2016

BRACU BAT GPS BASED TAXI FARE METER

Thesis Report

Supervisor: Dr. Md. Khalilur Rhaman

Conducted by:

MD. Toufiq Hasan
13121157

Benazir Mustafa
13321026

Tanvir Ahmed
13121091

Department of electrical and electronic engineering

BRAC University

Submitted on:
14 Dec 2016

Dedication

This project is sincerely dedicated to

Our prophet

MOHAMMAD [PBUH]

Mercy of all humankind

Our parents

MD. ISMAIL HOSSAIN
JANNATUL FERDUS

MD. MUSTAFA
SUFIA BEGUM JASMIN

MD. TAHIZ UDDIN
GULSHAN ARA BEGUM

Who have supported and encouraged us throughout the years

And

Our beloved motherland

DECLARATION

I hereby declare that this thesis is based on our own work and research. To the best of my knowledge and belief, it contains no material previously published or produced by another party in fulfillment, partial or otherwise, of any other degree or diploma at another University or institute of higher learning, except where due acknowledge is made in the text.

Signature of Supervisor

Signature of Authors

Dr. Md. Khalilur Rhaman

MD. Toufiq Hasan Anik (13121157)

Benazir Mustafa ID (13321026)

Tanvir Ahmed (13121091)

Acknowledgements

Praise to be Almighty ALLAH for giving us blessing, strength and knowledge in finishing this thesis entitled “BRACU BAT GPS BASED TAXI FARE METER” and peace be upon His best messenger **Mohammad**(SallallahuAlayhiWaSallam)**,** his pure descendant and his family and his noble companions.

We are especially thankful to Dr. Md. Khalilur Rhaman, we would like to express deep gratitude to him for his guidance, encouragement and gracious support throughout our project without which this thesis work would not have been possible. We are also grateful to the BRAC University Faculty Staffs of the EEE and CSE department, who have helped us during the whole study period at BRAC University, particularly during planning and building the thesis project. Finally, we must thank our family members and friends who supported us pursuing our goal.

Abstract

BRACU BAT GPS BASED TAXI FARE METER

The objective of this thesis project is to design technologically advanced fare meter for transportation. In our country all the taxicabs use digital meters which are imported from foreign countries. These meters are easily tampered and taxi drivers can change fare rate with the help of technicians. As a result passengers often pay more than the fare price. Our fare meter works with GPS to provide exact distance and time then calculate fare price according to authorized law to provide passengers the fare payment. Such device assures fare charge and also the charge rate cannot be changed by unauthorized persons thus eradicating taxi fare hacking dispute. Fare increment rate will be controlled by the government and device safety status can be observed by authorized persons online. Our device will also use Smart Fare meter technologies which are modern enough to connect with android phones. It provides options for display travel map on mobile screen, ensures security issues, mobile payment and vehicle

BRACU BAT GPS BASED TAXI FARE METER

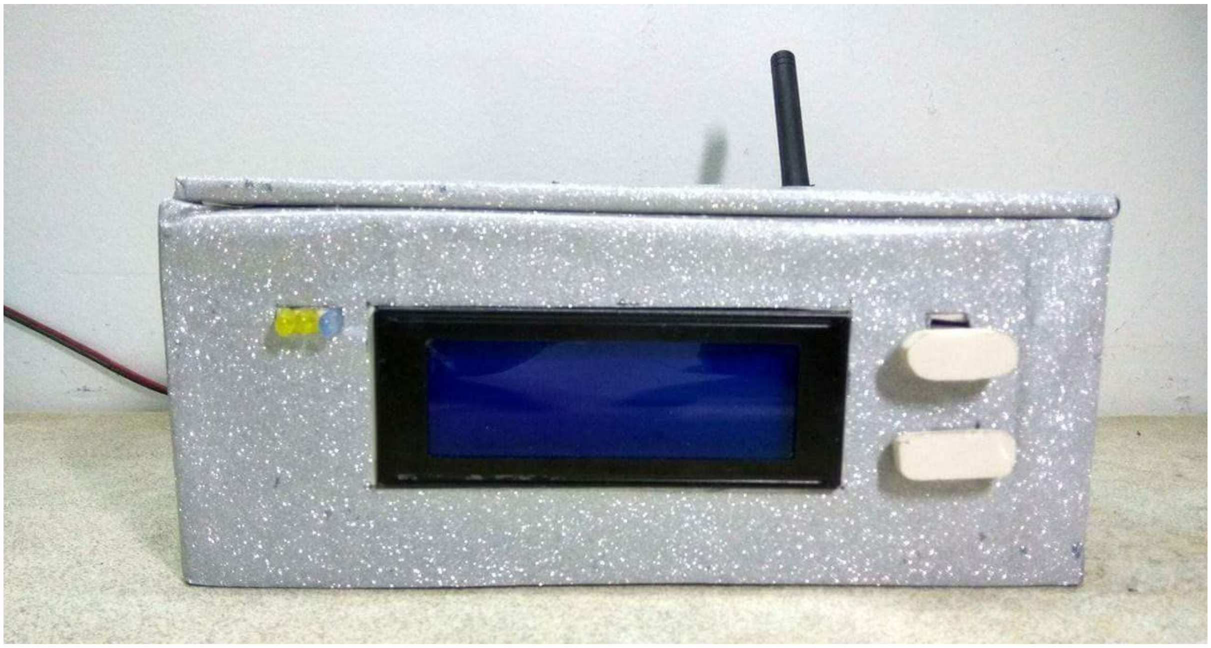


Table of Contents

LIST OF FIGURES

Chapter 1:
Introduction

Public transportation is one of the most demanded basic needs of a developing city/town. It has been decided to design an ingenious fare meter for vehicles so that passengers pay fare price along with the assurance that they will not become a victim of meter deceit. The system provides exact distance travelled, fare charge, travel time, shows route on user's android handset. Thus passengers get every single information that they were deprived off before. Also, the fare meter and system are modern enough to work with online server along with options to develop it in future as requirements.

1.1 Motivation

The demand of transportation in the cities and towns of Bangladesh has become extremely high. As a highly populated developing nation in our country urban areas provide most of the job opportunities and residence facilities. But unplanned urbanization created enormous problems, one them being transportation. Different public transportation Vehicles such as taxi cabs, CNG rickshaws, buses are the main transportation medium for urban areas like Dhaka city. Huge amount of people use this communication systems every single day of the year making them vulnerable to many kind of fraud and corruption. Because of seats on vehicles being scarce, these transportations are used by almost all classes of people even if they are paying more than the fare price. Passengers sometime have to bargain or indulge in quarrel with drivers against their wish. Crimes from meter tampering to robbing passengers, all related to public transportation are unusually high in cities like Dhaka. The device can solve all problems related to public transportation and it provides everything a modern fare meter requires nowadays.

1.2 Existing Taxi meter

Generally taximeter is an electro-mechanical device installed in taxicabs and auto rickshaws that calculates passenger fares based on a combination of distance travelled and waiting time. It does it by calculating distance through an electro-mechanical cable known as sensor cable. The electrical circuit on taxi meter calculates total distance and time of the travel period.

Fig 1: General Electro-Mechanical taxi meter display

The sensor cable is connected through a long cable directly to the taxi meter. There is a ring inside the sensor cable which turns as the wheel rotates. For every rotation it produces electric pulse, this is sensed by the taxi meter motherboard and calculates the distance.

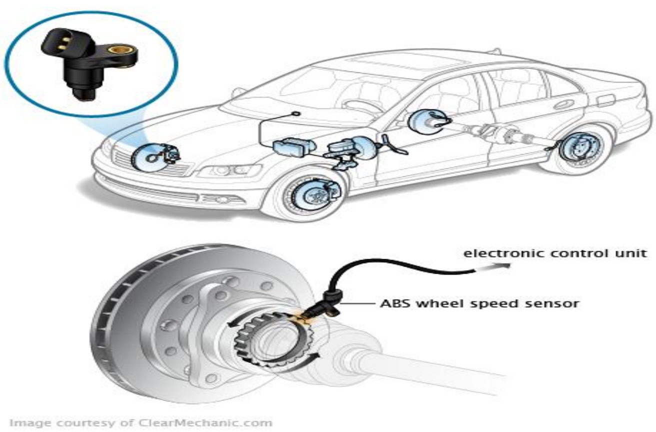


Fig 2: Odometer on a car wheel

There are several ways of measuring distance using odometer. For example most used system is to put a magnet on wheel and a device on the chassis of a car known as pickup device. This device counts every rotation of the wheel through magnetic sensors. In modern taxi meters a single cable is used for the purpose of odometer. This allows measuring distance and speed at the same time. Another way is the sensor cable which produces electric pulse to the meter to calculate distance. This is very cheap but the device wears out very soon and has to be changed quite often. On average it costs 200-350 taka to buy it while taxi drivers are trained to change it. Another more modern method is to use light sensors on pickup device. This works just like a computer mouse and is very durable. The light sensor picks up voltage for every rotation and sends this data to taxi meter. Nowadays, GPS based odometers are also used, but due to complexity and cost its limited to special vehicles.

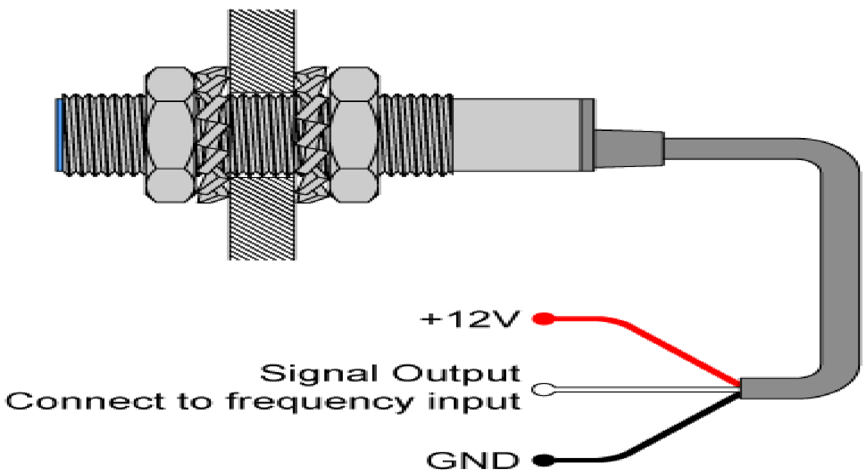


Fig 3: odometer sensor cable

There are complex electrical circuits on a taxi cab meter. This circuit receives all information from sensors and calculates distance, waiting time as well as fare charge. For no distance travel against time the meter adds charge for waiting time. Taxi meter circuit board produces clock pulse, more than 200 pulses per second. This pulse and data from pickup device are compared. Very little voltage spike per second from sensor cable compared to high pulse from mother board indicates the vehicle is not moving at all. This puts the fare charge count to waiting fare charge.
The complex circuit board also does many vital jobs to make the meter reliable. For example, taxi meters can detect reverse motion of vehicles, thus going reverse will not reduce the total

distance travelled but show the correct calculations based on the command given in the programming of the device.

1.3 Objectives

The first goal is to design a fare meter which provides fare charge for distance travelled. And it has nothing to do with persons who have access to the fare meter. Which means drivers or technicians who can replace or open fare meter cannot tamper the device. Thus meter tampering is out of question in our device. On the other hand, the device calculates distance based on GPS which gives it the modern and technological advantage that a fare mater nowadays must have. Positions taken by satellites and calculated on device produce more accurate and damping fee information. This system provides location information to passengers so that passengers know where they are at a time, making travel more safe and informative to the passengers. The fare charge per kilometer, waiting time and other charges all information needed for total payment charge are collected from online server continuously as the vehicle is on the move. Being connected to online and as positions are taken from GPS guidance system the very rout of travel can be seen on an android device if passenger or authority wills. This provides much needed security information related to transportation. Owner of the vehicle an also oversee their owned vehicle. The very vehicle can be tracked by authority if needed in case of a suspicious criminal motivation. The device is capable of connecting passenger's android phone thus travel rout, fare charge and other information such as travel time, traffic jam time wastage are can be shared with the passengers. This action may become a must have option in future as a passenger may be asked for travel details and all information by this employers as time is very valuable for working class people. GSM module is used to interact with online server through internet. The very device uses recent technology and processors platform which allows further modification if more requirements occur.

1.4 Intelligence of the device

The device is designed based on platforms that have capability to work with modern computers devices. The GPS positioning system provides not only travel distance but exact location from time to time thus passengers know where they are. His feature is appreciated by any passenger. If there is an accident, sudden vehicle malfunction or emergency the device is intelligent enough to send message to proper authority (police/owner of the vehicle/ government). It has been considered that traffic jam as a very important factor during the designing of the fare meter as many places of Dhaka consume several hours every single day. The waiting period during this traffic jam is crucial and the waiting time is calculated in a very special way so that the driver is not deprived of the fare charge. The processor of the device senses a traffic jam from its movement through GPS positions. Entire rout of the travel is saved on server as a CSV file which can be processed later for possible security issues or passengers safety. The fare meter connects with passenger's android device through Bluetooth thus information is shared very easily. Finally the modules, sensors and processor units all are compatible with each other and structured together in a stacking system which makes it 70% smaller than conventional devices. The same characteristics allow easy replacement of components if needed.

Chapter 2:
Primary Data collection

Before the construction of any device it is important to get information from the users that will use it regularly. As they use it on daily basis and know everything about using it. The very small requirements and big troubles are known through the persons who keep direct contact with the device. There for primary data collection has been done in our project as well before starting the construction of the fare meter.

Importance was given on taxi driver’s demands, complaints and suggestions. It has been researched thoroughly that what makes the present taxi obsolete. The research target was also to know that the advanced and secure features of our device can turn it into a very useful device which may replace all existing taxi fare meters. Many taxi cab drivers were questioned thoroughly about very specific questions. They were asked if the new meter would make it easier to eliminate bargaining, meter forgery and all related crimes. Also it is found if user friendliness and advanced features are compatible with requirements or not. The cost of new GPS based meters and their long term service life were discussed. New meters are needed indeed to provide the best service thus public opinion about what they want is very important.

Primary data:

Before the construction of the BRACU BAT taxi fare meter primary data were collected directly from taxi cab drivers. 120 taxi drivers were asked about the questions mentioned below to learn about the present state of the CNG meter.

1. **আপনারসিএনজি’তেমিটারআছে?(১২০জন)**
   হ্যাঃ৯৩নাঃ২৭
   ঢাকাশহরএরভিতরেমিটারআছে- সবসিএনজিতেই
   ঢাকাশহরএরবাইরেমিটারআছে-৫০% এরকম

2. **প্রতিকিলোমিটারভাড়াকতবর্তমানে?(৮০জন)**
   ১২টাকা
3. **প্রতিকিলোমিটারেরজন্যআপনারাকতভাড়াচান?(৪৫জন)**
   ৩৪জনঃযেটাসরকারঠিককরেদিয়েছে(১২টাকা)
   ১১জনঃ১৪টাকাকরাউচিৎ
4. **সিএনজিমিটারব্যবহারকরেন্যায্যমূল্যপাওয়াযায়কি?(৪০জন)**
   হ্যাঃ২৬জন
   নাঃ১৪জন, সিএনজিগাড়িরঅনেকখরচআছেতাইবাড়তিভাড়ালাগে
5. **যানজটেরজন্যকিআপনাদেরআয়েপ্রভাবপরে?(৭০জন)**
   হ্যা, ট্রাফিকজ্যামেদিনেকয়েকঘন্টাবসেথাকতেহয়
6. **ওয়েটিংটাইমএরসময়প্রতিমিনিটকতভাড়াপানআপনারা?(১০জন)**
   নির্ধারিতভাড়াপ্রতিমিনিট২টাকা
7. **ওয়েটিংটাইমেআপনারাপ্রতিমিনিটকতভাড়াচান?(৪০জন)**
   ১।৪টাকাকরাউচিৎ, কারনঅনেকসময়ব্যয়হয় (৭জন)
   ২।যেটাআছেতারবেশিচাইনাহ, রাস্তারযানজটকমানোরদাবিজানাই
8. **সিএনজিমিটারএরবিক্রয়মূল্যকত?(১০জন)**
   ২০০০টাকাথেকে১০০০০টাকা, গুনাগুনেরউপরেনির্ভরকরে
9. **কোনকোনদেশেরসিএনজিমিটারবাংলাদেশেপাওয়াযায়?(২০জন)**
   ইন্ডিয়া, চায়না, কোরিয়া, তাইওয়ান
10. **সিএনজিমিটারলাগিয়েনিতেকতখরচহয়?(১০জন)**
    ৩০০থেকে৪০০টাকা
11. **সিএনজিমিটারএরকোনকোনপার্টনিয়মিতপরিবর্তনকরারপ্রয়োজনপরে?(২৫জন)**
    মিটারখারাপহয়মাঝেমাঝে, কিন্তুমিটারেরতারপরিবর্তনকরারপ্রয়োজনপরে
12. **মিটারেরতার(ওডোমিটারেরকেবল) এরমূল্যকত?(১০জন)**
    ৩০০টাকা
13. **কতসময়পরেমিটারেরতারপরিবর্তনকরারপ্রয়োজনপরে?(২০জন)**
    প্রতি২-৩মাসঅন্তরঅন্তর
14. **সিএনজিগাড়িরমালিককে?(১২০জন)**
    চালকঃ২৪জন
    ব্যবসায়ীঃ১৫৬
15. **সিএনজিগাড়িরমিটারভালকাজকরছেকিনাগাড়িরমালিককিতাজানেন? (৯০জন)**
    হ্যা, মালিকগাড়িরসবখবররাখেননিয়মিত।কিছুনষ্টহলেজরিমানাকরেন
    নাঃ০জন
16. **প্রতিদিনসিএনজিগাড়িরভাড়াকতদিতেহয়?(২০জন)**
    ৯০০টাকা
17. **আপনারান্যায্যমুল্যেরবেশিভাড়ানিয়েথাকেনকেন?(৮০জন)**

রাস্তাখারাপতাইগাড়িনষ্টহয়, যানজটেসময়নষ্টহয়, পুলিশেরজরিমানাঅনেকবেশি, নির্ধারিতভাড়ানেয়াহচ্ছেকিনাতাতদারকিকরাহয়নাহ।

18. **নির্ধারিতভাড়ারকতবেশিপেলেআপনারাযাত্রীনিয়েথাকেন?(১২০জন)**

১।কমদুরত্তেমিটারেরভাড়ারথেকে৫০টাকাবেশি (৩৬ জন)

২।বেশিদুরত্তেমিটারেরভাড়ারথেকে১০০টাকাবেশি(৩৬জন)

৩।যানির্ধারিতভাইনেয়াহয়, বেশিনেইনাহ (৮৪জন)

19. **কোনপধ্যতিতেগেলেবেশিআয়হয়মিটারেনাকিমিটারছাড়া?(৫০)**

১।তেমনতফাৎনেই, মিটারেসামান্যকমভাড়াউঠে (৪৩)

২।মিটারেগেলেঅনেককমভাড়াউঠে, মিটারছাড়াবেশিআয়হয় (৭জন)

20. **সিএনজিমিটারেকিকারসাজীকরেভাড়াবাড়ানোযায়?(১২০জন)**

১।আমারজানানেই (৬১জন)

২।নাকরাযায়নাহ (২৭)

৩।হাঅনেকেকরে (৩২জন)

21. **চালকেরাকিমিটারেরভাড়াবারাতেপারেন?(১২০জন)**

নাঃ১২০জন

22. **কারামিটারেরভাড়ারপরিবর্তনকরতেপারেন?(১২০জন)**

১।আমারজানানাই (১০৩)

২।মেকানিকপারেন (১৭জন)

23. **মিটারখারাপথাকলেপুলিশকতজরিমানাকরে?(২০জন)**

১।মিটারখারাপথাকলেকোর্টেকেসদেয়, ১২০০টাকাজামিন

২।মিটারেরতারখারাপথাকলে৪০০টাকাজরিমানাদেয়

24. **ব্রাকমিটারসঠিকভাড়াহিসাবকরলেআপনিতাব্যবহারকরবেন?(১২০জন)**

ভালহলেঅবশ্যইব্যবহারকরব, তবেনির্ভরযোগ্যহতেহবেএবংমূল্যকেনারসামর্থেরভেতরহতেহবে।আমাদেরদেশেভালমিটারতৈরিহলেআমরাখুশিহব।

25. **ব্র্যাকমিটারেরসবগুবিধাকিআপনারভাললেগেছে? (৪০জন)**

ভাললেগেছে, অনেকআধুনিক (সঠিকউত্তরপাওয়াযায়নি, চালকেরাএতনতুনপ্রযুক্তিরসাথেপরিচিতনাহ)

26. **এতআধুনিকসিএনজিমিটারআপনারাব্যবহারকরতেপারবেনকি?(২০জন)**

হাতে-কলমেশিখিয়েদিলেঅবশইপারবো, যেহেতুআধুনিকতাইব্যবহারেসুবিধাইহবে।

2.2 Data analysis

Based on the primary data we collected from 120 taxi cab drivers some very important discoveries have been made about making the BRACU fare meter. The very reasons of high expense and maintenance are found through primary data collections. Those problems are mostly solved in the new fare meter. The extra payment and bargaining for fare charge instead of offering the meter charge is researched. What makes the meters expensive, unreliable and prone to failures are found through primary data collection. The data analysis shows all the needed answers.

