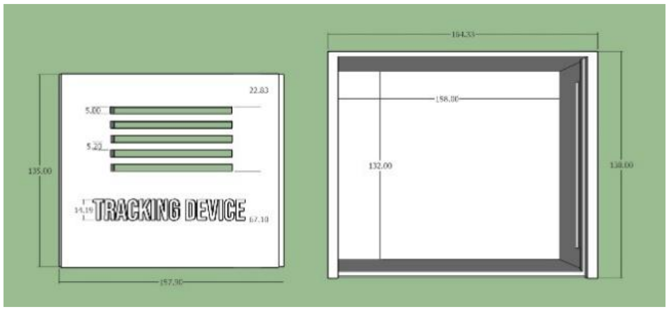


Fig. 2. Top View 3-dimensional Model.

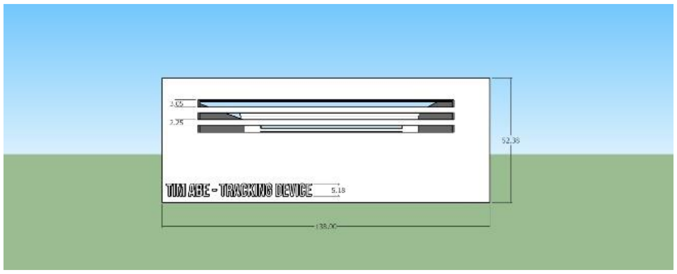


Fig. 3. Right View 3-dimensional Model.

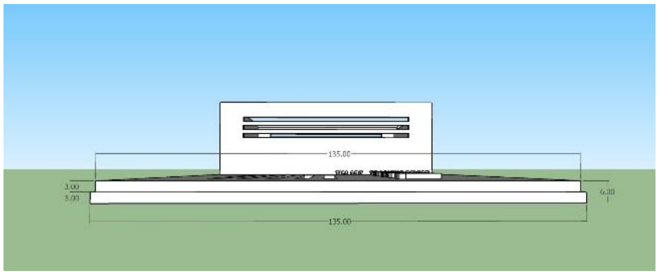


Fig. 4. Left View 3-dimensional Model.

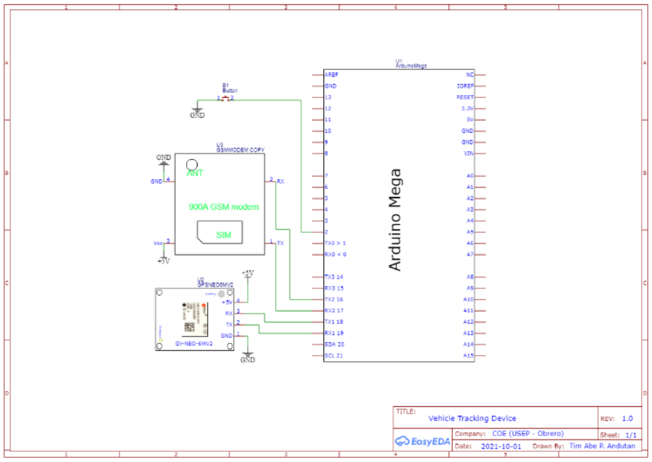


Fig. 5. Tracking Device Schematic Diagram.

Fig. 6 showed the list of all the components used in building the prototype and showed the placement of each component inside the 3D-printed ABS polymer enclosure.

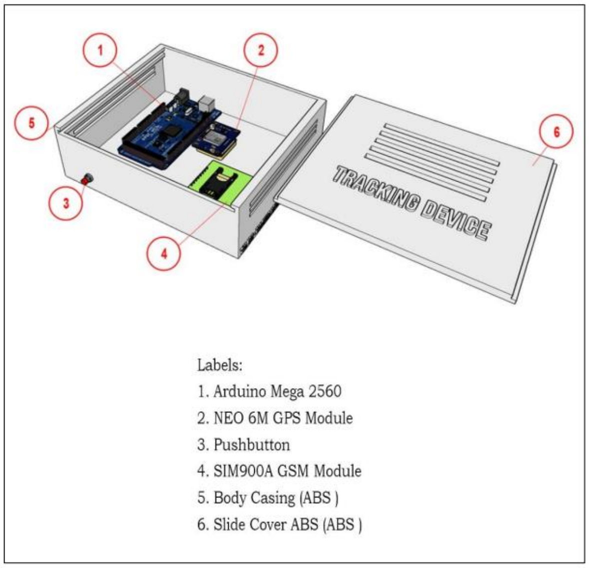


Fig. 6. Project Layout (Isometric Diagram).

