

A Remote Mobile Controlling and Monitoring Vehicle System

Xutao.Liu^{1, a}, Feng.Luo^{2, b}

¹Clean Energy Automotive Engineering Center, School of Automotive Studies,
Tongji University, Shanghai, China

²Clean Energy Automotive Engineering Center, School of Automotive Studies,
Tongji University, Shanghai, China

^abenliuxutao@163.com, ^bluo_feng@tongji.edu.cn

Keywords: self-driving; TCP/IP; Car-mounted terminal; CAN; 3G DUT; windows server; android application;

Abstract. In this paper, we created a remote mobile controlled and monitored vehicle system, which is used in the self-driving and self-parking cars for the last kilometer. The system consists of three main parts: the car-mounted terminal, which is installed in the car for receiving remote orders from the mobile application and transferring information of the car to the application; the windows server, which is run in one computer that is connected to the internet, the server works as an information exchange bridge, which means that it is connected to the mobile application and car-mounted terminal using TCP/IP protocol; the mobile application, which is run in an Android mobile phone. The purpose of the system is to realize completely self-driving^[1] and self-parking by means of giving orders from a mobile phone to the car and in the same time the state and the geographical position of the self-driving car can be monitored on the mobile phone, which is very convenient as the rapid development of the smart mobile phone and the mobile internet. The key technologies of the system include CAN bus communication technology, which is widely used in the car for solving the problems of the increasing automobile electronic communication system; 3G DTU, which is integrated with TCP/IP protocol and supports WCDMA and GPRS mode for obtaining DNS server address and IP address automatically.

Introduction.

As fully self-driving car has been faced so many social and ethical problems, the semi intelligent car has been proposed and developed for a long time, even some of this kind of technologies have been put in use, such as ‘the final kilometre’ project. And one important part of the project is how to make the car to work its intelligent part of job, such as self-parking, and at the same time we human beings do our own business after getting off the car with knowing what the car is doing itself as long as we want to know. In this paper we introduced a solution for the part of the project—a vehicle remote controlling^[2] and monitoring system consisting of a smart android phone, a computer as server and an on-board device.

This paper is organized as follows: section 2 introduce the architecture of the vehicle remote controlling and monitoring system, section 3 present how the on-board device transfer and retrieve vehicle information and remote orders, section 4 present building connection between on-board device and the smart android phone by the server, section 5 present an android App helping the car owner monitoring the car, section 6 communication protocol is presented and the summary is given in the last section.

Architecture.

The system uses C/S structure and consists of three main parts, the on-board device and android app, which work as clients of the system, and the server centre working as server. The Fig. 1 bellow shows the architecture of the system.

One important point of the system is to ensure the communication stable between the two clients, i.e. the smart android phone and the on-board device in vehicle. And as the two clients connected to

internet through wireless mobile network signal, both of them cannot keep using one specific IP address to keep their communication stable, which is not wanted, so here comes the C/S architecture, which would perfectly avoid the problem.

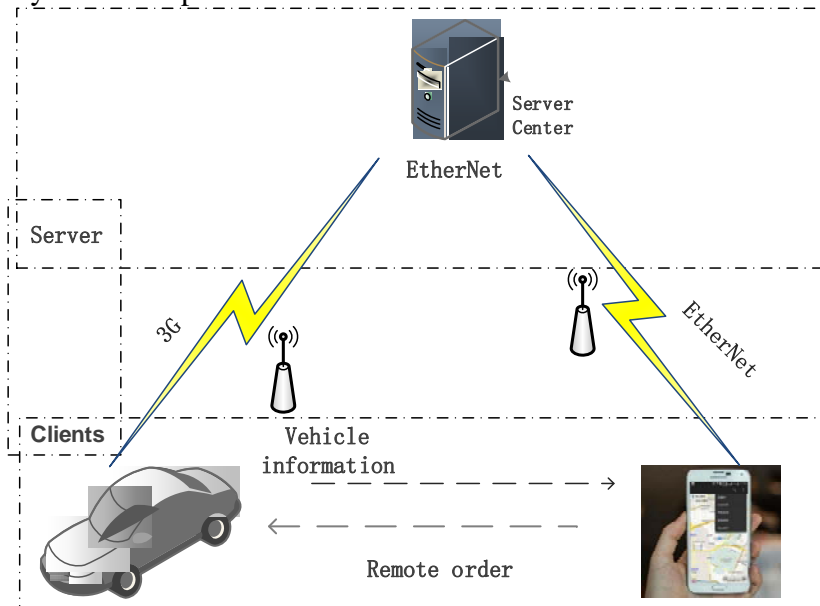


Fig. 1. System architecture

Another advantage by using the C/S architecture for the remote controlling and monitoring system is that it can take advantage of the hardware environment of both sides and rationally allocate the task to server and client to improve the efficiency and have better management of the system if there are hundreds and thousands of clients.