2.2.1 Fare charge

It was found form the research and data analysis that the fare charge price per kilometer selected by the authority was sufficient and satisfactory. However, many taxi cab drivers are not honest enough to accept the fare price. They manipulate the taxi meter or bargain to get more money. After talking to many drivers and taxi meter mechanics, it was found that meter forgery is common and not a very difficult work to do. As a result many taxi cab drivers use meter forgery to earn more money. The easiest way to do that is to manipulate the wire that connects wheel sensor to the meter. This results in voltage variation in the input of the taximeter and causes the meter to count more distance then the practical value. There is no way to stop it as long as the present meters are in use because this method is easy for the mechanics to follow. For the passengers they can’t detect or do anything about it.

2.2.2 Meter maintenance

The maintenance and service cost of present day meters is very high. The meters work mainly in electro-mechanical method which involves mechanical movement of components. This causes the parts to wear out. The taxi drivers have to change the sensor cable every 4-5 months. Often sensor wire malfunctions and it is very difficult to understand whether there is any malfunction or not. From the primary data it was found that police checks this wires on a regular basis because they wear out quickly. If they find damaged wire they fine 400-1200 taka and also court case against the driver. This factor makes the use of these meters very expensive and taxi cab drivers had to bear all the expenses. As a result drivers often charge more or fabricate meter to earn more for their losses in law suit.

2.2.3 Demand of the primary users

In the primary data and research drivers and mechanics were extensively asked about the problems they faced quit often. Also they were asked for solution and if new meter were issued to them how would they like to have those meters. The drivers demanded for reliable machines at first. They were tired of changing parts of the meters and paying much money to the mechanics for the same reason. Primary users also talked about the meter forgery and its solution. It came out as a result that meter forgery can't be stopped unless new meters are design with an intension to hack proof hardware.

Moreover, drivers welcomed the options and advantages of the BRACU fare meter. They asked about reasonable cost of the new meter. They liked the BRACU meter in a sense that it requires almost no maintenance as there are very few parts that may need to be changed.

Chapter 3:

Architecture of BRACU BAT Taxi Fare Meter

Fig 4: Overall system

3.1 General Overview of the System:

The entire system of this project involves BRACU taxi meter, satellite connectivity, server data store unit, mobile application service and Bluetooth connectivity. To get the location of the transport vehicle it is required to get at least three satellite connection at a time so that it can calculate latitude and longitude position of the meter device. By using the latitude and longitude values, the meter calculates the distance and combines all necessary calculation for the meter. It has online server offered by data spark fun to store the data information about the vehicle movement. GSM module connected with the meter device sends these data information to the cloud and stores all the movement information in the associated link of that particular meter. So the owner of vehicle can only get the access to the information. An android mobile application has developed to help the passengers to get the route they are travelling along with the map and also to satisfy them with the taxi service. This mobile application is connect via Bluetooth with the meter device. By establishing the Bluetooth connectivity user or passenger can get the travelling data information and some more features. This application has a link to connect the passenger with a map providing passenger location and traveling route.

3.2Understanding Tasks and Our Approach:

To design a fare meter all tasks have been divided in small partitions each of which requires specific arrangement with different groundwork. Meanwhile the separated tasks have to be compiled together in a single program with as little memory as possible to run it smoothly. Arduino platform was selected as processor to run the program. The different tasks are GPS positing, distance calculation, fare charge calculation, output display, initial memory storage, permanent memory storage, online server storage, server data sharing, power supply.

3.2.1. GPS Positioning

Global positioning system (GPS) was chosen as the method of getting position and distance. But not all GPS devices provide precision location while neither we have sources to work with very high end expensive devices, therefore choosing available GPS module in the budget was a challenge. Some devices will not work inside a vehicle or fluctuate under covered area such as beneath a flyover. On the other hand, devices can be very big and will not fit with the processor units. Thus reliable device is chosen from renowned GPS module manufacturers who provide on line services of the same products. This GPS module is also compatible with other working units we have planned to work with.

3.2.2Distance Calculation

The very concept of distance measurement using GPS is to work with one position with the next position and calculate the length of those two locations. This is a complex procedure which requires mathematical calculation in our main program. The program is however not just for two consecutive locations but for locations which are taken every second through the GPS module. These positions have to be used to measure distance. Location co-ordinates are taken every milliseconds thus they may consume large memory. Furthermore, location's co-ordinates must run in a serial sequence for the main program to run perfectly otherwise calculations can stop abruptly. Every time distance is measured it must be saved to memory for calculating the total distance travelled. Continuous addition of distances with the memory stored distance sum up to the final distance travelled. While the working algorithm seems straight forward there are several issues that may cause the program to malfunction. These issues are initial networking issues, waiting time.

3.2.3 Initial Networking Issues

We are working with different modules and sensors that are inter-connected with each other. This can cause initial calculation problems as devices take time to provide accurate information. Also the program needed to run for a few moments at the beginning before it can put all data together and show correct output. The GPS antenna needs several seconds to connect with the satellites. In real world every single GPS device needs initial time for connection before task execution. This issue shows that GPS has to be turned on the vehicle all the time thus whenever processor starts working it gets instant co-ordinates from strong satellite networks. Finally, the main program is given a minimum starting time to run initial phases properly before output is shown on display

3.2.4 Waiting Time

We have mentioned earlier about distance calculation plans, which work in continuous sequence. However, while the vehicle is moving and distance calculation is processing if the vehicle stops for some reason the distance measurement is stopped. But taxi cabs are not only paid for distance travelled but also for time that takes to reach a destination. This basic concept provides taxi driver his fare payment. The new device has to sense a complete stoppage or a traffic jam situation. Otherwise the fare meter will not be a practical equipment. This issue is known as waiting time. The taxi drivers get a specific amount of fare charge during a waiting period. The waiting period can be detected by sensing very little or zero movement compared to time lapse. This means movement of a vehicle during traffic jam has to be examined and our device has to sense that movement characteristic to work properly. Finally, waiting time is used in a different formula to calculate fare charge and then added to the total travel cost.

3.2.5 Fare Charge Calculation

The total cost of travel seems easy to calculate by multiplying distance with fare charge rate fixed by the authority. However, in practical application the calculations are far more complex than it seems. From the research we found that fare charge has to be calculated instantly as distance is measured by the processor. This method requires programming that takes the calculated distance from temporary memory then multiplies it with fare charge rate and saves in another temporary memory. As more distance is covered more calculations are run and total payment is continuously increased per distance travelled. At the very beginning of the travel charge for first two kilometer is fixed by the authority and for further distances the fare rate is different. This is done through calculation of exact two kilometers than measuring fare charge for it then adding further charge for more distance travelled. Meanwhile, there are two more situations that have to be considered. These are waiting time and traffic jam.

In the case of a traffic jam the device has to sense it and put stop to the fare charge counting. The total time spend during traffic jam is calculated after the sensors predict possible clear traffic. This total time spend during traffic jam is multiplied with special traffic jam fare rate and added to previous total charge. The device then resumes normal distance to fare charge calculations. On the other hand if the passenger makes the vehicle stop then the wailing time measured has different fare charge rate compared to traffic jam charge rate. This charge is also added to previous total in the same way and the device resumes normal fare calculation procedure. The fare calculation and totaling of fare charge is done several times in a second thus allowing correct fare charge even if the vehicle is moving in very high speed.

3.2.6 Output as Display

During the travel period all necessary information is showed to the passenger. This information can be distance travelled, total time elapsed, total fare charge, indication of states such as going or waiting etc. This information doesn’t need big or high resolution screens. In order to keep costs down and more reliable LED segmented display can be used. There are several advantages of this type of displays. First being cheap compare to android or TV display. Secondly, the hardware of LED display can withstand bumps and shaking during the movement of the vehicle. Thirdly, they consume much less energy due to the factor that there are LED diodes on the display that are very energy efficient. Finally, the main advantage is that LED displays can easily be added to the processor unit without adding any driver module which is mandatory for TV or android display systems.

3.2.7 Initial Memory Storage

Arduino processor is used which calculates all the data. The main program needs some temporary memory to run it on processor. This memory is in microcontroller of the arduino board. But for total data information of a complete travel a permanent memory is needed so that the data can be displayed and further exploited if needed. It was planned to use online server as the permanent storage for many reasons. However, initially a way was found to store memory elsewhere, because it was decided to work with a distance calculator program before we move to GSM internet connection. Initially laptop was used as the permanent storage device as it movable. The software in which the main program was programmed for fare charge calculation is installed in the laptop. Thus not only the program can be manipulated through laptop but the laptop also stores the information. The monitor of laptop provides initial display also.

3.2.8 Permanent Memory Storage

The planned programming does not need substantial amount of memory. However, during distance calculation, fare charge calculation and displaying all information memory is needed to store all the data. Most of the data is temporary and erased after the main program is run for a complete cycle. Several cycles are completed every second. Temporary memory is needed for calculation of time, distance and fare charge. While for displaying information some permanent storage is needed too. It was planned to store all possible data of a single travelling. This however needed very large memory. Also new modules were needed to add with processor unit to store data. This causes the device to become bulky and expensive. And so, it was planned to utilize the GSM module which is already used to get internet for GPS positioning. Here, sharing same internet access it was planned to store all the data on a virtual server or website.

The subsequent us of online server for memory storage and permanent data storage is much safer then hardware storage. The Arduino board has its own storage which works both as RAM and ROM during the run of main program. For the same reason the decision of using this processor was planned. While no extra memory is needed in the device permanent data is stored on server through GSM internet connection. The stored data on server can be used for further actions.

3.2.9 Online Server

The online server is basically a website. Work began with electronic equipment's and devices that are made by Sparkfun/USA which is a manufacturer of electronic research devices. Sparkfun provides temporary accounts in their website. This account works as a website which can directly receive data from the GSM device through internet. The account can be modified to send back the data as an SMS to android devices. The account can interact and receive command from our main program through internet. However, as we planned and worked there emerged many restrictions and limitation as it was a free account in a website. Therefore after working the initial targets the plan was to build a personal website for the project. If a workable website was created completely based on the necessities of the fare meter it can be further used for command, information analysis and security purposes. As a designer and owner of the website project members had no restrictions. Now it was possible to store as much data as it was needed for the research. As permanent memory storage this website can be used for extensive data analysis. Every single travel with its massive information is stored on this website. Most importantly the rate of fare charge per kilometer can be changed directly from our website. Thus the server works as the only way to change fare charge rate and solves the issue of meter tampering.

The online server was always planned as the information data base location for extensive research. Every single vehicle travel information and rout detail is saved here. Thus for security and surveillance the data can be shared by authorities who are legally allowed to have information.

3.2.10 Data Sharing

The server that was created is a website only controlled by legal authority and used for surveillance as well as tracking of every single vehicle that is equipped with the BRACU fare meter. The website was designed for storage of all the information from the fare meter in a CNF file. This is stored with date, time and device identification data. Thus when the owner of the vehicle or authority for security purposes want to see routs of the vehicle in a particular time of a particular day, they can easily get it. Thus crimes related to vehicles can be investigated and stopped effectively. On the other hand, vehicles that are not allowed to provide service outside a specific area can be tracked down. Passengers can get rout details along with travel information on a later date from server if they need for any security reasons. The server creates a complete archive of all possible data of the fare meter. In this way it can be achieved a complete transportation surveillance system for all the vehicles that are equipped with the fare meter. The use of GSM module on the device makes it easy to share information weather through internet or SMS.

3.2.11 Power Supply

BRACU CNG fare meter consists of several modules and elements each of which requires direct current source as power supply unit. The fare meter consists of GPS antenna, GPS shield, GSM module, display and arduino board. These elements were selected after elaborate research because all this components fit together on top of each other to form a multi-stack single device. Thus there is need of only one energy source which can then run the entire set of elements. The power input can be with a USB connection at initial stages when information storage and trial is done on laptop. Finally, there is a jack which works an input to five volts of DC source. This DC voltage can be applied from the batteries of any vehicle or through a DC battery separately. In our research and tests DC charger bank, intended for android phones was used.

Chapter 4:

Structure of Our Device

The fare meter is not a single piece of hardware but several electronic devices connected to each other. The very design of separate individual components connected together allows us to add more parts according to our need. However this method has a very big problem. When we talk about connecting different blocks of electronic components to each other it is not simple connection at all. The main component blocks are-

- Arduino board
- GSM module
- GPS shield
- GPS antenna
- Bluetooth
- Display
- Power supply

Circuit Diagram:

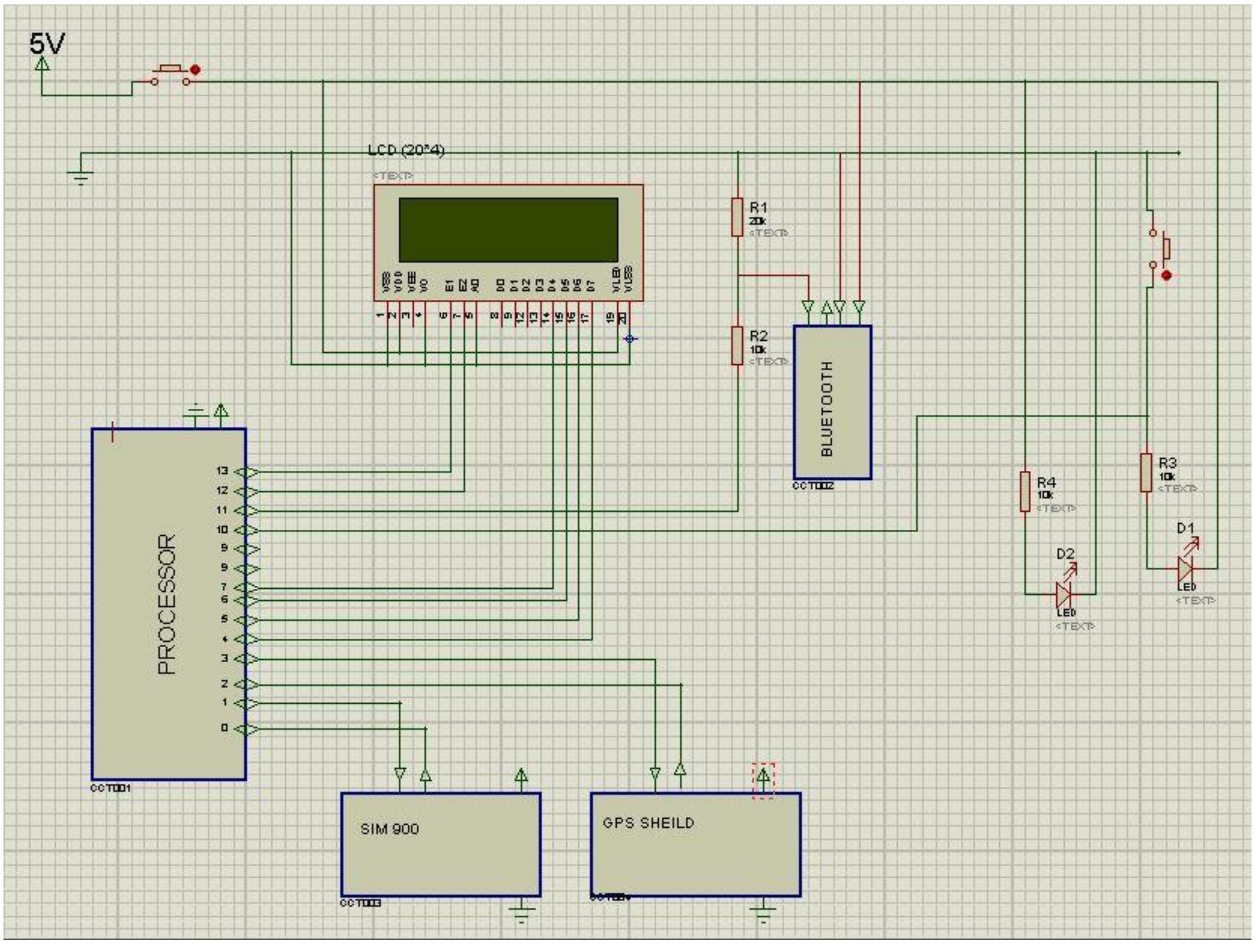


Fig 5: Circuit Diagram of BRACU BAT GPS Based fare meter

4.1 General Structure

The main processor arduino board has several pins. These pins are identified separately in two sets as input and output pins. The total number of pins is limited and as a pin is used it has to be defined that pin on the program for the main program to run. This means manually connecting these pins with all other shields, modules and display system creates two fatal problems. One is basic connection problem through conventional soldiering. While the other one is programming problem due to complex connections. The first problem is the most acute one. It was estimated that there was needed several dozen connections all together for the fare meter. Soldering so many points is not practical as well as soldering is not a reliable method to connect so delicate and fragile electronic components. So many connections would create problems to work with and on the other hand we also need to disconnect some elements to load new program or to try new experiments. These connections would also make the device very bulky and unsuitable for vehicle use.