3) Software development: The Arduino Mega board was programmed in Arduino Development Environment (Version 1.8.13) using C programming language. The Arduino Mega 2560 board was programmed to read the coordinates of a certain location and the speed (kph) on a real-time basis. The obtained coordinates were used to determine whether the coordinates were within the pre-determined route. These coordinates were used as a guide to determine if the current vehicle position deviated from its path. The coordinates of the pre-selected point were determined using Google Maps. If a deviation was recorded, the device automatically sent an alert message to the user. Moreover, the button served as an emergency button, in which it was programmed to send an emergency message to the user/owner when the button is pressed. On the other hand, the user can also send a message to determine the current status of the device by sending the keyword "<STATUS>." The device will then reply the following outputs: Current GPS coordinates, speed (kph), and a link directing to Google Maps to plot the location based on the coordinates received.

Fig. 7 showed the data flow diagram of how the prototype was programmed. It showed that when a signal was established for the GSM and GPS, the prototype started to read the coordinates and constantly checked whether the button is pressed, a status command was received, and if the coordinates is inside a specific preprogrammed location.

4) Prototype testing: Prior to the conduct of the data sampling, the prototype was tested initially. The prototype was tested within the residence of the proponent to check if the prototype was able to function accordingly. The code was revised several times to ensure the functionality of the whole system.

5) Data analysis: Prototype was analyzed and evaluated to the extent of the functionality of the whole system in terms of hardware and software development. The whole system was analyzed according to three parameters: (1) Able to detect if the vehicle deviated from its route, (2) able to send SOS notification when the SOS button was pressed, and (3) able to send the current status of the vehicle in terms of speed (kph) and the current location (coordinates). Observation was done for each major functionality and tabulated. Percent error was computed for each major function of the prototype. Percent error was used to quantify how close the number of successes to an estimated total value in which in this case, the number of trials, it also denotes the accuracy of the test based on the data collected. It approximated the error percentage of the data (experimental) relative to the expected standard value (theoretical).

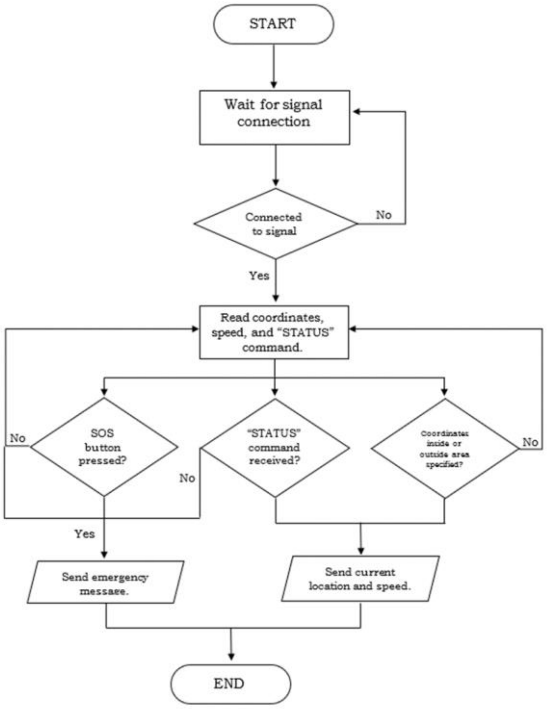
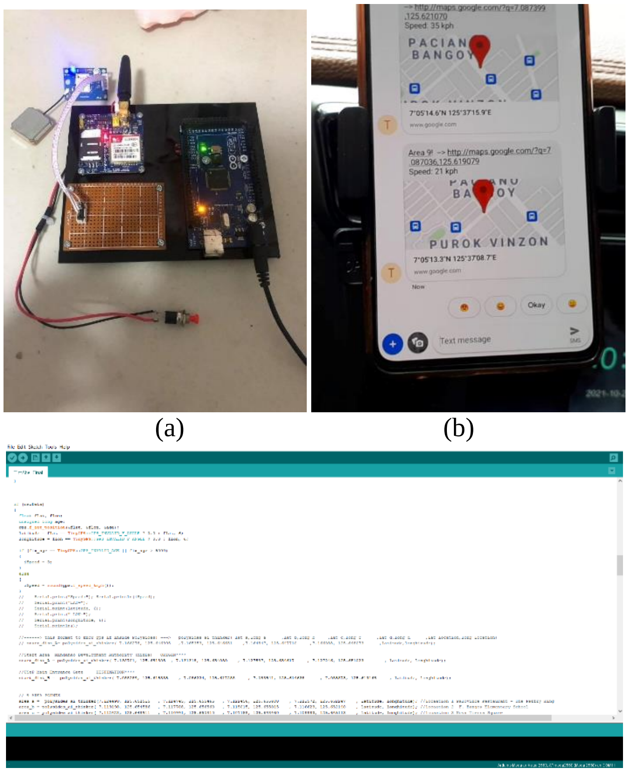