On-board device.

Hardware. The on-board device is installed in the vehicle as a node of the CAN bus and consist of a 3G DTU device, a CAN communication interface and a Freescale MCU as described in Fig. 2.

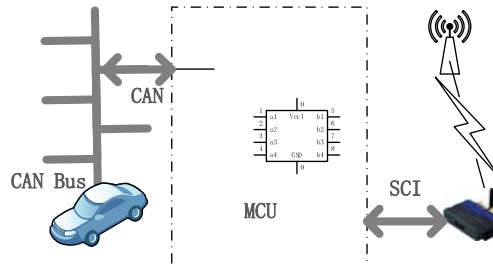


Fig. 2. Hardware

Through the CAN communication interface, all kinds of vehicle information, such as velocity, geographical coordinates, could all be get and transferred through 3G DTU to the server centre and in the same time remote orders received from serial port of 3G DTU can be processed and transferred to the vehicle in the form of CAN frames . And to isolate signal-in and signal-out circuit to suppress the system noise efficiently, either a normal CAN transceiver with digital isolation circuit or a specific CAN transceiver with an embedded isolation circuit needed.

In addition, a stable power supply for the on-board device is also important to ensure a reasonable lifetime of the device. However power supply in a car is usually unstable, so a good DC-DC converter is needed for the device and the power input of the converter should be protected by TVS tubes and relative capacitors.

Software. There are mainly three missions should be completed by the on-board device, i.e. receiving vehicle information, integration of the information and information transfer, receiving and transferring orders. And as part of the device, 3G DTU should trying to build connection with the server center as soon as it's powered and once the connection is built successfully, a respond message should be transferred to both of the on-board device and the server center to acknowledge

both sides that missions can be carried on.

The device mainly get different vehicle information through the CAN bus with different identifiers and then classify the information according to certain standards and fill them to different buffer to be transferred though SCI to the 3G DTU. 3G DTU support duplex communication and would packed the message with certain protocol before it is transferred and the received packed messages will be unpacked on the server side.

On the other hand, order from the remote server [3] side will be received in a timer interruption and would be integrated to a CAN frame to be transferred to the vehicle. The program flow chart is as showed in Fig.3.

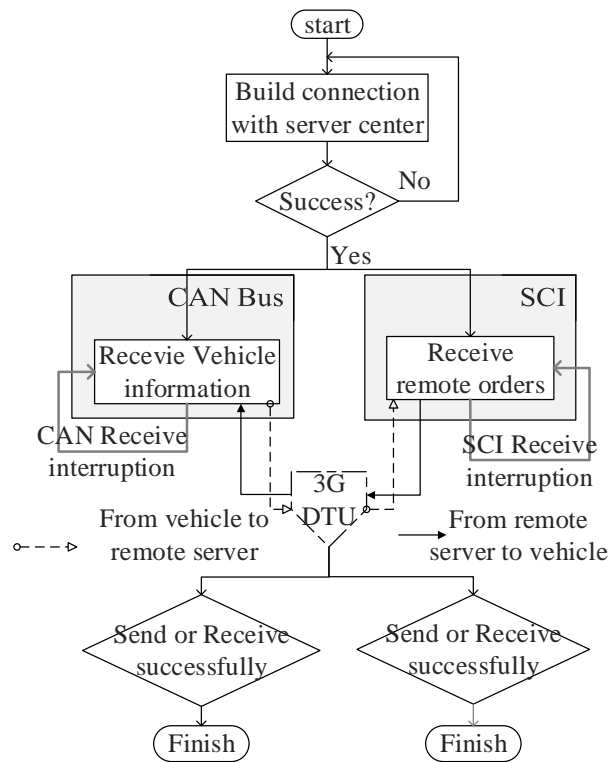


Fig.3. Software

Server center.

The server center is extraordinarily important for the whole remote mobile controlled and monitored system as both the on-board device and the smart phone app rely on it for communication, even part of the data analysis and processing is done in the server centre. In addition, the server is responsible for the user account authentication in order to build one unique relationship between one certain car and its owner. And the server centre should be always on line to ensure the stability of communication between the users and their cars.

In this paper, An Ethernet-connected computer is used as a server and the main processing of data and connection service of the server is put in a windows server to keep running as long as the computer is on-line.

The server, i.e. the server part of the C/S architecture, works in the way of providing three kind of services which seems like of a bridge. The two ends of the 'bridge' represent two kind of services, one of them is the service that works for vehicles in the system. It waits connection request from different vehicles for 24 hours a day and once the connection is built, a connection ID and account is assigned to the vehicle and then transferred its next coming data service request to the intermediate part of the 'bridge', which represent the data processing part of the service centre. The other end of the 'bridge' is designed for the clients consist of Android Apps on different mobile phones. This end of the 'bridge' offer the connection service requested from the apps and according the account offered

by the clients, the next coming data service request is assigned to the intermediate part of the ‘bridge’ and processed according the account identifier.