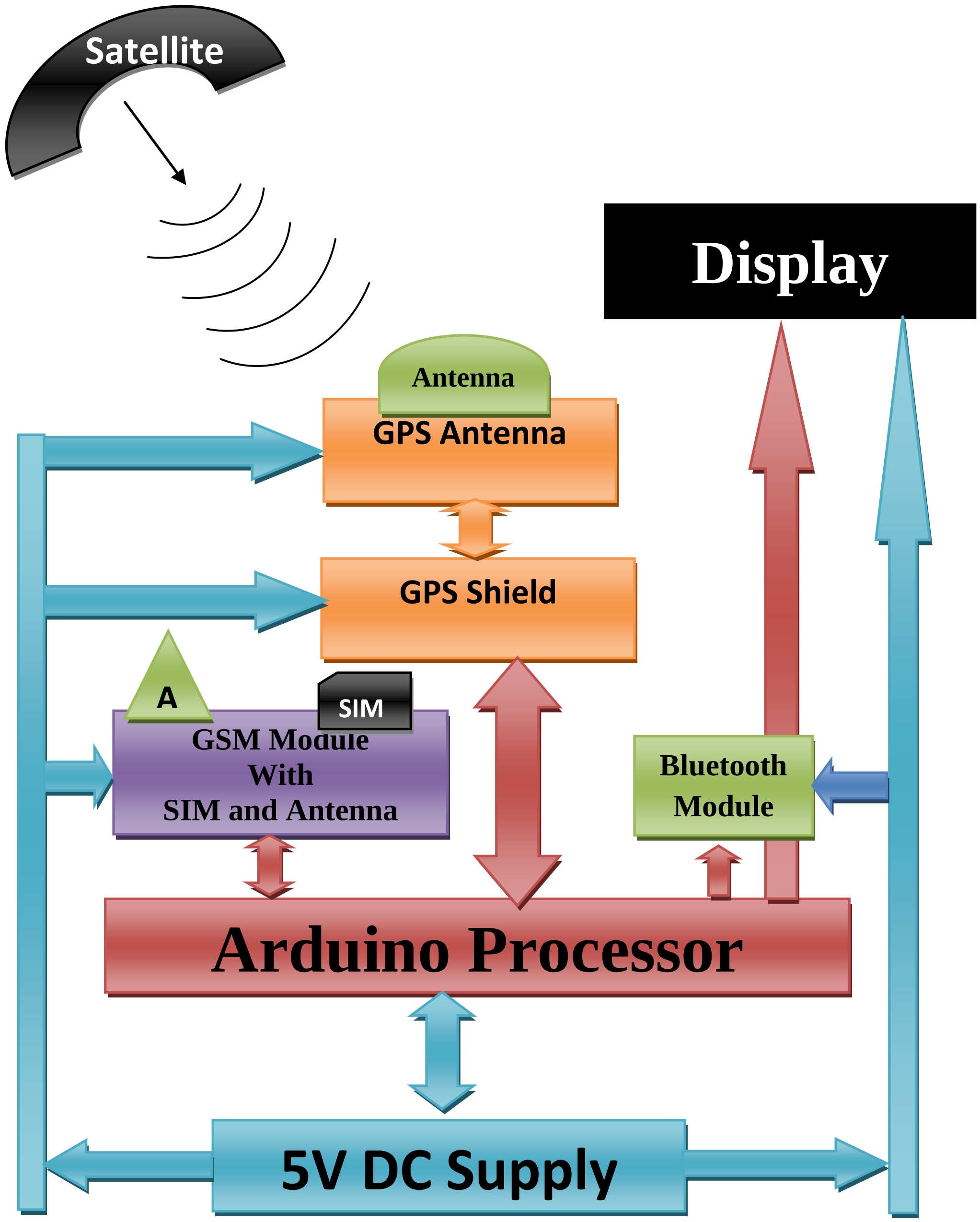


Fig 6: General connection of all the components of our device

The second problem is due to programming difficulty, as there is limited number of pins to work and every pin has to be defined in our main program. During the trial and error m method it was found that pin number is needed to change many times to get the best setting and make room for more pins so that other components can be added. Soldering would make the device work only for a specific setting and useless for other settings. So many components are used in the device and their connections are also complex and numerous. To overcome these issues it was deicide to take following measures.

4.2 Integrated Device:

Altogether considering every possible limitation and problem it was decided to work with shields and modules that are compatible to each other. This means instead of using basic elements Shield and Module will be used of those elements which are basically integrated circuit boards with all components already attached on a single board. These integrated boards have all registers, capacitors, voltage regulator, on/off switch, driver circuits needed to run the main IC chip. On this board there are many male/female pins which connect to other components using wire. Thus the only thing needed is to connect is input and output connections. Now these integrated boards have specially organized input/output pins. For example arduino board has input/output pin configuration which is similar to GSM and GPS shield both. This means we can stack the GSM module on top of the arduino board and on top of the GSM module GPS shield can be stacked. Thus these three components sit on top of another and complete the connections through male/female connection pins. In this way we don’t need to go through all difficulties to connect so many connections. This configuration helps us disconnect elements when not needed during research works.

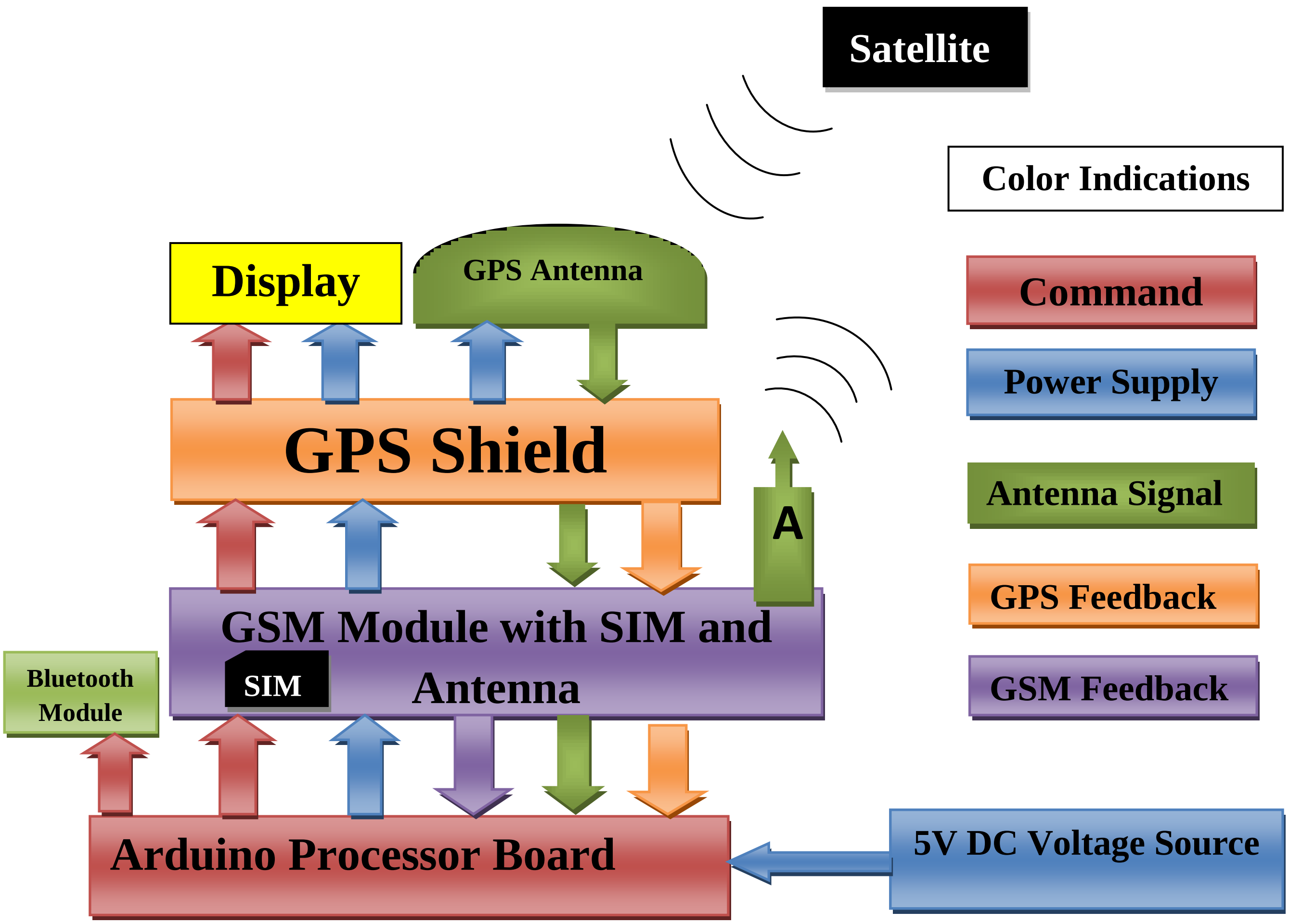


Fig 7: Stacking structure of all components

4.3 Main components of our device

Our final design of the fare meter consists of five main components. These components are_

i. Arduino processor board
ii. GPS shield with GPS antenna
iii. GSM module with SIM slot and antenna
iv. Bluetooth Module(HC-06)
v. Display
vi. Power source

Arduino Uno:

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains every driver and circuit to support the microcontroller. It can be connected to a computer with a USB cable to upload codes from computer to arduino uno. There is an arduino software for programming only for arduino uno board. After uploading the code arduino can be powered with a AC-to-DC adapter or battery to get started.

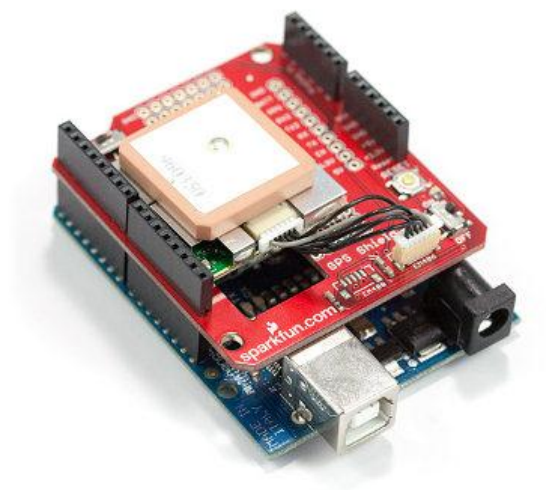


Fig 8: GPS shield stacked on top of arduino uno

Pin configuration of arduino Uno:

Arduino uno has 13 pins as digital input/output pin. These pins can be used for both as input from other elements and output command to other components. We have used pin 0(Rx) as input receiver from GSM SIM-900 and pin 1(Tx) as command output to GSM SIM-900. For GPS data we have used pin 2(Rx) as input receiver from GPS shield and pin 3(Tx) as output command to GPS shield.

Any one of the pin from 0 to 12 is chosen for input or output. These pins are defined and used in our main program. For display pin 4,5,6,7 are used as display connection (output) to display. While pin 11 and pin 12 is used for display control.

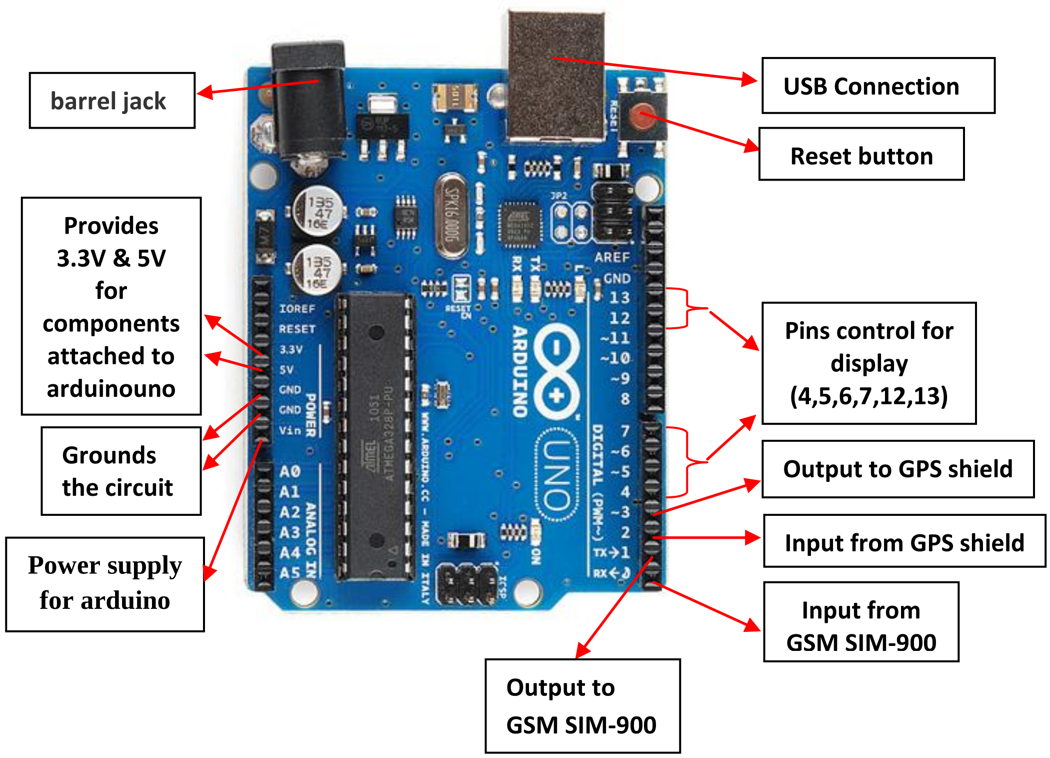


Fig 9: arduino pin connections

Initially power was provided to our device through USB port or through barrel jack. However power can be supplied to arduino through Vin pin on microcontroller side. The input recommended voltage is 7V-12V DC. Any voltage in that limit will work fine as arduino can manage its power consumption through its build in driver system. This power is managed by arduino to operate its function as well it also can supply power to multiple devices. Pin 3.3V labeled on microcontroller side provides 3.3V exactly to run sensors and motors. Also on the same side pin 5V provides 5V power to other devices. As most of the compatible devices run on these two voltage limits we don't need more batteries for different elements which are attached to arduino. During the programming on arduino software we can easily vary the output voltage according to our need. Thus arduino is very handy and flexible device to work with. There are ground points as pin GND to ground connected circuits.

Selection of arduino as main processor:

There are many options to choose from various microcontroller based processors. The very first option is microcontroller. However, working with microcontroller is very time consuming method along with several major disadvantages. Microcontroller cannot run any sensor or directly control devices. Driver circuits were needed so that loads can have power supply and work. For different devices connected to microcontroller different circuits were required to attach. This process is very complex as every single circuit is potential failure point as hand make circuits and so many joints can wire out very quickly. On the other hand, our fare meter will be used on a busy vehicle with long travels on bumpy roads as well uneven streets of the city.

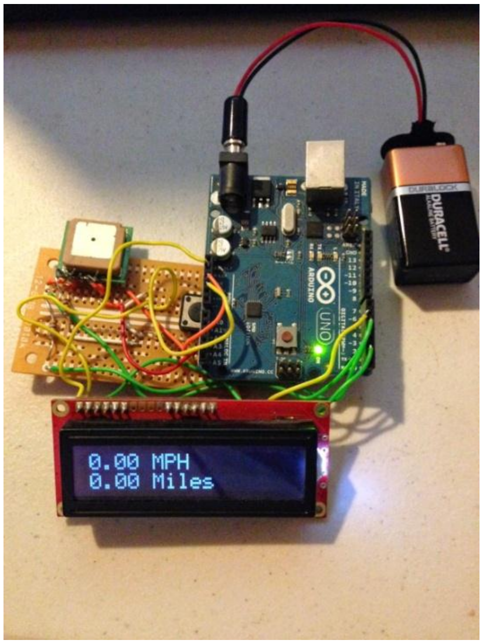


Fig 10: complex connections using manual soldering of wires

Upon consideration of these issues we decided not to spend time on building circuits or repairing them but to use very sophisticated integrated microcontroller. This will allow us to add as many devices we want to add and also is highly reliable and robust. Among many integrated microcontroller based processors arduino uno is one of the most known and easy platforms to work with.

Arduino has its own programming software. Using computer or android arduino main program can be written and uploaded through arduino USB connection. As we work step by step every time we add something to our main program we just need to detach the arduino board from all connections and upload the main program. The software can detect any faults in our program making it impossible to end up with a programming error. The flexibility and build in option to attach and detach the arduino board makes it very easy to work with. Otherwise we had to disconnect all the connections and attach them after loading the program to arduino board. This could damage the circuit as well would have been much time consuming.

GPS Receiver – EM-506 (48 Channel):

Fig 11: EM-506 GPS receiver

The EM-506 GPS receiver from USGlobalSat is based on the SiRFStarIII chipset. This complete module is built with on-board voltage regulation, LED status indicator, battery backed RAM, and a built-in patch antenna! 6-pin interface cable included.

This EM-506 GPS module is powered by SiRF Star IV, it can provide with superior sensitivity and performance even in urban canyon and dense foliage environment. With SiRF CGEE (Client Generated Extended Ephemeris) technology, it predicts satellite positions for up to 3 days and delivers CGEE-start time of less than 15 seconds under most conditions, without any network assistance.

Features:

- 48-Channel Receiver
- Extremely high sensitivity : -163dBm
- 2.5m Positional Accuracy
- Hot Start : 1s
- Warm Start : 35s / 15s with CGEE
- Cold Start : 35s / 15s with CGEE
- 45-55mA at 4.5-6.5V

Dimensions:

- 30mm x 30mm x 10.7mm
- 16g including cable

Soldering the components:

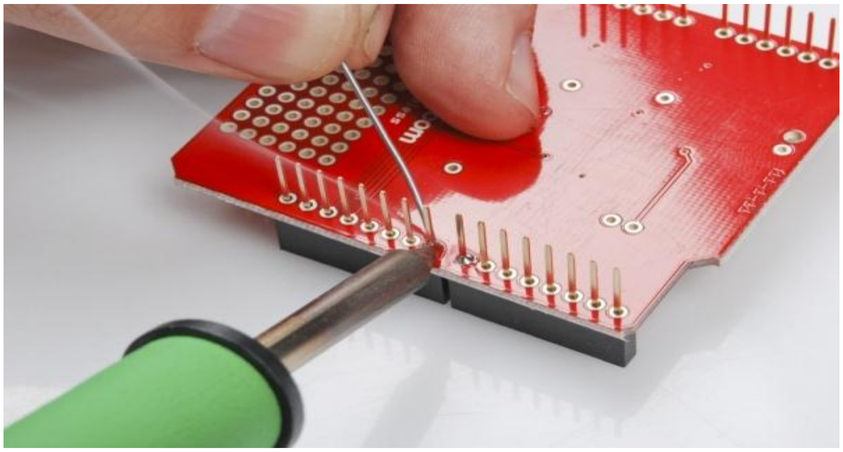


Fig 12: Soldering the kit to make stackable shield

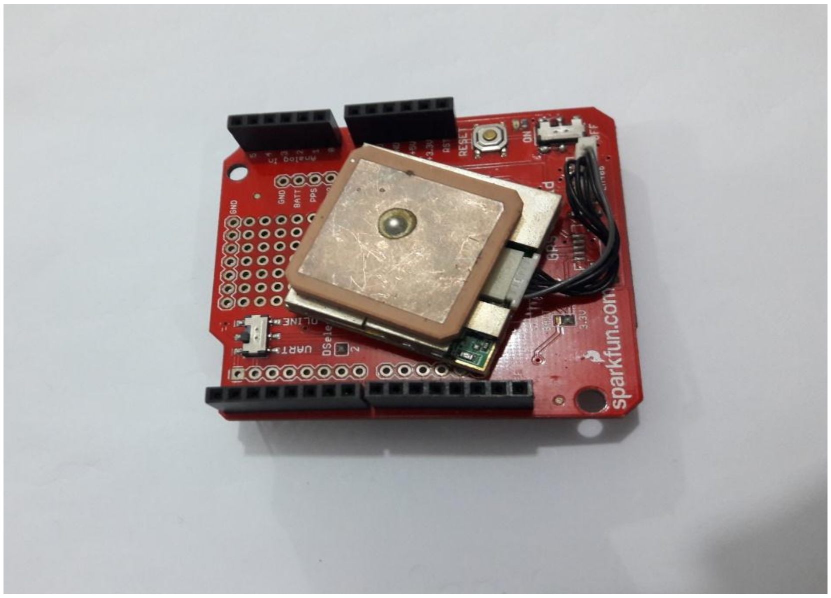


Fig 13: Elements soldered together with EM-506 antenna attached through connector wire

GSM Module-SIM900 (ICOMSAT 1.1)

IComsat is a GSM/GPRS shield for Arduino and based on the SIM900 Quad-band GSM/GPRS module. It is controlled via AT commands (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands), and fully compatible with Arduino uno.