START
Wait for signal connection
Connected to signal
No
Yes
Read coordinates, speed, and "STATUS" command.
No
SOS button pressed?
"STATUS" command received?
Coordinates inside or outside area specified?
No
No
Yes
Send emergency message.
Send current location and speed.
END

Fig. 7. Software Programming Flow Diagram.

IV. Presentation, Analysis and Interpretation of Results

A. *Objective 1*

The prototype was composed of both the hardware and software. For the hardware counterpart, the node of the data gathering transceiver was successfully developed using the four major components, which included the Arduino Mega 2560 board which managed the data processing, GSM SIM900A module for data transmission from the device to the mobile phone of the user, and the NEO 6M GPS module which facilitated the tracking of the location and the speed of the vehicle. For the software counterpart, the program for the route deviation detection, emergency command, command, tracking, and sending of SMS notifications was coded using the Arduino Integrated Development Environment (IDE) using the simplified C programming for Arduino boards. Fig. 8 presented the integration of the hardware and software of the prototype.



-> http://maps.google.com/?q=7.087399,125.621070
Speed: 35 kph

7°05'14.6"N 125°37'15.9"E
www.google.com

Area 9! -> http://maps.google.com/?q=7.087036,125.619079
Speed: 21 kph

7°05'13.3"N 125°37'08.7"E
www.google.com

(a) (b)

(c)

Fig. 8. Hardware and Software Integration of the Prototype (a) Node Circuitry Transceiver, (b) Received SMS from the Transceiver, (c) Arduino Mega 2560 Programming using Arduino IDE.

B. *Objective 2*

The prototype was tested to track the vehicles' location and speed through the use of the GPS module. The GPS module communicated to the satellites and to retrieve the vehicles' location using the coordinates. When the coordinates recorded by the GPS were inside a certain area, the prototype then sent an SMS notification to the driver and the dispatcher to locate the vehicles' current location.

Table I showed the results for tracking and speed monitoring. The prototype was tested in 14 distinct locations in Davao City, each location, the prototype successfully sent the specified area, its coordinates, and the speed of the vehicle. The mobile phone also successfully received an SMS, which contained the following mentioned details and the Google maps link. Table I also affirmed that the prototype sent a notification to the user when it traversed the correct path. Among the fourteen test locations, ten (10) locations were programmed as a correct path, while four (4) locations were programmed as the deviated path. It is shown that the prototype was successful in detecting the deviated path.

Fig. 9 showed the map for the entire testing. The red dotted line showed the restriction area, in the program, when the coordinates recorded by the GPS were inside the red area, the prototype sent a SMS to the user indicating that the vehicle deviated from its path. Within the Northern area, industrial companies were present and some portions of the area were used as a commercial space. In the Western part, residential spaces occupied most of the area. Commercial establishments and schools were present. For the Eastern part, sea ports, beaches, and some commercial establishments were present. Lastly, for the Southern part, schools, commercial spaces, and residential areas were present.

TABLE I. RESULTS FOR LOCATION AND SPEED MONITORING

Area	Coordinates (Lat, Long)	Speed (kph)	Deviated? (Yes/No)	Received SMS? (Yes/No)	Description (Success/Fail)
1	7.124496,125.653671	38	No	Yes	Success
2	7.119050,125.654754	43	No	Yes	Success
3	7.111478,125.650077	24	No	Yes	Success
4	7.105888,125.645469	48	No	Yes	Success
5	7.099695,125.640449	34	No	Yes	Success
6	7.092534,125.634613	36	No	Yes	Success
7	7.085145,125.626663	42	No	Yes	Success
8	7.086773,125.621627	19	No	Yes	Success
9	7.087036,125.619079	21	No	Yes	Success
10	7.087199,125.616294	24	No	Yes	Success
11	7.082574,125.620010	36	Yes	Yes	Success
12	7.097366,125.614967	43	Yes	Yes	Success
13	7.132690,125.661613	31	Yes	Yes	Success
14	7.109219,125.649734	31	Yes	Yes	Success