IComsat v1.1

- Support the software power on
- Add the 3.5mm socket for microphone and earphone interface on shield

Fig 14: IComsat v1.1

Specifications:

PCB size	77.2mm X 66.0mm X 1.6mm
Indicators	PWR, status LED, net status LED
Power supply	9~20V, compatible with Arduino
Communication Protocol	UART
RoSH	Yes

Fig 15: IComsat v1.1 (Sim slot)

Features:

- Quad-Band 850/ 900/ 1800/ 1900 MHz
- GPRS multi-slot class 10/8
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
- Class 4 (2 W @850/ 900 MHz)
- Class 1 (1 W @ 1800/1900MHz)
- Control via AT commands (GSM 07.07 ,07.05 and SIMCOM enhanced AT Commands)
- Low power consumption: 1.5mA(sleep mode)
- Operation temperature: -40°C to +85 °C

Installation:

When using the connection as Figure 4, the SIM900 connect to the ATMega328 chip on board.

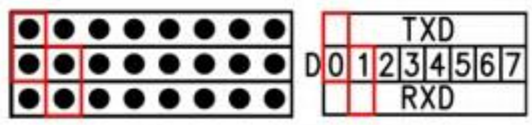


Fig 16: IComsat v1.1 (TX, RX Connection)

Bluetooth (HC05):

In this project HC-05 module has been used which is an easy to use Bluetooth SPP (serial port protocol) designed for transparent wireless serial connection setup. Serial port Bluetooth module is fully qualified Bluetooth v2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4 Ghz radio transceiver and baseband. The purpose of using Bluetooth module on the device is to make connection between taxi meter and our developed Android application for the BRACU taxi meter. In this device, data needs to be received from the taxi meter. So the Tx pin of Bluetooth module is connected to the Arduino pin as Bluetooth data receiver pin. In case of Rx pin Bluetooth module have the capacity limit of receiving 3.3V, where from arduino it gives 5V supply. So to reduce the amount it has voltage divider circuit using two 2k and one 1k resistor and give 3.3V to the Bluetooth receiver. This RX pin of the Bluetooth module is assigned with the TX pin number 11 of the arduino uno board.

Fig 17: Bluetooth module (HC-05)

Hardware Features:

- Typical -80dBn sensitivity.
- Up to +4dBm RF transmits power.
- Low power 1.8V operation, 3.3 to 5V I/O.
- PIO control.
- UART interface with programmable baud rate.
- With integrated antenna.
- With edge connector.

After completing the installation of hardware, the code segment for this Bluetooth module has been set according to the supported Android application code segment. On the coding segment, it has included SendOnlySoftwareSerial.h library and introduced a function to collect the expected data smoothly on the android application. Connecting the Android application of the passenger phone it needs to scan and pair up the HC-05 module by using PINCODE 1234 as a default code.

PIN configuration diagram:

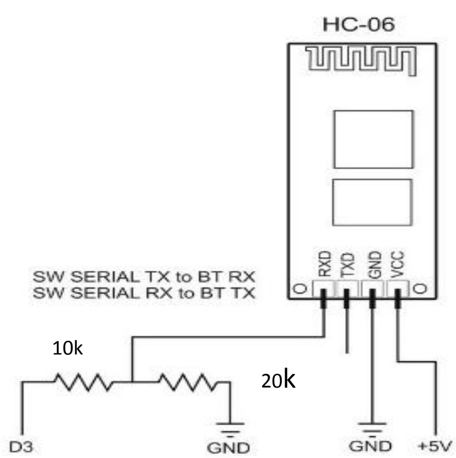


Fig 18: Bluetooth module (HC-05 Pin Connection)

LCD Display (20*4):

To see the amount of fare, distance and time a 20*4 LCD display has been selected as an output device. On the software part it has included LiquidCrystal.h library and assigned LCD display pin with the arduino pin 13,12,7,6,5,4.

Features:

- ❖ Wide viewing angle and high contrast.
- ❖ Industry standard HD44780 equivalent LCD controller built-in.
- ❖ +5V DC LED backlight.
- ❖ Don’t need separate power supply for backlight.
- ❖ Supported 4-line * 20-character.
- ❖ Operate with 5V DC.
- ❖ Free 16 positions male header.

Specifications:

- ❖ module size: 98mm * 60mm * 14mm
- ❖ black metal bezel: 98mm * 40mm
- ❖ viewing area: 76mm * 25.2mm

PIN configuration diagram:

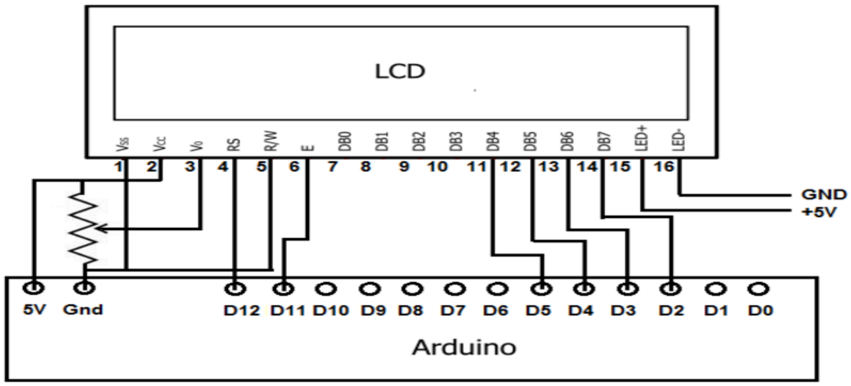


Fig 18: Display Connection diagram

Chapter 5:

Taxi meter Fare Calculation

Calculation of Fare charge is the most vital part regarding to the device mechanism system. Getting accurate fare change by fine calculation of distance, waiting time is one of the essential targets of this project. Fare calculation of taxi meter is depending upon some factors. Such as, getting good satellite coordinates value, waiting time calculation with fine accuracy. Combining all the parts in the coding with the most efficient posture is also a requirement of getting rational fare charge. It is known by all that the GPS system works with the satellite system and extracts satellite coordinates values by connecting satellites with the GPS receiver. GPS receiver need to get precise coordinates value to identify its original position. The latitude and longitude values are keys to calculate the overall distance measurement. GPS receiver requires minimum time to connect with the satellites. Considering this case, the GPS receiver has been given a minimum time by executing few blank cycles at the starting and introduced a hire button. Once the GPS receiver established its connection with satellite the minimum time requirement won't be a fact anymore. By using the hire button the meter device can instantly start the calculation processing and stop it when the passengers complete their travel. Total travelling distance of the passenger can't be obtained by only calculating the starting and finishing point's coordinates. Rather it needs to be calculated every time interval distance and add them up for the overall distance measurement. So the meter device has taken5sec time interval between Lat1, Lon1 and Lat2, Lon2 values so that every small distance can be calculated if the transport changes its directions very rapidly. Summing up all these small distance it obtains the total distance measurement.

For measuring distance there are several formulas available. Among them here Haversine formula has applied to calculate the great-circle between two points- that is, the shortest distance over the earth's surface. Haversine formula remains particularly well-conditioned for numerical computation even at small distance, unlike calculations based on the spherical law of cosines and gives more precise value compared to other formulas.

Haversine formula:

$a = \sin^2(dLat/2) + \cos Lat1 \cdot \cos Lat2 \cdot \sin^2(dLon/2)$

$ab = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{(1-a)})$

$d = R \cdot c$

Here the latitude and longitude values are grabbed by the GPS receiver using TinyGPS.h library code for this GPS module.

In Arduino code:

```
floatToRad = PI / 180.0;

float R = 6371;   // radius earth in Km

floatdLat = ((Lat2-Lat1)/10000) * ToRad;

floatdLon = ((Lon2-Lon1)/10000) * ToRad;

floata = sin(dLat/2) * sin(dLat/2) + cos(Lat1 * ToRad) * cos(Lat2 * ToRad) *   sin(dLon/2) * sin(dLon/2);

floatab = 2 * atan2(sqrt(a), sqrt(1-a));

d = R * ab*1000; // every time interval distance

dis=dis+d; // Total distance by adding small distance
```

5.1 Fare charges in Bangladesh:

As per the law of our Bangladesh Road Authority (BRTA), the government hiked the minimum fare to Tk 40 for the first two kilo meter. They will be charged Tk 12 for each kilometer and TK 2per minute as the waiting charge. Previously, the rate was Tk 25 for the first two kilometers and Tk 7.64 for the next. Waiting charge was Tk1.40 per minute. In our country BRTA follow this fixed charge rate regardless of oblige peak/ off-peak hour charge separately.

In terms of waiting charge, waiting time started as per speed of the vehicle goes down and with a minimum speed rate waiting time has been calculated. By calculating and comparing small interval distance with the waiting time speed rate of existing meter, a small interval distance value has been set for starting waiting time for this meter device. This calculate d value is considered as 14 meter. When small interval distance, d is less than 14 meter waiting time will be started and for the value d is greater than 14 meter; finished waiting time will get started. Thus the waiting fare charge has been calculated by multiplying the imposed charge amount with the time. Finally adding the fixed charge TK40, waiting charge and per kilometer charge after exceeding first two kilometers, the meter gives a reasonable fare charge amount.

Fare charge calculation in Arduino code:

```
void waiting()
{
if(d<14)
  {
if(dis>2000)
    {
  n=(dis-2000)*0.012;
    }
if(c==1)
    {
start = millis();
Serial.println("Started...");
    c=2;
    }
  }
if(d>14)
  {
if(dis>2000)
    {
```

```
  n=(dis-2000)*0.012;
  }
if (c==2)
   {
finished = millis();
delay(200); // for debounce
duration = finished - start;
duration= abs(duration);
elapsed= elapsed + duration;
duration=0;
    c=1;
    }
  }
  h = int(elapsed / 3600000); //hour
over = elapsed % 3600000;
  m = int(over / 60000); // minute
over = over % 60000;
se = int(over / 1000); // second
ms = over % 1000; // millisecond

wfare=h*120+m*2+se*.033; // waiting charge
k=fare+wfare+n; // Total fare charge
```

5.2 Flow chart for Fare calculation:

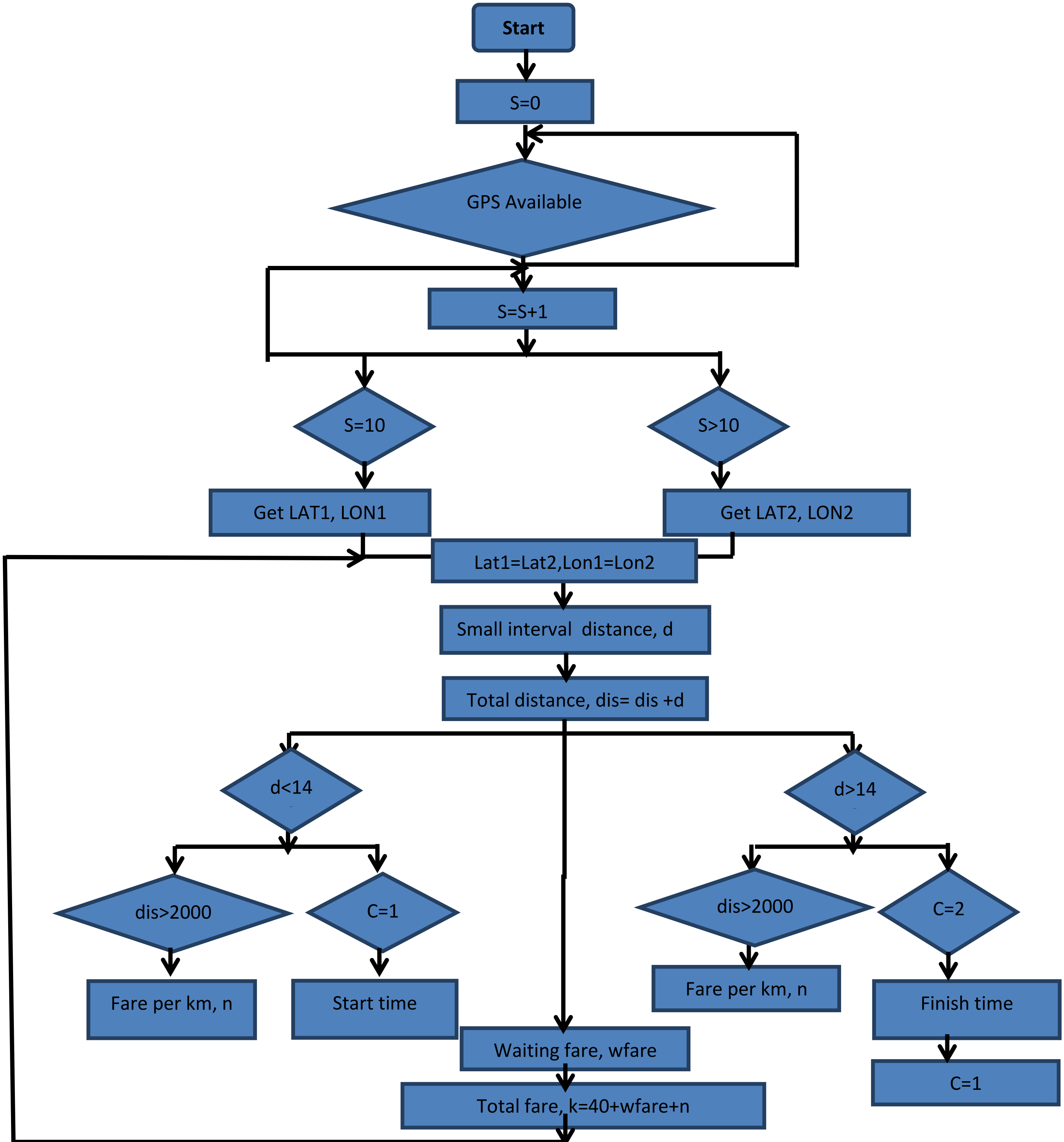


Fig 19: Flow Chart of the system

Chapter 6:

Server Communication

Spark fun data logger:

Data sparkfun provides cost free server facility to push data and store them. It is a free and robust service and open source for all. Here all streams are public and can be accessed by anyone with URL. In this project to store the data information a server has been introduced so that the owner of the vehicle can watch over their taxi positions and take necessary step if any suspicion arises. It will also be beneficial for the security of the country. If any wrong doing or criminal activity places in any area of the country, it can find out the nearby vehicles information for investigation. In data spark fun server each stream has a maximum of 50mb. After hitting the limit, it won't stop storing data but the oldest data will be erased and new data will be installed. The limitations of storing limited data value can be removed if users installed on their own server. Logging is limited to 100 pushes in a 15 minute window. This allows us to push data in bursts, or spread them out over 15 minute windows. This server offered by data spark fun also provide JSON, CSV, MySQL, PostgreSQL, Atom files automatically which gives the facility to use these values in different sectors.

To create a server for the meter device, a new stream titled as GPS Based BRACU BAT Fare Meter has been created in data spark fun. The field names are distance, fare, latitude and longitude according to the values user wants to observe and store in it. By filling up all the information data spark fun provide the public URL and keys for our GPS Based BRACU BAT Fare Meter server site. To connect this server with the meter device and get the data values these keys are the most supreme element of the coding part. The following keys are arranged by data spark fun with the https secure link for the server site- https://data.sparkfun.com/bracu_bat

Public URL

http://data.sparkfun.com/streams/q5G022d1ymcqnD4Gj4Qo

Public Key

q5G022d1ymcqnD4Gj4Qo

Private Key

BVqWEEpbgxSy0Ka5Ga7Y

Delete Key

Rvg3ll5zZ7didJOzENz74

First AT Command for SIM 900 need to be set to get the values from the meter device to the GPS Based BRACU BAT Fare Meter server. Going through the AT command manual, only recommended commands have used in the coding which serves the purpose of this meter device. The following AT commands are carefully selected for the coding segment.

AT+CSQ: Checks the received signal quality in terms of signal strength

AT+CGATT : Allows to attach and detach from GPRS service.

AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\: bearer settings for applications based on IP.

AT+SAPBR=3,1,\"APN\",\"gpinternet\: setting the APN, the second need you fill in your local application server.

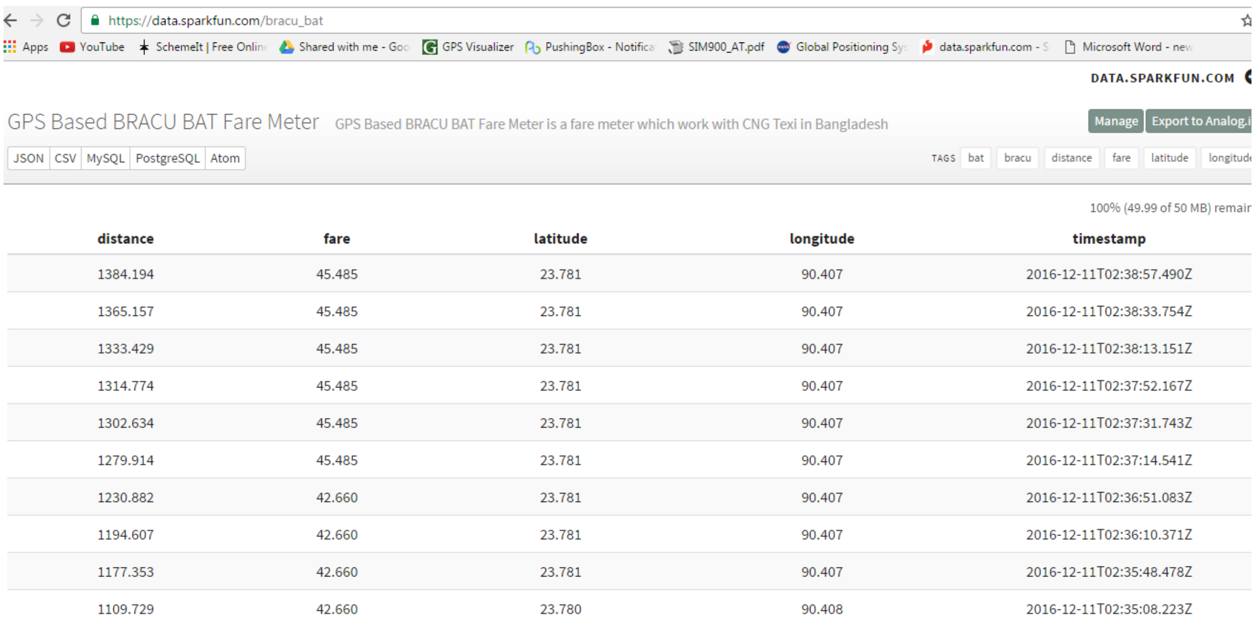
AT+SAPBR=1,1: setting the SAPBR, for detail you can refer to the AT command manual.