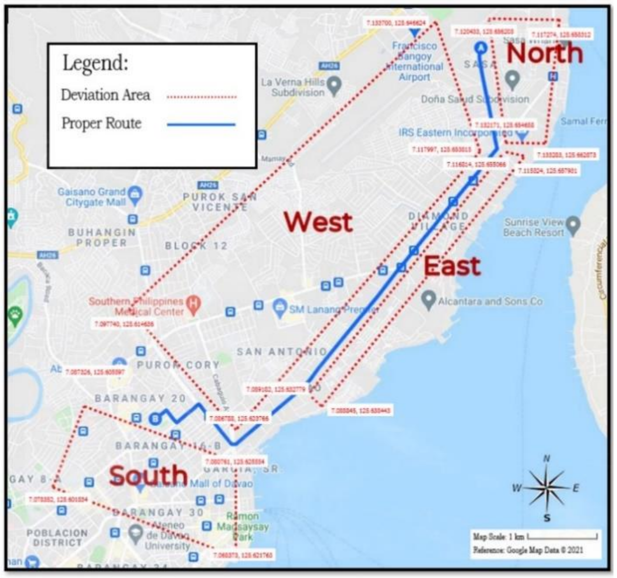


Fig. 9. Testing Route (Davao City).

C. *Objective 3*

The program showed the syntax on how the areas in the map were coded in the Arduino IDE. The numbers in the program were the coordinates of each corner of the area. There were four coordinates used to identify an area, since the area was identified by enclosing a specific location using a quadrilateral polygon. The highlighted line indicated the format of inputting the coordinates.

```
//------> this format to know GPS if inside Polysides! --->
polysides_ai_thinker(lat a,long a      ,lat b,long b     ,lat c,long c       ,lat d,long
d      ,lat location,long location)

   //Start Area Mindanao Development Authority (MinDa)   ORIGIN****
   start_dist_A = polysides_ai_thinker( 7.130701, 125.651305  , 7.131318,
125.654030    , 7.127597, 125.654617     , 7.127246, 125.651023        ,
latitude, longitude);

   //USeP Main Entrance Gate     DESTINATION****
   start_dist_B =  polysides_ai_thinker( 7.088289, 125.615335    , 7.086224,
125.617288     , 7.083541, 125.614638      , 7.085873, 125.613168     ,
latitude, longitude);


   // Sample Area points
   area_a =  polysides_ai_thinker(7.124490, 125.652515   , 7.124745,
125.655465   , 7.122456, 125.655680    , 7.121572, 125.652247    , latitude,
longhitude); //location 1 Palovince Restaurant - The Pakfry King
   area_b = polysides_ai_thinker( 7.119198, 125.654586   , 7.117708,
125.656560   , 7.115015, 125.655015    , 7.116623, 125.652140    , latitude,
longitude); //location 2  F. Bangoy Elementary School
   area_c = polysides_ai_thinker( 7.112428, 125.648911   , 7.110991,
125.650810   , 7.107158, 125.648460    , 7.108883, 125.645113    , latitude,
longitude); //location 3 Nova Tierra Square
   area_d = polysides_ai_thinker( 7.107084, 125.643794   , 7.105274,
125.646347   , 7.099698, 125.643910    , 7.102839, 125.639050    , latitude,
longitude); //location 4 Caltex Lanang
   area_e = polysides_ai_thinker( 7.100859, 125.638728   , 7.098911,
125.641829   , 7.095366, 125.638857    , 7.097048, 125.635092    , latitude,
longitude); //location 5 Diamond Hardware Republic
   area_f = polysides_ai_thinker( 7.093535, 125.632753   , 7.091746,
125.636186   , 7.089095, 125.633579    , 7.091906, 125.628644    , latitude,
longitude); //location 6 Techno-Trade Resources Davao

//Deviation Area 1 West
   deviation_a = polysides_ai_thinker( 7.133700, 125.646624    , 7.117997,
125.653815  , 7.086788, 125.623766    , 7.097740, 125.614636         ,
latitude, longitude);

   //Deviation Area 2 East
   deviation_b = polysides_ai_thinker( 7.116814, 125.655066   , 7.115324,
125.657931   , 7.088845, 125.638443    , 7.089182, 125.632779         ,
latitude, longitude);

   //Deviation Area 3 South
   deviation_c = polysides_ai_thinker( 7.080761, 125.625554    , 7.068373,
125.621763   , 7.078352, 125.601834     , 7.087326, 125.605597       ,
latitude, longitude);

   //Deviation Area 4 North
   deviation_d = polysides_ai_thinker( 7.132171, 125.654658    , 7.133283,
125.662873   , 7.117274, 125.658312     , 7.120433, 125.656203       ,
latitude, longitude);
```

The program and syntax for determining whether the point (longitude, latitude) from GPS module is inside a certain area enclosed by four points. A library has been made and was used to easily code the Arduino program. Each location, the proponent should choose and enclose a specific area using the four points. The longitude and latitude of each point from the four points should be defined in the program with the use of Google Maps. If the longitude and latitude from the GPS was inside the enclosed area, the program returned a value of 1, otherwise, 0. An if-else statement have constructed whether the program returned 1 or 0, if the program returned a value of 1 (point is inside the area), an SMS notification will be sent, otherwise, the program will continue the loop.

- Route Deviation

It notified that the prototype was able to create distinction between the correct path and the deviated path. The 14 areas were identified strategically in which the proponent can easily traverse all the areas in a minimal time period and achieve the objectives. Also, the chosen areas were used, since these areas have known landmark for easier validation and documentation. For areas 1 to 10, the prototype did not receive an alert message that indicated that the vehicle deviated, since area 1-10 were programmed as a correct path. However, when the vehicle was in area 11 to 14, the driver and the dispatcher received an alert indicating that the vehicle deviated from its correct path. It was evident that for every area, the functional objective was satisfied; thus, a success remark was recorded and prototype was able to traverse all fourteen (14) locations. The prototype sent four (4) deviation notifications to the user as it traversed four (4) deviation routes.

Table II denoted that in fourteen (14) areas where the prototype was tested, the prototype was able to distinguish the four deviation areas that was programmed in the microcontroller and successfully notified the operator. Thus, all the testing from area 1 to area 14 yielded a successful remark.

TABLE II. RESULTS FOR ROUTE DEVIATION DETECTION (TRIAL 1)

Area	Coordinates (Lat, Long)	Speed (kph)	Speed Limit (kph)	Deviated? (Yes/No)	Received SMS?	Remarks (Success/Fail)
1	7.124496,125.653671	38	40	No	Yes	Success
2	7.119050,125.654754	43	40	No	Yes	Success
3	7.111478,125.650077	24	40	No	Yes	Success
4	7.105888,125.645469	48	30	No	Yes	Success
5	7.099695,125.640449	34	30	No	Yes	Success
6	7.092534,125.634613	36	30	No	Yes	Success
7	7.085145,125.626663	42	30	No	Yes	Success
8	7.086773,125.621627	19	30	No	Yes	Success
9	7.087036,125.619079	21	30	No	Yes	Success
10	7.087199,125.616294	24	30	No	Yes	Success
11	7.082574,125.620010	36	40	Yes	Yes	Success
12	7.097366,125.614967	43	40	Yes	Yes	Success
13	7.132690,125.661613	31	40	Yes	Yes	Success
14	7.109219,125.649734	31	40	Yes	Yes	Success

TABLE III. RESULTS FOR ROUTE DEVIATION DETECTION (TRIAL 2 – DRIVER)

Area	Coordinates (Lat, Long)	Speed (kph)	Speed Limit (kph)	Deviated?	Driver Received SMS?	Remarks (Success/Fail)
1	7.124536,125.653678	51	40	No	Yes	Success
2	7.117696,125.654991	16	40	No	Yes	Success
3	7.111468,125.650077	38	40	No	Yes	Success
4	7.105462,125.645187	44	30	No	Yes	Success
5	7.098969,125.639831	32	30	No	Yes	Success
6	7.092573,125.634574	33	30	No	Yes	Success
7	7.085191,125.626670	31	30	No	Yes	Success
8	7.086732,125.621635	33	30	No	Yes	Success
9	7.087046,125.619064	28	30	No	Yes	Success
10	7.087119,125.616203	21	30	No	Yes	Success
11	7.109208,125.649772	18	40	Yes	Yes	Success
12	7.120164,125.657218	48	40	Yes	Yes	Success
13	7.084045,125.615516	25	40	Yes	Yes	Success
14	7.095824,125.616249	32	40	Yes	Yes	Success