AT+HTTPINIT: initialize the HTTP request

AT+HTTPPARA=\"URL\",\"http://data.sparkfun.com/input/q5G022d1ymcqnD4Gj4Qo?private_key=BVqWEEpbxgSy0Ka5Ga7Y&distance=dis&fare=k&latitude=Lat2/1000&longitude=Lon2/1000 : connecting link for the GPS Based BRACU BAT Fare Meter server.

AT+HTTPACTION=1: to submit the request.

Finally the data values start to store in the server link and the owner can access it from the below link- https://data.sparkfun.com/bracu_bat



https://data.sparkfun.com/bracu_bat

DATA.SPARKFUN.COM

GPS Based BRACU BAT Fare Meter GPS Based BRACU BAT Fare Meter is a fare meter which work with CNG Texi in Bangladesh

Manage | Export to Analog.i

JSON | CSV | MySQL | PostgreSQL | Atom

TAGS bat bracu distance fare latitude longitude

100% (49.99 of 50 MB) remain

distance	fare	latitude	longitude	timestamp
1384.194	45.485	23.781	90.407	2016-12-11T02:38:57.490Z
1365.157	45.485	23.781	90.407	2016-12-11T02:38:33.754Z
1333.429	45.485	23.781	90.407	2016-12-11T02:38:13.151Z
1314.774	45.485	23.781	90.407	2016-12-11T02:37:52.167Z
1302.634	45.485	23.781	90.407	2016-12-11T02:37:31.743Z
1279.914	45.485	23.781	90.407	2016-12-11T02:37:14.541Z
1230.882	42.660	23.781	90.407	2016-12-11T02:36:51.083Z
1194.607	42.660	23.781	90.407	2016-12-11T02:36:10.371Z
1177.353	42.660	23.781	90.407	2016-12-11T02:35:48.478Z
1109.729	42.660	23.780	90.408	2016-12-11T02:35:08.223Z

Fig 20: Server Data

Chapter 7:

Mobile App Interface

Android is a mobile operating system. Based on the Linux kernel and designed Google developed it. User interface of Android's is mainly based on direct manipulation, with touch gestures that loosely correspond to real-world actions, such as tapping, swiping and pinching to manipulate on-screen objects. A virtual keyboard is also used for text input. Android application use different sensors to collect different values. The processor manipulates the date depending on the android application code written for the specific work to be done.

Nowadays, every system is getting smarter. People are relying more on digital devices. Every single work is depending on smart technologies. Engineers are designing their product with easy interface for the users. Android operating system smart devices are becoming more popular among the people for cheap pricing along with the smart features. So, engineers are choosing Android for easy interface with the embedded system.

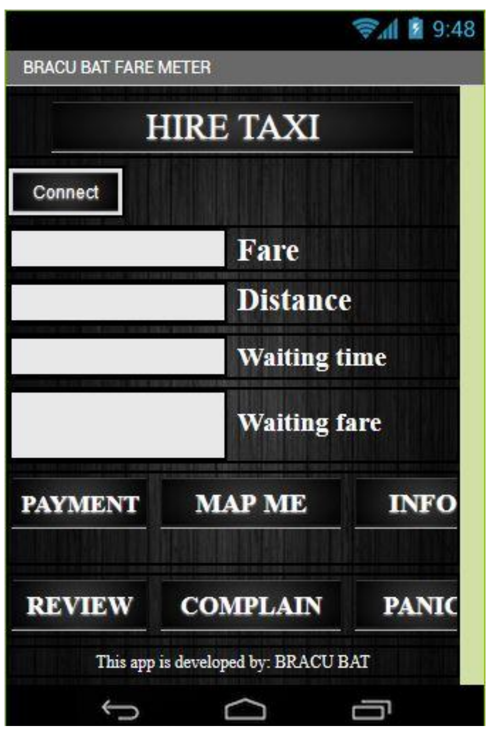


Fig 21: BRACU BAT Fare meter Android Application.

We have developed an android application for our system management. This application connects with our device and creates an easy interface with the users. This application has different features. Those are shown with points below.

Features:

- Fare, Distance, waiting time and waiting fare display
- Payment system
- Showing global mapping position
- Information of the taxi, taxi driver, etc.
- Hire Taxi
- Complain to the authority
- Online review
- Panic Call

7.1 Android development platform:

'MIT App Inventor 2'is chosen for developing this systems android application. App Inventor is an open-source web application originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT) for developing Android application. It uses visual, blocks language for building Android Apps. It allows users to drag-and-drop visual objects to create an application that can run on Android devices. .Here coding is like plugging puzzle pieces together. For broadening participation in computer science it is being used in classrooms all over the world to create software applications for the Android operating system (OS) and it allows newcomers to computer programming. So, it is easy for the engineers to develop the android application for their system and this is why the platform is chosen for this system.

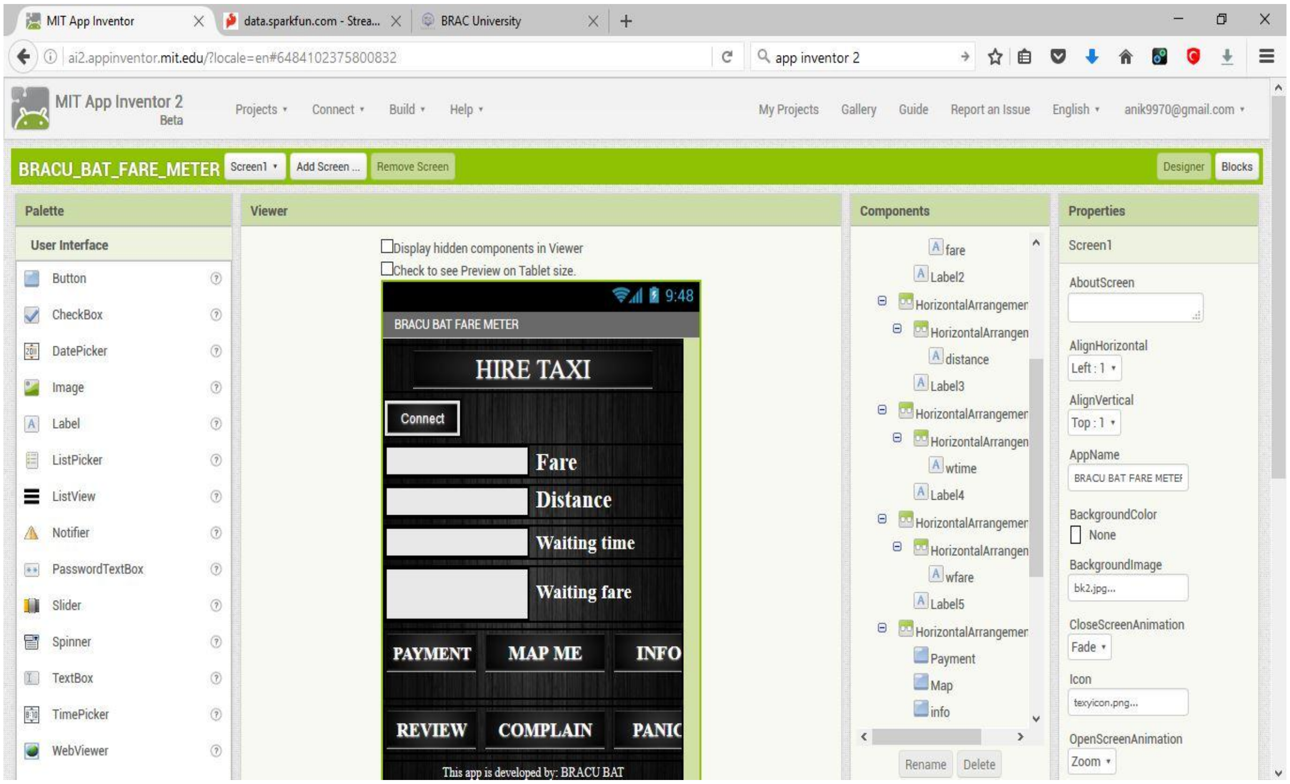


Fig 22: Web page of App inventor (Design).

Firstly, it needs a Google account to create an account in this website. Then it shows the option to run projects. There are two states of a project. One is Design layer where all the graphical designs are made. Another one is block layer where all the codes are written as blocks. Block layer work as the code. This works as the programmer to be run in this system.

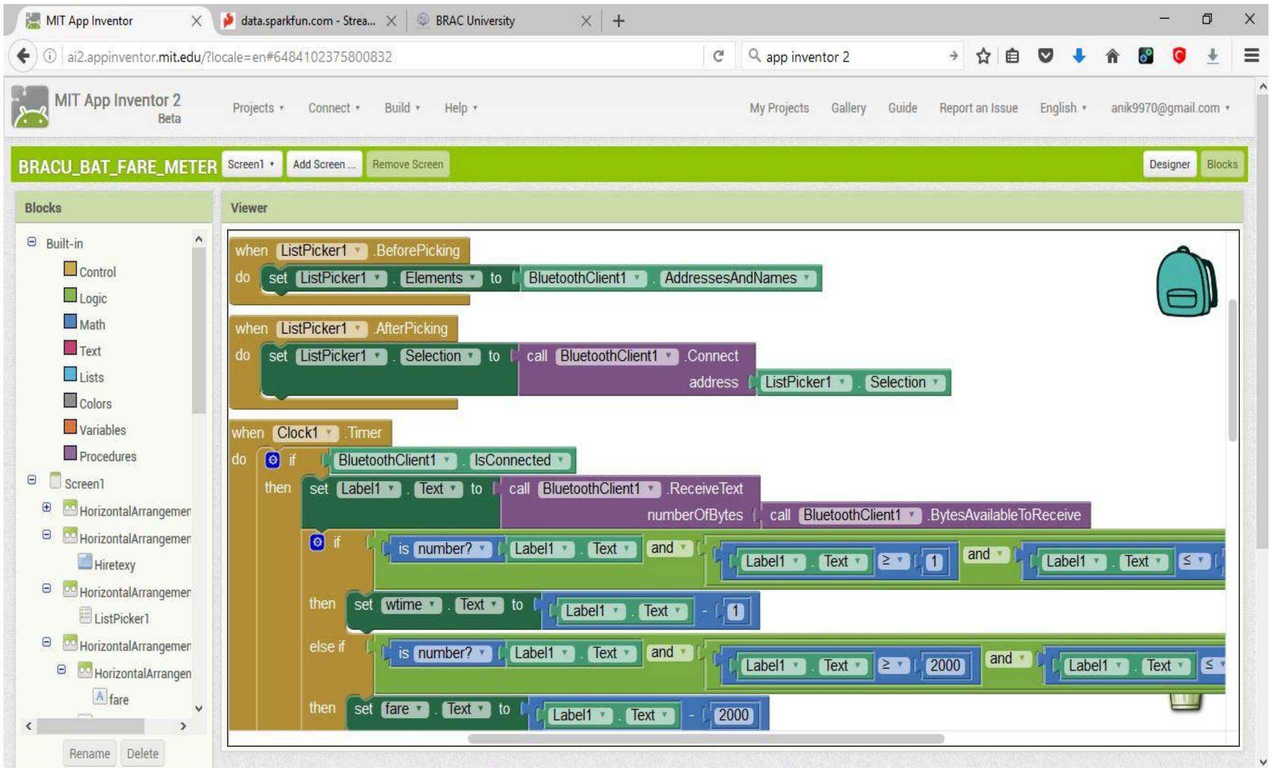


Fig 23: Block coding of the application.

7.2 Application Communication:

This application uses two types of communication system. One is Bluetooth and another one is GSM. Bluetooth communication is used to connect with the Arduino and the GSM Used for calling and GPRS communication with the server and websites. For Bluetooth communication between Arduino and the application a Bluetooth module is connected with the Arduino. The Bluetooth connects with the Android's Bluetooth and communicate between them.

7.3 Fare, Distance, Waiting time and Waiting fare display:

All the data of Fare, Distance, waiting time and waiting fare comes from the processor. The android device connects with the processors Bluetooth module and shares the packets. This system only sends the packs to the android. So, only the Tx pin of processor is connected with the Bluetooth module's Rx pin. The application receives the data as only the packets. It does have that intelligence to differentiate between the variables. So to differentiate between the variables special techniques has been used. The data is mapped between a ranges on the processor and send it through the Bluetooth module. The android receives the data as packets and it again compares the ranges of the data and differentiate between the variables. Then the data is again back mapped to its original value. Then the data is shown in the application correspondence that specific variable. For example, to send the fare data to the android the fare date is mapped between 2000 and 5000. When the android application receives any data between this ranges it understand it is fare and it map it back by subtracting the first range 2000. Then it puts the data correspondence to the fare value in the application and display it.

Example code on processor:

```
float Fare=map(k, 0, 3000, 2000, 5000);

Bluetooth.println(BFare);
```

Example code on Application :

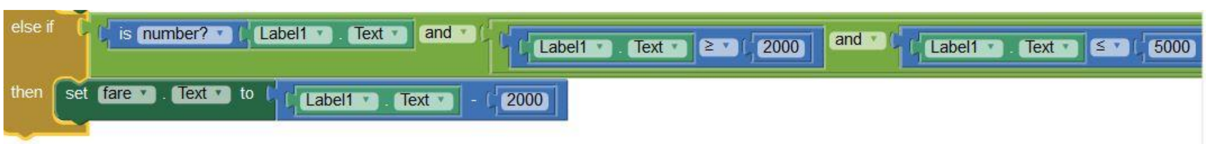


Fig 24: Block code for variables remap

This is how all the variables Fare, Distance, waiting time and waiting fare are shown in the android application. This gives us the real life data visualizing from this system as an interface to the user.

7.4 Payment system:

Nowadays mobile or electric pay become more popular. People always want an easy and secure way of money transaction. So this application accepts the mobile or electric payment system. To know the information of the payment this system uses the QR code system. Where the information are written as the code logo and a QR code scanner used android camera sensor to scan the QR code and show the information. In this system the QR code will be used as a sticker and stacked on passenger cabin. The user can easily scan the code and get the information of the payment. For example, a payment information and a QR code is showed below in the table:

Payment Info	QR Code
BKash: 0173743..... Rocket: 0152149.... Sure Cash: 0168197...... E-pay: 458965....	

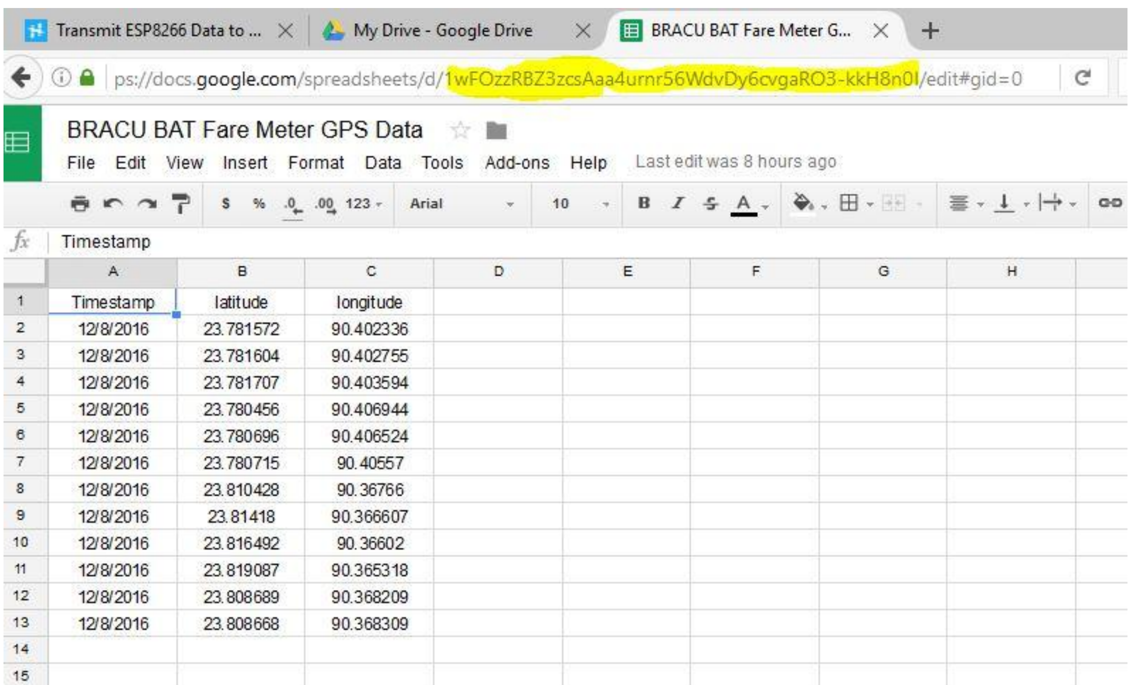
Fig 25: QR Payment information

<u>7.5 Showing positions on global map:</u>

Showing global mapping positions of the taxi cab in the application is another latest technology. This is the first fare meter system which provides the latest tracking system of the vehicle. As this system uses GPS module to get the global position so another tracking system has been implemented here. For showing the position on the map another database server system has been implemented. Where only the latitude and the longitude are saved from the processor. The processor sends the date through SIM900 by GPRS communication. The server that is created here is the more secured database system. This system used google Excel from to save the data. As the google use secure system for their web application so they use 'https' get method. But the processor and the communication module we used is cannot use the 'https' get method. It only can execute the 'http' get method. So, solving the issue of https and http here another web service has been used here. This converts the http request to https request. There are many web services available but the www.pushingbox.com is found easy to work with.

Steps of uploading data to Google sheet from processor:

1. Log in to the Google account and create a new "Google Sheet,". At this point you will need to copy and save your spreadsheets URL key. This key is listed in the URL between the "/d/ " and the "/edit " of your new spreadsheet (see yellow circle).



	A	B	C
1	Timestamp	latitude	longitude
2	12/8/2016	23.781572	90.402336
3	12/8/2016	23.781604	90.402755
4	12/8/2016	23.781707	90.403594
5	12/8/2016	23.780456	90.406944
6	12/8/2016	23.780696	90.406524
7	12/8/2016	23.780715	90.40557
8	12/8/2016	23.810428	90.36766
9	12/8/2016	23.81418	90.366607
10	12/8/2016	23.816492	90.36602
11	12/8/2016	23.819087	90.365318
12	12/8/2016	23.808689	90.368209
13	12/8/2016	23.808668	90.368309

Fig 26: Google Spreadsheets

2. Create a Google App Script (similar to JavaScript) this will process the data and fill-up spreadsheet correctly. To insert our Google App Script, go to the Script Editor from tool option.

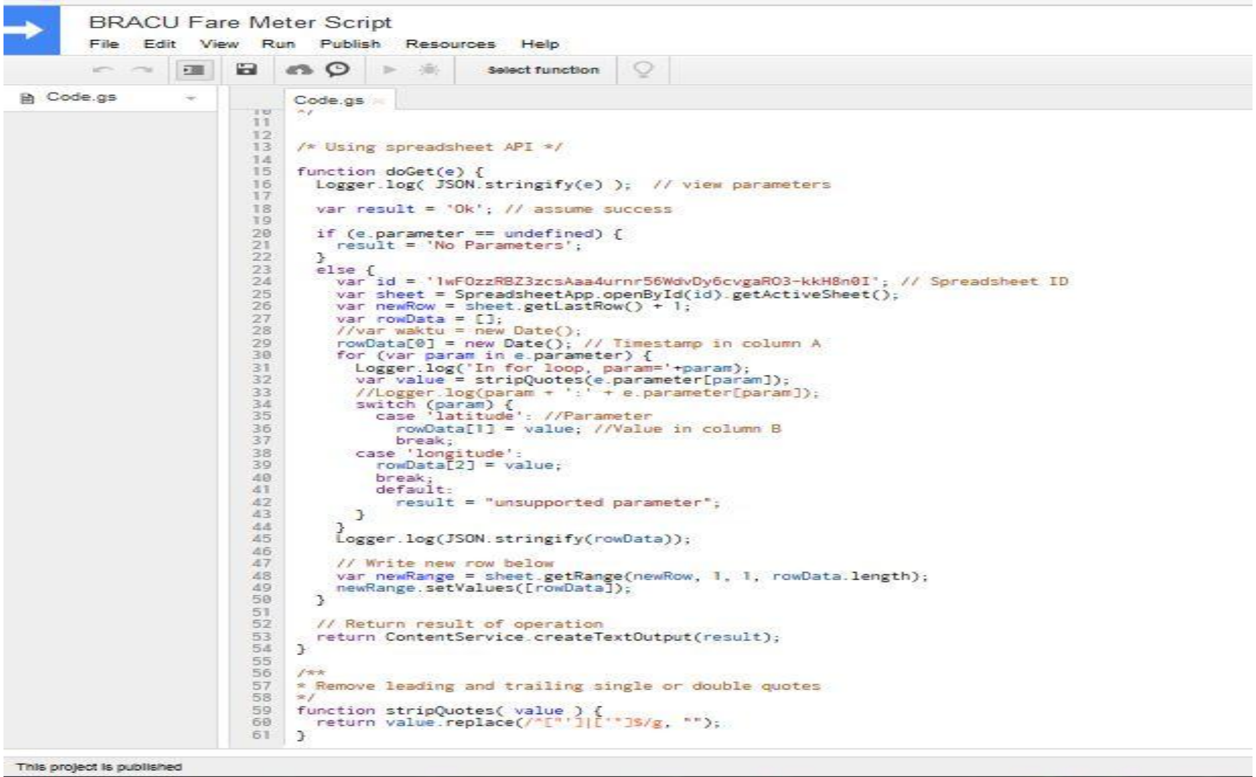


Fig 27: Google script

3. Copy and Paste the above Google Script code into the Editor sheet. Edit the speed sheet id to the specific id field. Save it and publish it as a web app. A pop up screen will appear. Copy the current URL field and save it. This will be needed later.
4. **Pushing Box Configuration**

Pushingbox.com serves as simple, free, and easy API middleman in allowing to upload to Google Sheets. The need for using the Pushing Box API intermediary is to turn our HTTP transmitted data into Google compliant HTTPS encrypted data.

Steps:

a) Create a PushingBox account using your Gmail.
b) At the top click on “My Services”
c) In “My Services” go to the “Add a service” box and click on “Add a service”.

d) The service you will want to create is the last item on the list of services and is called: ”**CustomURL,** Set your own service !". Now select this CustomUrl service.

e) A box will open up and ask for three items. Fill them out as below, it will depend on the customized situation by the user. The saved URL will be used here as Root URL. Then hit submit.

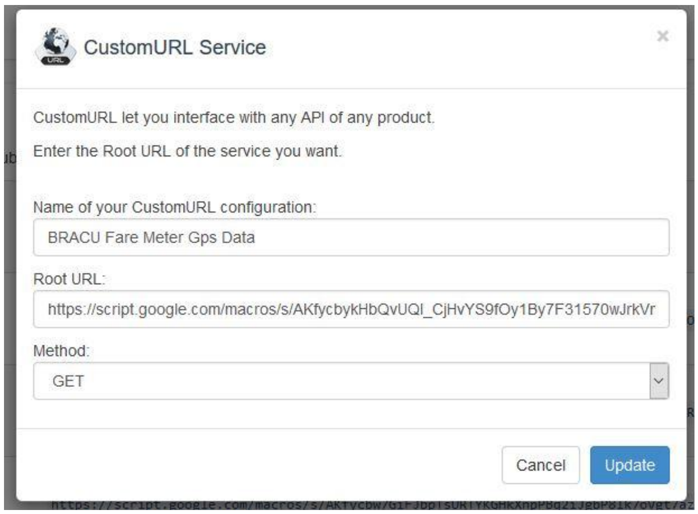


Fig 28: Google Script publishing

f) Create a scenario from “My Scenarios”

g) Now choose the “Add an action to this service” button listed next to the service name created in the previous step.

h) Use GET method and link variable names listed in your Google App Script to the names listed in processor code sketch.

i) Use this type of string into your box written below.(it will depend on the customized variables)

‘?latitude=$latitude$&longitude=$longitude$’

j) Test the api as Clicking

http://api.pushingbox.com/pushingbox?devid=(YOUR-DEVID-HERE)
& latitude=25.66&longitude=77.2

This will upload the data to the specific google sheet by clicking on this. But Pushing box only support 1000 click in a day. It can be updated by premium account by paying services.

Now it is the time for Map positioning. For map positioning this system used a specific website which shows the position on the map. This website takes the data from the Google sheet that has been created in previous steps. The website name is ‘Gpsvisualizer’. Address is ‘http://www.gpsvisualizer.com’. Where this system used the dynamic load mapping option. This system has created a link between the spreadsheet and the website. On this link below:

‘http://www.gpsvisualizer.com/map?dynamic_data=https%3A%2F%2Fdocs.google.com%2Fspreadsheets%2Fd%2F1wFOzzRBZ3zcsAaa4urnr56WdvDy6cvgaRO3-kkH8n0I%2Fpubhtm’

By clicking on the link it goes to a web page, where the website load the data from the google spreadsheet and post it to the html page and shows the position point to point as a red indicator on the google map.

On application this system used a web viewer screen. On application tapping on ‘MAP ME’ option it will direct to another screen for showing the position on the map. It will show the current and previous positions on the screen.

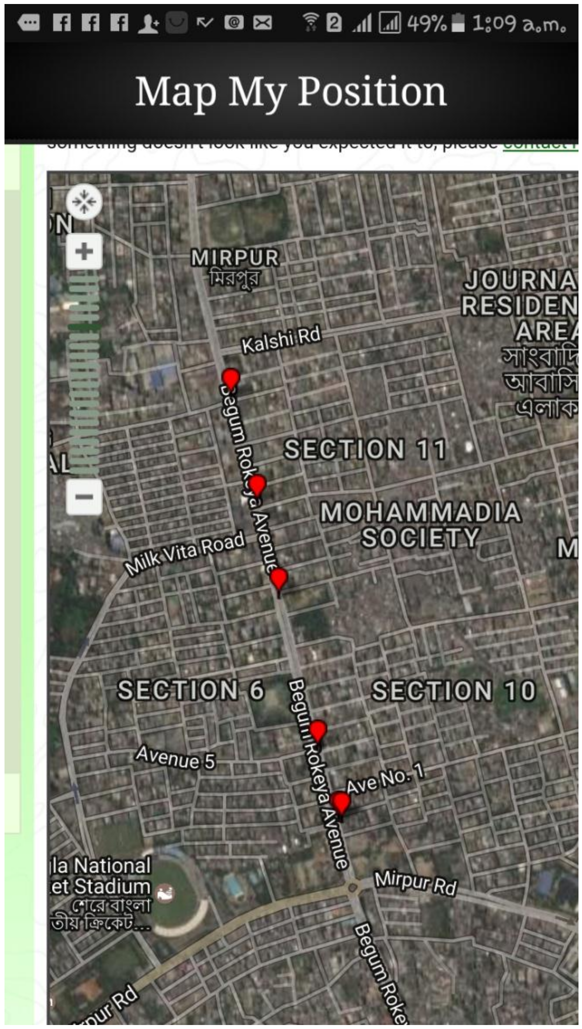


Fig 29: Map on Application

Again, by tapping on ‘Map My position’ it will reload the page and show the updated position on the map from the data of google spreadsheet which are updated from the processor of this fare meter system.

The code of upload data is written on the processor. So, every time when the processor is uploading the values it changes the variable. Below highlighted part is the variable which is uploading in the server.

```
Serial.print("AT+HTTPPARA=\"URL\",\"http://api.pushingbox.com/pushingbox?devid=v6678D5A4926B65E");
Serial.print("&latitude=");
Serial.print( Lat2/10000,6);
Serial.print("&longitude=");
Serial.print(Lon2/10000,6);
Serial.println("\"");
delay(1000);
```

Fig 30: Server uploader code

7.6 Information of the taxi, taxi driver:

Security issue in any taxi cab has become more serious issue nowadays. Robbery, kidnapping and snatching are become common problem. To reduce the problem this application uses several security system. Where getting the Information of the taxi, taxi driver is another solution. From this information can analyze if any occurrence has taken place. We can find the vehicle driver or owner related to that occurrence from the information. This will reduce the security problem issue in great percentage.

To know the information of the taxi, taxi driver this system uses the QR code system. Where the information are written as the code logo and a QR code scanner used android camera sensor to scan the QR code and show the information. In this system the QR code will be used as a sticker and stacked on passenger cabin. The user can easily scan the code and get the information of the taxi and taxi driver. For example, a taxi and taxi driver information and a QR code is showed below in the table:

Taxi and taxi driver information	QR code
Owner: Name: MD. Ismail Hossain. Address: Atapara, Bogra, Bangladesh. NID: 4578697526….. Mobile: 0176443…. **Taxi Reg no:** 128796489 **Driver:** Name: MD. Abirhasan Address: Moddhobadda, Dhaka , Bangladesh. Licence no: 459768….. NID: 1995784684826 Mobile: 0179845….	

Fig 31: Taxi and taxi driver information

7.7 Hire Taxi:

Hiring Taxi is become more difficult these days. The drivers doesn't want to go sometimes, don't follow the meter price, want excess price than fare meter and offer higher price. So, the process is becoming tough day by day as the population increasing along with the increase of fuel price.

If there is a controller service which can select and hire a taxi for the passengers this is going to be a good solution. Where we can call and share our location, then the controller service operator can find the nearby empty taxi cab from the empty taxi locator on the map.

So, this application introduced a Hire Taxi option. By tapping on it will make a phone call to the service provider. So, hiring a taxi cab will become so easy.

7.8 Complain to the authority:

Quality insurance has become the main priority nowadays in every sector in the business. People also give more priority to the quality services. To maintain the quality of the service of this system a complain call canter can be established. Where people can make complains if any quality issue occurs. This will ensure the quality as well as ensure the security of the service can be done.

7.9 Online review:

Blogging system, Online reviews, online marketing has become more popular both for market growth and quality insuring. This system has introduced both online reviews and social media reviews. The authority can take the reviews through the online on Google spreadsheet which will ensure the quality service. Again social media reviews are also added here. This will increase the marketing as well as user will get to know about the services quality.

7.10 Panic Call:

Panic call has been designed for any accidental situation handling. With this button the victim can contact with the nearest police station. By this any unexpected occurrence or accident can be handled. This will reduce percentage of dangerous condition of the accidental victim.

Chapter 8:

Experiment and Result analysis

The whole system is developed under the trial and error practice. There was no exact tutorial on the internet to work with. For developing the system every single steps has been done with the research analysis and real life experiment. The problems have been solved with trials and errors every time.

This system has been developed and experimented in few steps. The steps are explained below.

8.1 GPS Position:

GPS module has been connected with the processor unit. Then the code is written in the software and uploaded in the processor. For getting the position from the GPS Module the latitude and the longitude value has been called. Then the experiment has been done in real life. The device has been checked several times in different positions.

The module gives the accurate position once it is connected perfectly. This module takes around 35s time while connecting with the satellites. Therefore, in the code delay has been implemented to ensure accuracy. The more satellite is connected the more accurate value it shows. Every time it has been checked with another GPS locator from the mobile phone. Both the values has been compared and found the both values are same most of the time. This system has been checked more than 100 times in different position with the mobile GPS and the result is accurate most of the time. Only a little fluctuate happens in both devices which in not more than 1m depending on the position and weather.

LAT 1 :23.773687362670898

LON 1 :90.399574279785156

s :5

LAT 2 :23.773681640625000

LON 2 :90.399574279785156

Distance dlat : -0.000000109083068

Distance dlong : 0.00000000000000

Distance is : 0.694968271255493

Total Distance is : 0.694968271255493

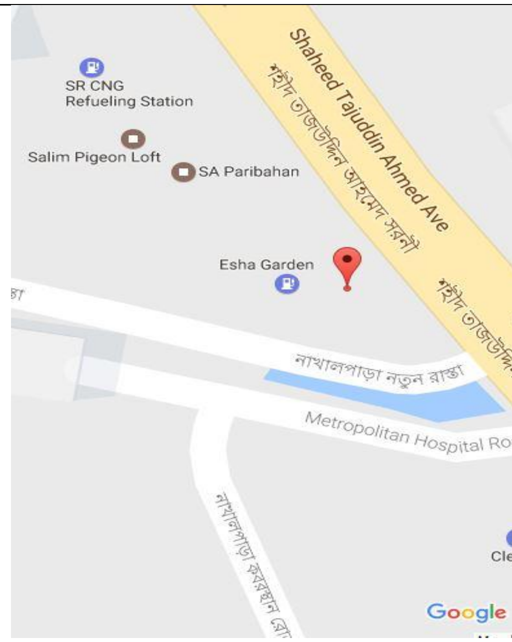


Fig 32: Position checking

8.2 Distance calculation:

When the latitude and the longitude values are available the distance has been calculated in the processor. The positions has been taken on around every 4 seconds. Then it calculates the distance between two positions by adding two positions as a line. Then it repeats the same. On the final count it shows the total distance between the first and the last point on the GPS.

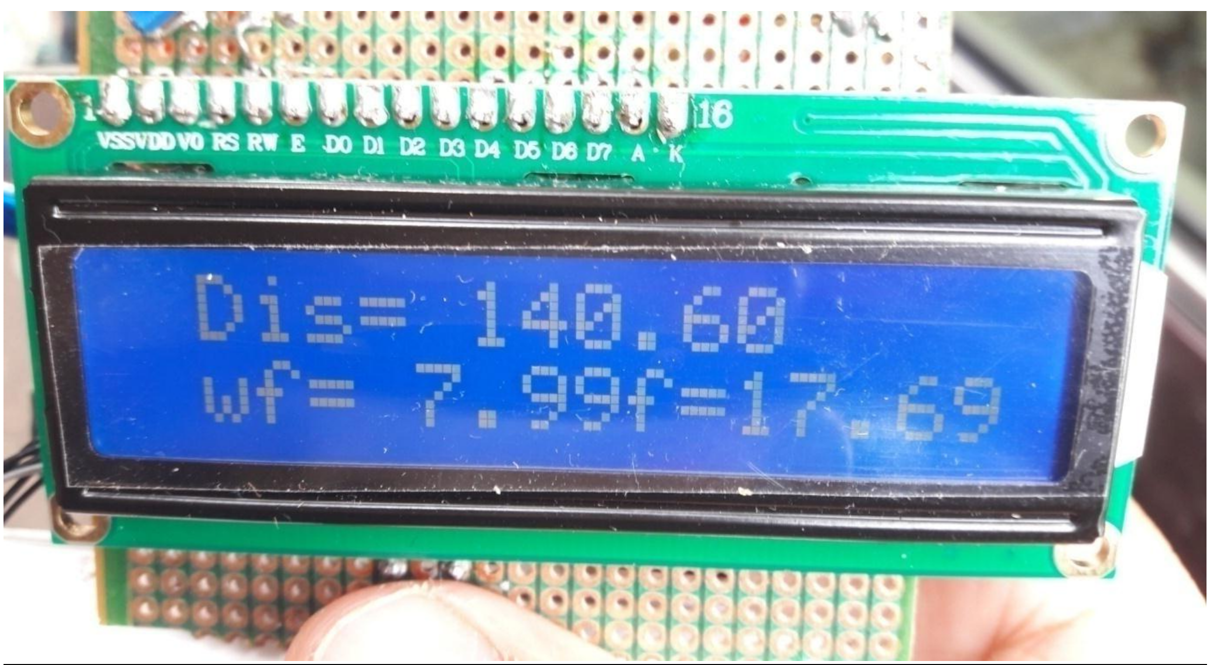


Fig 33: Distance Calculation on the device initially

This system has been checked more than 200 times and solved the accuracy problem by changing the code. The accuracy issue has been solved by depending on the time of interval, algorithm and the values from the module. Time of interval is fine found 3s to 5s between two points as the efficient timing. The time is found by trials and errors checking. Different algorithm has been tried and found the best algorithm as efficient. The values from the module are fractions. So, more than 15 characters have been taken after point.

For Example:

LAT 1: 23.776951

LON 1: 90.398437

LAT 2: 23.776956

LON 2: 90.398437

Distance dlat : 0.000000081812324

Distance dlong: 0.000000000000000

Distance is: 0.521226

Total Distance is: 10.424524

The distance is calculated in three different methods. One is on device, second is on Google map, third one is the mobile application.