TABLE IV. RESULTS FOR ROUTE DEVIATION DETECTION (TRIAL 2 – DISPATCHER)

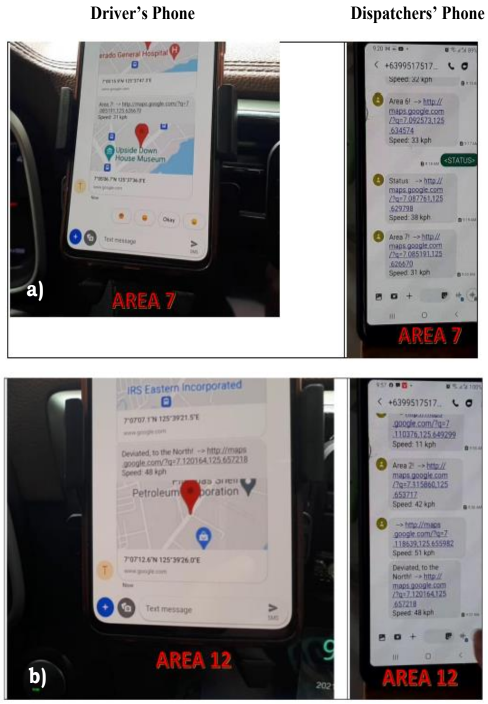
Area	Coordinates (Lat, Long)	Speed (kph)	Speed Limit (kph)	Deviated?	Dispatcher Received SMS?	Remarks (Success/Fail)
1	7.124536,125.653678	51	40	No	Yes	Success
2	7.117696,125.654991	16	40	No	Yes	Success
3	7.111468,125.650077	38	40	No	Yes	Success
4	7.105462,125.645187	44	30	No	Yes	Success
5	7.098969,125.639831	32	30	No	Yes	Success
6	7.092573,125.634574	33	30	No	Yes	Success
7	7.085191,125.626670	31	30	No	Yes	Success
8	7.086732,125.621635	33	30	No	Yes	Success
9	7.087046,125.619064	28	30	No	Yes	Success
10	7.087119,125.616203	21	30	No	Yes	Success
11	7.109208,125.649772	18	40	Yes	Yes	Success
12	7.120164,125.657218	48	40	Yes	Yes	Success
13	7.084045,125.615516	25	40	Yes	Yes	Success
14	7.095824,125.616249	32	40	Yes	Yes	Success

Tables III and IV still showed the results for the route deviation detection for trial 2. In trial 2, there were two receivers that were involved in the trial, the driver and the dispatcher. It revealed that the prototype was still able to create distinction between the correct path and the deviated path. The prototype was able to traverse all fourteen (14) locations. The prototype sent four (4) deviation notifications to the driver and the dispatcher as it traversed four (4) deviation routes. The selection of the locations was strategically chosen for it to be near and along the downtown area that have familiar landmarks for convenient validation of the results. Four locations were chosen for the deviation to enclose the preselected route used for the entire duration of the testing. The deviation areas were strategically positioned to the Eastern, Western, Southern, and Northern part relative to the proper route.

TABLE V. CONTENT COMPARISON

Test	Driver's Phone	Dispatcher's Phone	Remarks
Coordinates	Matched		Success
Received Message	Matched		Success

Table V showed the comparison between the driver's phone and the dispatcher's phone. The coordinates received by both the driver's phone and dispatcher's phone were the same, which indicated that the coordinates received by the GPS from the satellites were successfully fetched to both devices. Fig. 10 shows both the smartphones received identical messages indicating the coordinates, and the speed of the vehicle, which denoted that the message being received were not altered and manipulated.



+6399517517...

Area 7! -> http://maps.google.com/?q=7.085191,125.626670
Speed: 31 kph

Speed: 32 kph

Area 6! -> http://maps.google.com/?q=7.092573,125.634574
Speed: 33 kph

<STATUS>

Status: -> http://maps.google.com/?q=7.087761,125.629798
Speed: 38 kph

Area 7! -> http://maps.google.com/?q=7.085191,125.626670
Speed: 31 kph

7°07'07.1"N 125°39'21.5"E

Deviated, to the North! -> http://maps.google.com/?q=7.120164,125.657218
Speed: 48 kph

7°07'12.6"N 125°39'26.0"E

google.com/?q=7.110376,125.649299
Speed: 11 kph

Area 2! -> http://maps.google.com/?q=7.115860,125.653717
Speed: 42 kph

-> http://maps.google.com/?q=7.118639,125.655982
Speed: 51 kph

Deviated, to the North! -> http://maps.google.com/?q=7.120164,125.657218
Speed: 48 kph

Fig. 10. (a) Route Tracking for Area 7 for both Driver and Dispatcher; (b) Deviation Detection (North) for both Driver and Dispatcher.