Some values of data has been shown:

NO.	**Point 1**	**Point 2**	**BAT Meter km**	**Google Map km**	**Mobile Application km**	**Percentage of change Respect to google map**	**Percentage of change Respect to Mobile Application**
1	23.799154 90.401928	23.793732 90.408606	1.02	1	1.015	-2	-0.49261084
2	23.860697 90.400478	23.780441 90.40.6969	10.95	11.4	11.1	3.947368	1.351351351
3	23.825845 90.425342	23.829436 90.443165	1.805	1.86	1.79	2.956989	-0.83798883
4	23.780463 90.406989	23.773554 90.41548	1.69	1.75	1.72	3.428571	1.744186047
5	23.780463 90.406989	23.827546 90.433418	9.35	9.7	9.44	3.608247	0.953389831
6	23.794949 90.413998	23.804015 90.410732	1.01	1.04	1.02	2.884615	0.980392157
7	23.828305 90.436074	23.860744 90.400278	6.75	7.19	6.9	6.119611	2.173913043
8	23.780441 90.40.6969	23.780687 90.425539	1.93	1.9	1.88	-1.57895	-2.65957447
9	23.7807777 90.416798	23.7946444 90.414462	1.6	1.51	1.63	-5.96026	1.840490798
10	23.780441 90.40.6969	23.753869 90.371446	6.4	6.3	6.45	-1.5873	0.775193798

Distance Comparison between BAT and Google map:

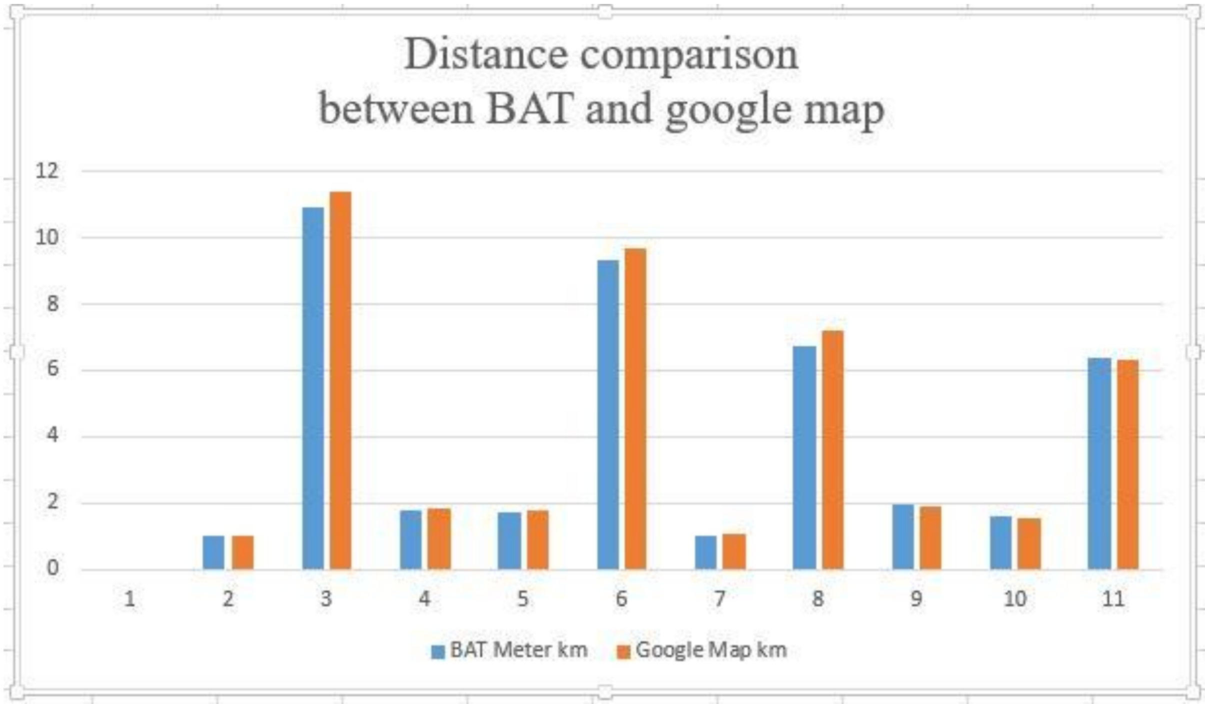


Fig 34: Distance Comparison between BAT and Google map

Distance Comparison between BAT and mobile application:

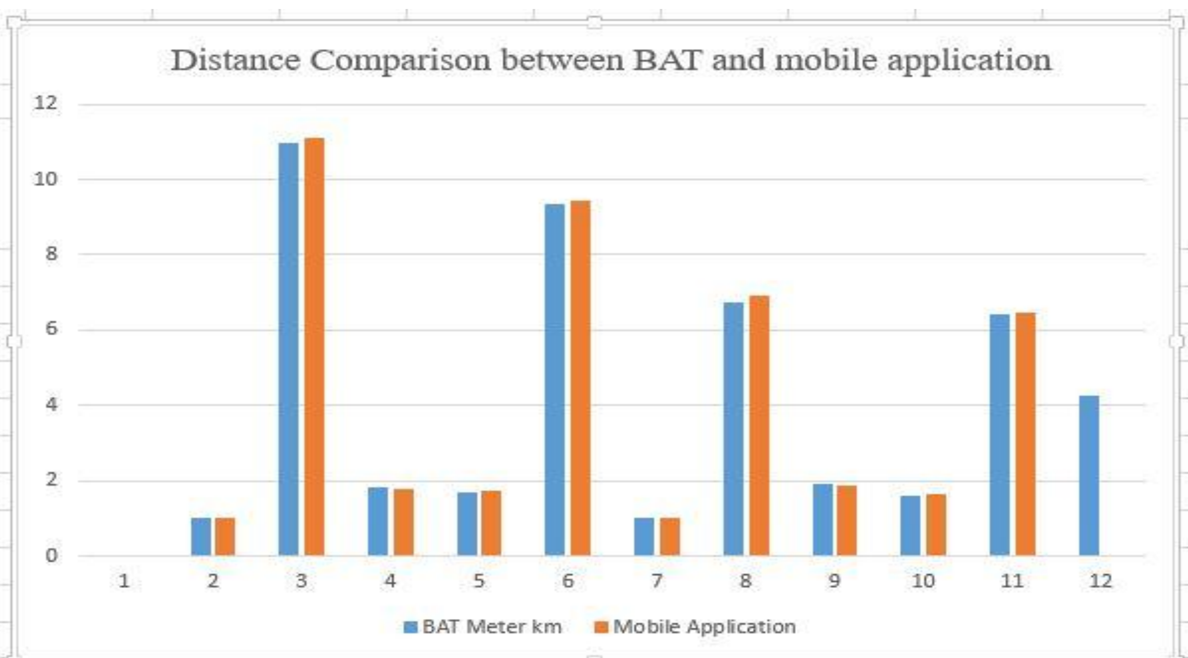


Fig 35: Distance Comparison between BAT and mobile application

Comparison between Average changes:

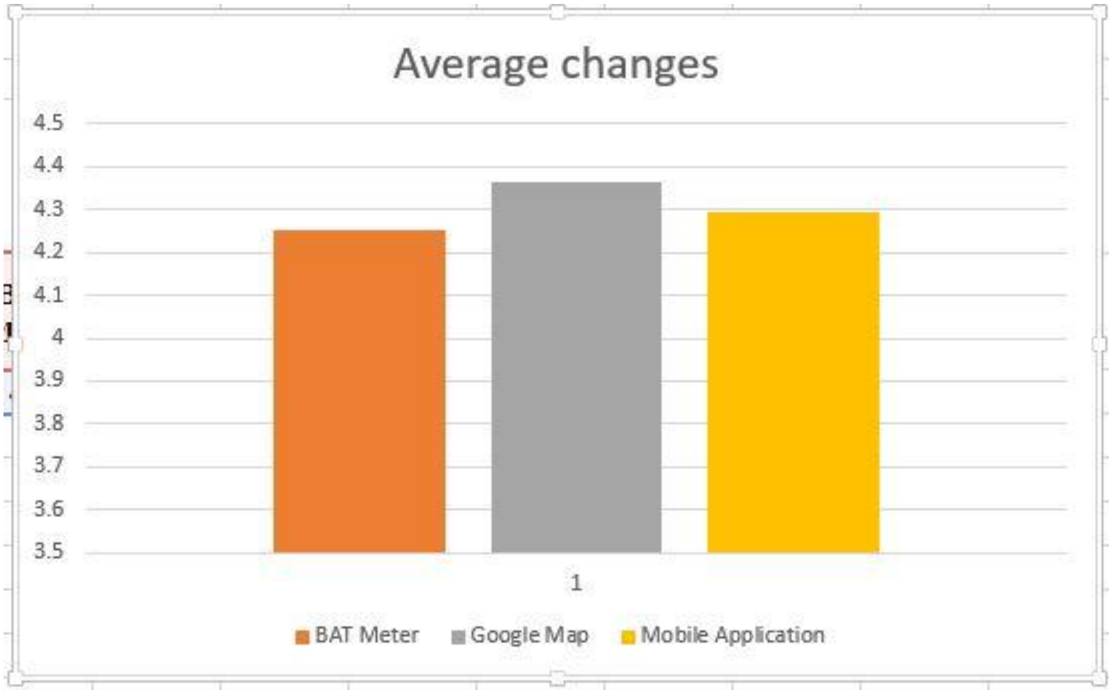


Fig 36: Comparison between Average changes

Comparison between percentages of changes:

- percentages of changes between BAT Meter and Google map

=((Google map-BAT Meter)/ Google map)*100%

- percentages of changes between BAT Meter and Mobile application

=((Mobile application -BAT Meter)/ Mobile application)*100%

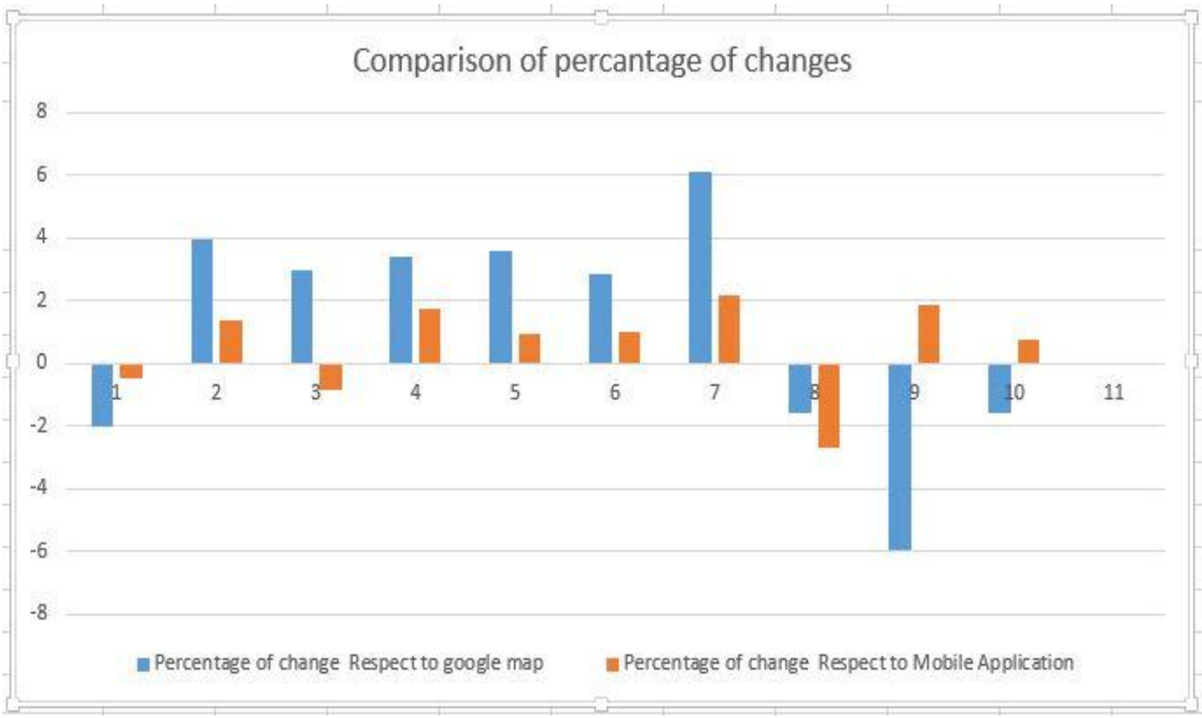


Fig 37: Comparison between percentages changes

Comparison between average percentage:

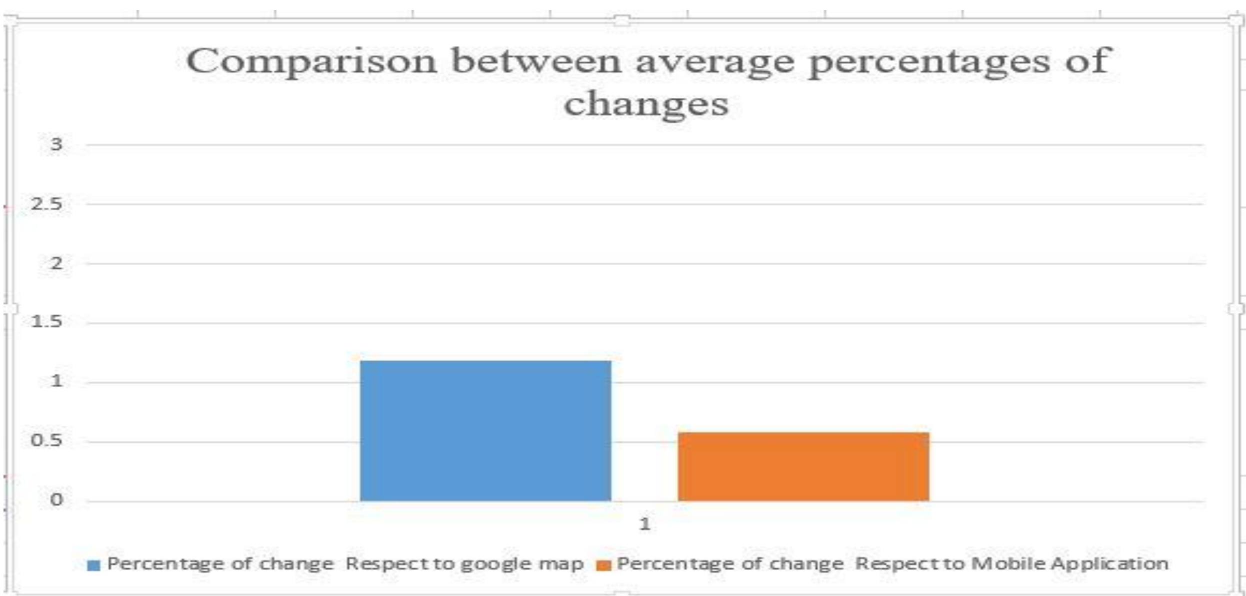


Fig 38: Comparison between average percentages changes

Distance data analysis:

From the data it can see that between BAT and Google map maximum Changes in data is 6.119611%, minimum Changes in data is -5.96026% and average Changes in data is 1.181889%

Again, from the data it can see that between BAT and Google map maximum Changes in data is 2.173913043%, minimum Changes in data is -2.65957447% and average Changes in data is 0.582874289%

From the analysis it can be considered that this system it has an error of +-1.18% average. So the error is very much little negligible. So the acceptance of the distance data is high.

8.3 Waiting time calculation:

There is no particular law in our country when the waiting time starts and stops. The fare meters are used in our country use different methods. Some of them start when the wheel starts and stops when the wheel stops. And again other fare meters use different speed limit. The limit is not particular to all the devices. The devices follow the rules of their manufacturing country. The limit is coded into the processor and they do not share with others. For developing this waiting system many companies was taken interview and they refused to publish it.

So, the real life experiment has been done and found that most of the fare meters in Bangladesh use the speed limit of 10km/h.so, this Waiting time calculation in the taxi is starts when the vehicle is in the speed less than 10km/h. and stops when it is more than 10km/h.

Some of the waiting data has taken in different experiments. Those are shown below:

Number	BAT Waiting Time	BD Fare meter Waiting time	Percentage of changes
1	0:30:01	0:31:12	3.566838
2	0:17:18	0:16:48	-4.24757
3	0:16:32	0:16:17	-0.92764
4	0:25:22	0:24.58	-2.60374
5	0:12:56	0:13.05	3.754789
6	0:37:25	0:38:02	2.02525
7	0:09:15	0:09.27	1.294498
AVG			0.408917

- percentages of Waiting time changes between BAT Meter and BD Fare meter

= ((BD Fare meter -BAT Meter)/ BD Fare meter)*100%

Graphical Waiting time comparison:

Fig 39: Waiting time comparison

Waiting data analysis:

From the data it can see that between BAT and BD fare meter maximum Changes in data is 3.566838%, minimum Changes in data is -4.24757% and average Changes in data is 0.408917%

Finally, from the analysis it can be considered that this system it has an error of +-0.48917% average. So the error is very much little and negligible. So the acceptance of the waiting data is high.

8.4 Fare calculation:

As the fare calculation is totally dependable on the distance and waiting time. So, the accuracy is also done in depending on the distance and the waiting fare covered by the taxi. Fare calculation has been done depending on the fare policy of Bangladesh government in CNG taxi cab.

Though some real life data has been collected from this device and the BD fare meter. Those are shown below:

Number	BAT Distance	BAT Waiting time	BD fare meter distance	BD Waiting time	BAT Fare Taka	BD Fare meter Fare Taka	Percentage of change In Fare
1	10.95	0:30:01	11.1	0:31:12	207.43	211.60	1.970699
2	6.75	0:17:18	6.9	0:16:48	131.60	132.40	0.60423
3	9.35	0:16:32	9.7	0:16:17	161.27	164.97	2.242832
4	1.69	0:25:22	1.75	0:24.58	90.73	89.93	-0.88958
5	6.4	0:12:56	6.3	0:13.05	118.67	117.60	-0.90986

- percentages of Fare changes between BAT Meter and BD Fare meter

  =((BD Fare meter Fare-BAT Meter Fare)/ BD Fare meter Fare)*100%

Graphical Fare Difference:

Fig 40: Graphical Fare Difference

Graphical Percentage of change in Fare:

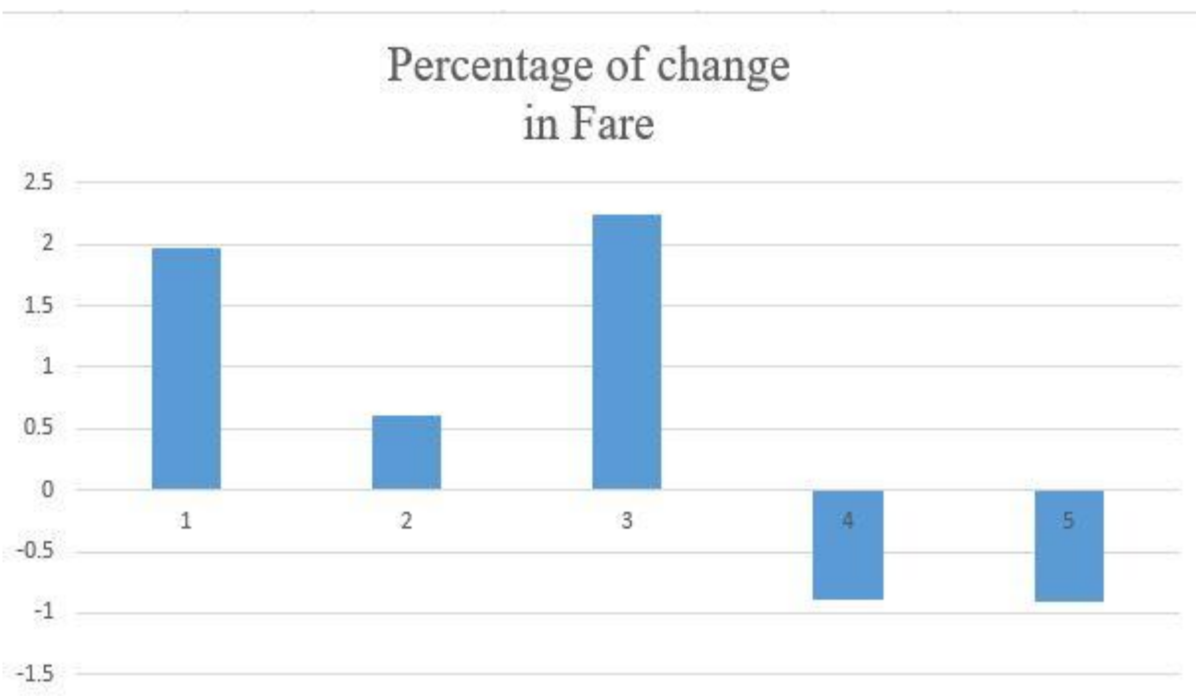


Fig 41: Graphical Percentage of change in Fare

Fare data analysis:

From the data it can see that between BAT and BD fare meter maximum Changes in fare data is2.242832%, minimum Changes in data is 0.603663% and average Changes in data is 0.408917%

Finally, from the analysis it can be considered that this system it has an error of +-0.603663% average in fare calculation. So the error is very much little and negligible. So the acceptance of the fare data is high.

Though, the BD meters are most of them are tempered. So, most of them give wrong values. So the distance has been calculate both with taxi cabs and Google map and taken the best values. Then the fare has been calculated from both the BD meter and manually depending on the fare policy of Bangladesh government in CNG taxi cab.

Chapter 9:

Discussion

It is the blessing of Almighty Allah that we managed to complete our work with all of our goals achieved within limited time period. Our project was to design a very effective taxi meter that would have the capabilities of a modern taxi meter. It could replace all the old taxi meters with tremendous advantages of its own. Even though we finished our work but the device we designed can be further developed with more equipment added. However, things did not work as perfectly as we planned. There were many problems that we did not anticipate. And we had to work hard to come out with solutions.