- Status Command

In Table VI, it revealed that the prototype was tested ten (10) times to evaluate the functionality of the status command. STATUS command was evaluated by sending a STATUS request to the prototype. The proponent sent five (5) requests to the prototype and the prototype sent its current location (coordinates), speed (kph), and a link to the Google Maps to plot the location of the vehicle based on its coordinates. It affirmed that prototype was able to respond to the request five (5) times successfully.

TABLE VI. RESULTS FOR STATUS COMMAND

Entry	STATUS Requested? (Yes/No)	Received SMS (Yes/No)	Description (Success/Fail)
1	Yes	Yes	Success
2	No	No	Success
3	Yes	Yes	Success
4	No	No	Success
5	Yes	Yes	Success
6	No	No	Success
7	Yes	Yes	Success
8	No	No	Success
9	Yes	Yes	Success
10	No	No	Success

On the other hand, the proponent also did not send a message to the prototype five (5) times. Then, the prototype also did not respond, which indicated that the STATUS command was successful.

- Emergency Request (SOS Command)

Table VII showed the results for the SOS command. The SOS command was evaluated by pressing the SOS button five (5) times. It was expected as the user pressed the SOS button, the prototype should send SMS notification to the receiver (operator or main office) with the current location and an “Emergency Alert” phrase to indicate a distress signal. The proponent sent five (5) SOS command and the prototype was successful in sending five (5) emergency messages to the receiver. However, the proponent also did not press the SOS button, which indicated no emergency. Thus, the prototype also did not send an emergency message to the receiver; thereby, it suggested that the prototype worked.

TABLE VII. RESULTS FOR SOS (EMERGENCY ALERT BUTTON)

Entry	SOS Requested?	Coordinates	Received SMS?	Description (Success/Fail)
1	Yes	7.117279,125.654731	Yes	Success
2	No	-	No	Success
3	Yes	7.115866,125.653663	Yes	Success
4	No	-	No	Success
5	Yes	7.113910,125.652076	Yes	Success
6	No	-	No	Success
7	Yes	7.109687,125.648620	Yes	Success
8	No	-	No	Success
9	Yes	7.098881,125.639793	Yes	Success
10	No	-	No	Success

Furthermore, the algorithm of the program was to check if the coordinates from the GPS module were inside a certain area enclosed in a quadrilateral polygon. If the coordinates retrieved by the GPS module were inside a certain area, the device sent an SMS notification indicating the specific area where the coordinates were situated.

D. *Objective 4*

Table VIII showed the percent error of the three major functions of the prototype. For route deviation detection, out of 14 total number of trials, the prototype performed its functionality 14 times consecutively, which yielded to a percent error of 0%. For the status command, the functionality was assessed by sending five status requests and five no status requested (total of 10), in which the prototype was able to perform each function. The percent error yielded to 0%. Moreover, the route deviation was also tested for sending SMS notification for both driver and the dispatcher, in which the percent error still yielded to 0%. Lastly, for the emergency command, the SOS button was pressed five times and the prototype sent an SMS five times also. Moreover, the SOS button was not pressed five times. The button also did not send an SMS notification. Table VIII showed 0% of percent errors for all the major functions, which denoted that the prototype was able to perform its desired functions.

TABLE VIII. PERCENT ERROR OF MAJOR FUNCTIONS

Functionality	Number of Success	Total Number of Test	Percent Error
Route Deviation Detection (Driver)	14	14	0%
Route Deviation Detection (Driver & Dispatcher)	14	14	0%
Status Command	10	10	0%
Emergency Command (SOS)	10	10	0%

E. *Interpretation of Results*

The proposed vehicle tracking system using Arduino Mega 2560, NEO 6M GPS module, SIM900A GSM module, and push button was successful. Trials were conducted to assess the functionality of each major function, which included deviation detection, current status monitoring, and SOS SMS notification. Table I showed that out of the 14 distinct locations tested, all got a successful remark. The prototype was able to distinguish whether the area was proper route or deviation route. All ten proper routes and four deviation routes were successfully identified. Table VI showed that out of ten trials, the remarks were all successful. Five trials were performed to simulate a status request from the prototype and five trials were also conducted to request no status. Each trial that requested status report was successful as well as to the no status requested. Table VI also suggested that the STATUS command’s functionality was a success since no failure remark was recorded. In Table VII, the data showed that all ten trials for SOS command were successful. Five trials were conducted to simulate an emergency alert in which the prototype successfully sent five emergency notifications. Moreover, five trials were also done to test the no emergency (button not triggered), the data indicated that the prototype did not send emergency notification five times consecutively. Overall, all the tests recorded a successful remark and no failure.

Apparently, based on the reviews of previous literary works, there were no present prototypes, as of the date during the conduct of the study, were published that could detect if the vehicle deviated from its route, which is the primary feature developed in this study.

Lastly, during the conduct of the study, it was encountered that the device had taken time connecting to the satellite before acquiring the GPS coordinates that were needed to track the vehicle, which could identify if the vehicle deviated from its path and calculate the vehicles' speed.

V. Conclusion

Based on repeated testing and evaluation, the proponent can conclude that the study met its objective. The study proved that developing a vehicle tracking prototype with route deviation detection, emergency command, and current status command is possible using Arduino Mega 2560, uBlox NEO 6M GPS module, SIM900A GSM module and a pushbutton. The results showed that the prototype was able to distinguish the deviated path from the correct path. Moreover, the prototype also was able to distinguish whether the user pressed the SOS button or not, and if the user sent a "<STATUS>" command or not. The results also suggested that based on the actual testing, the prototype successfully performed its functional objective as a whole. There were issues encountered, such as the delay in receiving the text messages due to signal connectivity constraints, and the GPS module also had taken time connecting to the satellites due to buildings and walls that may block the signal and weather factor. Compared to the use of camera and any visual guidance system, the proposed prototype can detect deviation and deal with non-conventional factors, such as uneven road, too bright and/or too dim pathways.

References

[1] T. Noman, S. Hossain, S. Islam, M. Islam, N. Ahmed and M. M. Chowdhury, "Design and implementation of microcontroller based anti-theft vehicle security system using GPS, GSM, and RFID," in 2018 4th International Conference on Electrical Engineering and Information Communication Technology (iCEEiCT), 2018.

[2] V. K. Sadagopan, U. Rajendran and A. J. Francis, "Anti-theft control system design using embedded system," in Proceeding of 2011 IEEE International Conference on Vehicular Electronics and Safety, 2011.

[3] M. Abuzalata, M. Momani, S. Fayyad and S. Abu-Ein, "A Practical Design of Anti-theft Car Protection System Based on Microcontroller," American Journal of Applied Sciences, p. 709, 2012.

[4] "Insurance Information Institute," February 2021. [Online]. Available: https://www.iii.org/fact-statistic/facts-statisticsauto-theft.

[5] "World Data Atlas," February 2021. [Online]. Available: https://knoema.com/atlas/ranks/Private-car-theft-rate.

[6] D. J. Barrera, R. M. Omaña, G. Alviola, R. Grapa, J. Ojales, K. Antonio and J. Montinod, "Motor Vehicle Theft in Negros Oriental, Philippines: Patterns Across Space, Time, and Targets," Prism, 2015.

[7] J. M. M. Khin and N. N. Oo, "Real-Time Vehicle Tracking System Using Arduino, GPS, GSM and Web-Based Technologies," in International Journal of Science and Engineering Applications, Myanmar, 2018.

[8] R. L. Sharp, T. T. Cable and A. Burns, "The Application of GPS Visitor Tracking Implications for Interpretation at Heritage Sites," in Journal of Interpretation Research, Manhattan, 2020.

[9] R. M. Shukla, "Low-Cost DTMF Controlled Landmine Detection Rover," in International Research Journal of Engineering and Technology (IRJET, Maharashtra, 2019.

[10] R. C. Reddy, P. Krishna, M. Chaitanya and M.Neeharika, "Security System Based on Knock-Pattern Using Arduino and GSM Communication," International Journal of Engineering and Techniques, pp. 154-157, 2018.

[11] C. Sakhtivel, S. Umar Maktar, M. S, R. G and K. V, "Vehicle Security System using Embedded and GSM Technology," Global Research and Development Journal for Engineering, pp. 269-2743, 2018.