9.1 Limitation:

Due to the characteristics of different elements of our device our taxi fare meter has some limitations. GPS antenna we used is not a commercial grade but experiment grade element. Thus getting satellite connections at the beginning is very hard. But once connected it keeps continues connections. GPS accuracy is another big issue for our fare meter. GPS accuracy also depends on very sophisticated receivers, which is impossible to fit in our device because of its size. Even though we managed an accuracy of 1 meter to 2.5 meters, which is very good in these types of experiments. Again, we initially planned to change the fare charge rate from our server, but proved to be very difficult task. The processors and modules do not work directly with http protocol. Thus main programming became very complex as we had to inter-connect two websites to change the internet protocol and then send the data to server. In short, this means every time we want to change the fare charge rate we have to disassemble the device then upload the program and reassemble it again.

Power consumption is another issue. For our experiment it was more than enough time to work with single 9V battery. But GPS module along with GSM data upload and display together consumes very large amount of charge. We should not forget a better device consumes more energy but more energy consumption means taxi cabs must have new power plugs to supply the device with power.

Data storing on server has its limitation too, because we are using website made by us it can store data with very little space. But that is also a matter of our research state. For commercial or bigger work it takes proper licensing work to get website with great storage capability.

9.2 Future Work:

The project of making a modern and advanced CNG fare meter is complete within the period of our thesis course. However, the project has been designed in such a way that it can be further developed. The very platform on which the device is build provides opportunity to add many other features. BRACU fare meter is modern enough to eliminate meter forgery and provide reliable service along with accuracy. On the other hand, in future when more features will be required to add it will be easy to add to the BRACU fare meter. Due to shortage of time there were many options that were not added to the BRACU fare meter.

- Digital payment system. BRACU fare meter works with server, where all the information is saved. It is possible to involve payment system to the server. Online payment service providers such as mobile payment companies or banks can be added. Thus passengers may pay directly through their accounts. In today's world cash payment is becoming defunct as payment through account has become famous and secure. The Government may even get the tax money that they could never get using present day meters. Also online payment is to become main method of payment in near future. Thus BRACU fare meter has the potential to keep pace with all possibilities and challenges to come.

- LCD display for showing of map and rout can be added to the new meter. As it will be expensive it may not be installed for long time but there is option to add Basic LCD monitor. Maps and routs can be shown for the driver and passenger. This would make it easy for drivers to know where to go and they do not need to be local to be a good driver. On the other hand, passengers would feel secure and comfortable to see where they are going.

- CNG cabs themselves may become developed and advanced. Just like in Europe where the standard of the taxi is very high. BRACU fare meter can work fine with those requirements. Such as passenger number detection, camera installation, emergency call system etc. These systems can be integrated with our BRACU fare meter very easily.

Fig 42: android app for finding taxi cabs

- Taxi service system can be constructed based on BRACU fare meter system. The server used in the BRACU system can be modified and extended massively to interact with all the taxies using BRACU meters. This means all those taxies and their current positions on street can be shown on the server. The users may see empty CNG cabs through BRACU CNG application on their android devices. The server and android application modification are possible as the BRACU CNG meter device is designed to accept such modifications. Therefore, the meter that is developed can compete all the challenges that may come in future.

Fig 43: taxi server system and user access

- Taxies can be tagged with individual identifications to find them where they are at any moment. Using server and IDs it is possible to identify any specific CNG cab and information about it. This will eliminate CNG related all crimes. Authorities will have upper hand to provide security to all the passengers.

9.3 Conclusion:

The very system of taxi meters and their usage is very old and obsolete in this country. Most of the passengers hardly hire a taxi and check its meter; rather they bargain for a fare charge before they even get into the taxi cab. However, there are enormous amount of taxies all around the country. Because of lack of a better taxi cab meter the passengers become victim of forgery and corruption. It is not acceptable at all in this present era where people seek to pay their bills fairly. The BRACU fare meter can eliminate all the corruption and injustice to taxi fare payments. It works accurately with the modern standards. Some of the very best and recent technologies are used in making of the BRACU fare meter, which makes it acceptable to all aware passengers. It also provides more information and option to the primary users. Thus our designed BRACU fare meter will revolutionize the taxi fare policy.

Chapter 10:

References

1. DESIGN AND FABRICATION OF FARE METER OF TAXICAB USING MICROCONTROLLER, Md. Tofiqul Islam, BUET, Dhaka International Conference on Mechanical Engineering 2005

2. "ICOMSAT GSM / GPRS SIM900 MODULE EXPANSION BOARD SHIELD WITH ANTENNA FOR ARDUINO MEGA." https://www.itead.cc/icomsat.html N.p., 17 Aug. 2012. Web.

3. A_CAVIS. "GPS Shield Hookup Guide." https://learn.sparkfun.com/tutorials/gps- shield-hookup-guide. Web.

4. "Arduino/Genuino Uno." https://www.arduino.cc/en/Main/ArduinoBoardUno, N,.p.n.d. Web.

5. B_E_N. "What is an Arduino." https://learn.sparkfun.com/tutorials/what-is-an-arduino, N,.p.n.d. Web.

6. "GPSShieldv16."https://cdn.sparkfun.com/datasheets/Dev/Arduino/Shields/GPSShield-v16.pdf. N.p., 17 May. 2011. Web.

7. Mason. "GLOBALSAT GPS Module." https://cdn.sparkfun.com/datasheets/GPS/EM506_um.pdf. . N.p., 08 Oct. 2013. Web.

8. "GPS Receiver-EM506." https://www.sparkfun.com/products/12751. (n.d.): n. pag. Web.

9. Patrickccabe. "Fundamentals of GPS guided vehicle." http://letsmakerobots.com/node/19554. May 17, 2010. Web.

10. Stanley. "Distance measuring with Arduino." http://arduinodev.com/arduino-gps-distance-measure/. 21 April, 2012.

11.“DATASHEET BLUETOOTH TO SERIAL PORT MODULE HC05.” http://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf. N.p,.n.d. Web.

12.“LCD-20x4B_Darasheet.”http://site.gravitech.us/MicroResearch/Others/LCD-20x4B/LCD-20x4B_Datasheet.pdf. N.p,.n.d. Web.

13. ‘’Benefits of Database for Embedded Systems.’’ http://www.becom.sk/files/benefits_of_database_for_embedded_systems_feb_2011.pdf .N.p,.n.d.web.

14. Zores.B. ‘‘The Growth of Android in embedded Systems.’’ THE LINUX FOUNDATION TRAINING PUBLICATION. N.p,.n.d.web.

15. Case, Jenn. "*Arduino to Arduino Serial Communication.*" *Arduino to Arduino Serial Communication*. Robotic Controls, 06 Feb. 2013. Web.

16.Veness, chris. “Movable Type Script.” http://www.movable-type.co.uk/scripts/latlong.html. N.p,. 2002-2016. Web.

17. "SIM900 AT Command Manual V1.03." (n.d.): n. pag. Web.

Chapter 11:

Appendix

<u>Code section:</u>

<u>Arduino Source code:</u>

```
//Library:

#include <SoftwareSerial.h>

#include <SendOnlySoftwareSerial.h>

#include <TinyGPS.h>

#include <Time.h>

#include <LiquidCrystal.h>

//functions called:

getgpsLAT(bracu);

getgpsLON(bracu);

calculateDistance();

waiting();

displayResult();

httprequest();

Bluetoothdata();

mapdata();


//Get Lat,Lon values:

while(uart_gps.available())     // While there is data on the RX pin...

  {

    int c = uart_gps.read();    // load the data into a variable...
```

```
 if(bracu.encode(c))      // if there is a new valid sentence...


{
  hireState = digitalRead(hirePin);
  delay(25);
 if (hireState ==HIGH)
 {
  Serial.print("HIRED :");
  Serial.println(hireState);
   if(s==10)
   {
   delay(10000);
   Lat1 = getgpsLAT(bracu);
   Lon1 = getgpsLON(bracu);
   Serial.print("LAT 1 :");
   Serial.println((Lat1/10000),15);
   Serial.print("LON 1 :");
   Serial.println((Lon1/10000),15);
   Serial.println();
   delay(5000);
  }


  else if(s>10)
  {
   Lat2 = getgpsLAT(bracu);
```

```
        Lon2 = getgpsLON(bracu);
        Serial.print("LAT 2 :");
        Serial.println((Lat2/10000),15);
        Serial.print("LON 2 :");
        Serial.println((Lon2/10000),15);
        Serial.println();
        Lat1 = Lat2;
        Lon1 = Lon2;
        s=11;
}
     }
       s=s+1;
      }
   }
  }
}
//distance calculation:
void calculateDistance()
{
  float ToRad = PI / 180.0;
  float R = 6371;   // radius earth in Km
  float dLat = ((Lat2-Lat1)/10000) * ToRad;
  float dLon = ((Lon2-Lon1)/10000) * ToRad;
  float a = sin(dLat/2) * sin(dLat/2) +
  cos(Lat1 * ToRad) * cos(Lat2 * ToRad) *
```

```
  sin(dLon/2) * sin(dLon/2);
  float ab = 2 * atan2(sqrt(a), sqrt(1-a));
  d = R * ab*1000;


  lcd.setCursor(12, 3);
  lcd.print(d,1);


  dis=dis+d;
  Serial.print("Total Distance: ");
  Serial.println(dis,15 );
  lcd.setCursor(10, 2);
  lcd.print(dis,2);
      }
//waiting time calculation:
void waiting()
{
  if(d<14)
  {
    if(dis>2000)
    {
  n=(dis-2000)*0.012;
    }

 if(wtc==1)
    {
```

```
    start = millis();


    Serial.println("Started...");
    wtc=2;
    }
  }
  if(d>14 )
  {
    if(dis>2000)
    {
     n=(dis-2000)*0.012;
    }
    if (wtc==2)
    {
    finished = millis();
    delay(200); // for debounce
    duration = finished - start;
    duration= abs(duration);
    elapsed= elapsed + duration;
    duration=0;
    wtc=1;
    }
  }
    h = int(elapsed / 3600000);
    over = elapsed % 3600000;
```

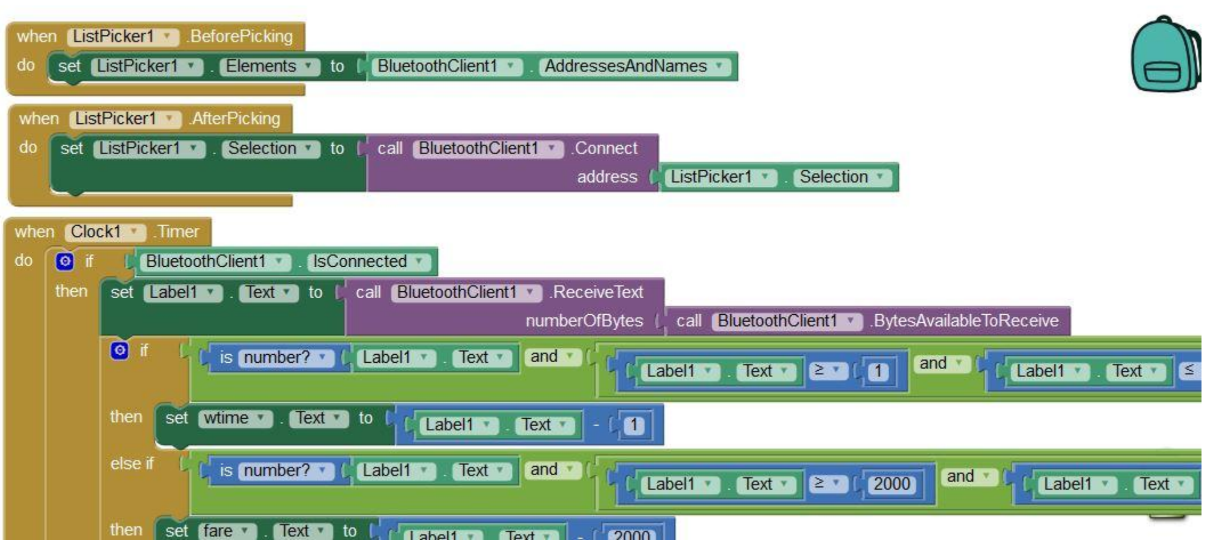
```
    m = int(over / 60000);
    delay(20);
    over = over % 60000;
    se = int(over / 1000);
    ms = over % 1000;
}
//waiting fare calculation:
 wfare=h*120+m*2+se*.033;
  lcd.setCursor(6, 3);
  lcd.print(wfare,2);

  //Fare calculation:
  k=fare+wfare+n;
  Serial.print("total fare:");
  Serial.println(k);
  lcd.setCursor(6,1);
  lcd.print(k,2);
}
//server request :
void httprequest()
{
  delay(1000);
  Serial.println("AT+HTTPINIT"); //init the HTTP request
  delay(1000);
```

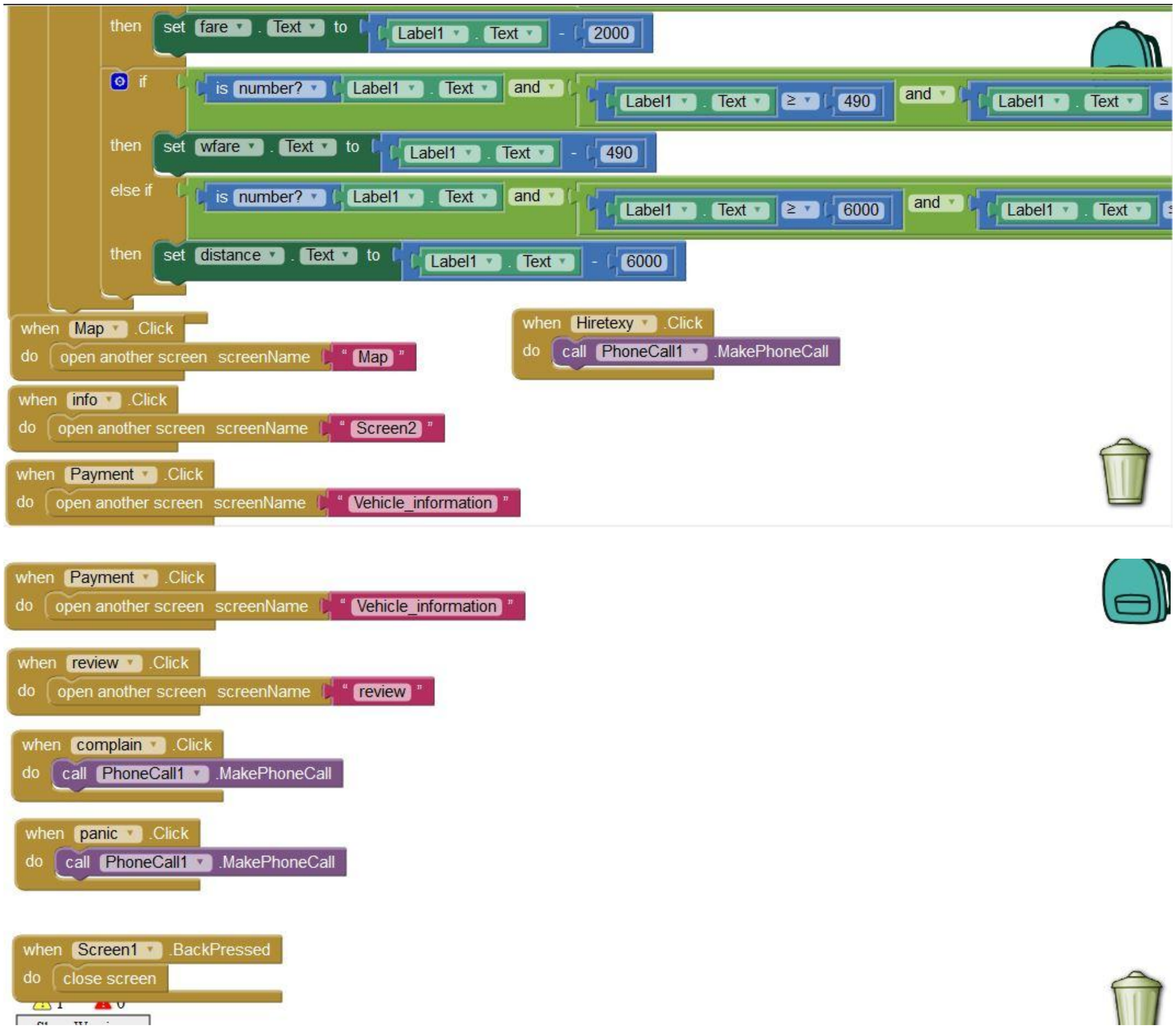
```
Serial.print("AT+HTTPPARA=\"URL\",\"http://data.sparkfun.com/input/q5G022d1ymcqnD4Gj
4Qo?private_key=BVqWEEpbxgSy0Ka5Ga7Y&distance=");

  Serial.print(dis,3);

  Serial.print("&fare=");

  Serial.print(k,3);

  Serial.print("&latitude=");

  Serial.print( Lat2/10000,3);

  Serial.print("&longitude=");

  Serial.print(Lon2/10000,3);

  Serial.println("\"");

  delay(1000);

  Serial.println("AT+HTTPACTION=1");//submit the request

  Serial.println("Data successfully uploaded to Sparkfun server!!!");


    }


//Gps visualizer request

void mapdata()

{

  delay(1000);

  Serial.println("AT+HTTPINIT"); //init the HTTP request

  delay(1000);

Serial.print("AT+HTTPPARA=\"URL\",\"http://api.pushingbox.com/pushingbox?devid=v6678
D5A4926B65E");

  Serial.print("&latitude=");
```

```
  Serial.print( Lat2/10000,6);
  Serial.print("&longitude=");
  Serial.print(Lon2/10000,6);
  Serial.println("\"");
  delay(1000);
  Serial.println("AT+HTTPACTION=1");//submit the request
  Serial.println("Data successfully uploaded to google server!!!");
}

//Android App request
void Bluetoothdata()
{
     delay(100);
     float BFare=map(k, 0, 3000, 2000, 5000);
     Serial.print("BLUE Tooth done ");
     Bluetooth.println(BFare);
     Bluetooth.println();

      delay(100);
      float BDistance =map (dis, 0,100000,6000,106000);
      //Bluetooth.println("BDistance: ");
      Bluetooth.println(BDistance);
      Bluetooth.println();

      delay(100);
```

```
 float BWFare =map (wfare, 0, 1500, 490, 1990);
// Bluetooth.print("BWFare: ");
Bluetooth.println(BWFare);
Bluetooth.println();


delay(100);
 float BWtime= map(m, 0, 480, 1, 481);
   Bluetooth.println(BWtime);
   Bluetooth.println();


}
```

Android Application code:




then set fare . Text to Label1 . Text - 2000
if is number? Label1 . Text and Label1 . Text ≥ 490 and Label1 . Text ≤
then set wfare . Text to Label1 . Text - 490
else if is number? Label1 . Text and Label1 . Text ≥ 6000 and Label1 . Text
then set distance . Text to Label1 . Text - 6000
when Map .Click
do open another screen screenName " Map "
when Hiretexy .Click
do call PhoneCall1 .MakePhoneCall
when info .Click
do open another screen screenName " Screen2 "
when Payment .Click
do open another screen screenName " Vehicle_information "
when Payment .Click
do open another screen screenName " Vehicle_information "
when review .Click
do open another screen screenName " review "
when complain .Click
do call PhoneCall1 .MakePhoneCall
when panic .Click
do call PhoneCall1 .MakePhoneCall
when Screen1 .BackPressed
do close screen