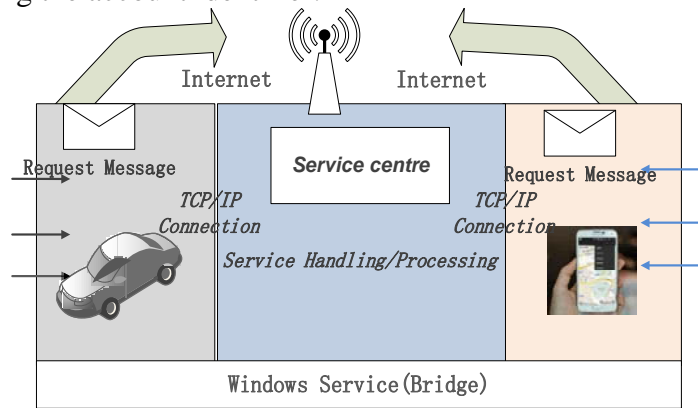


Fig. 4. Service Center

Figure 4 presents how the service centre-‘bridge’ works and the internet connection uses TCP/IP Socket protocol. The basic code structure of the service center is showed in Fig. 5.

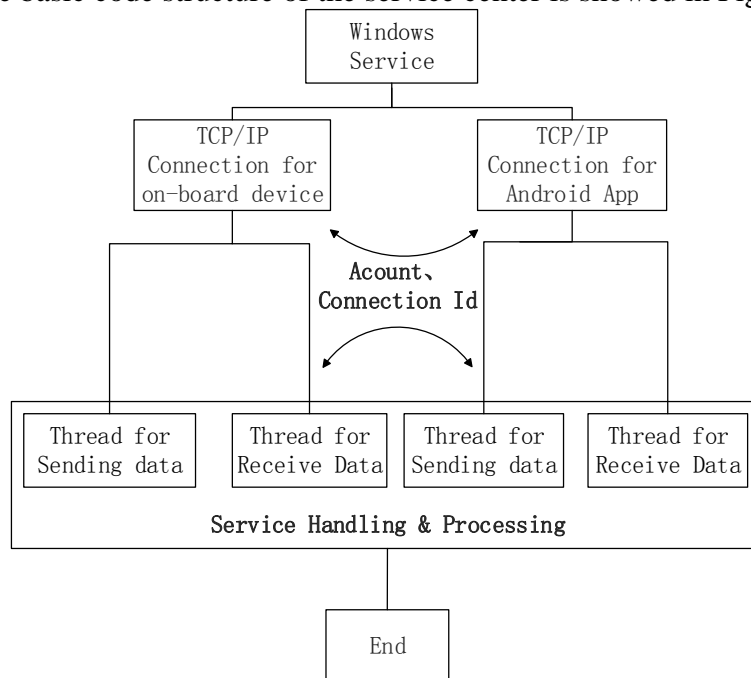


Fig. 5. Code Structure

Android App Design.

Android is an operating system based on Linux’s free and open code and after a few year’s development, its UI design and User Experience are much more nearer to perfect, which is the main reason why Android mobile phone has been used widely since a couple years ago.

Here we designed an android app which includes the function of monitoring vehicle’s state on a baidu map (the vehicle’s geographical position can be viewed dynamically) [4]and an App page which can view the vehicle state parameters such as velocity, battery SOC, charging states and so on. The page design is showed in Fig.6.

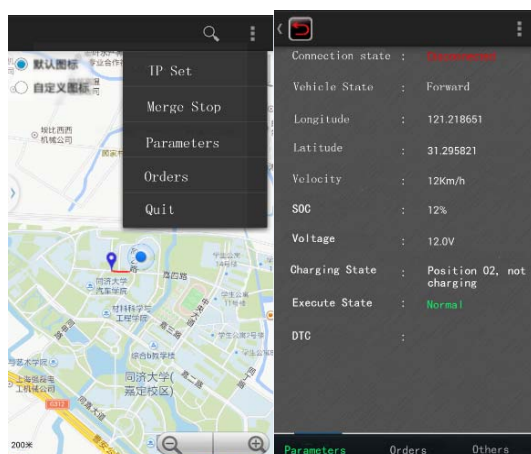


Fig. 6. App Design

Some kind of orders can also be sent by the app to the vehicle, which means duplex communication is available for the vehicle and the vehicle owner.

There is user authentication application in the app and some kind of global configurations can be saved in a file so whenever the app is restarted, a default configurations would be loaded from the file. And the user authentication mechanism ensures the vehicle-to-owner mode of service. The functions that the app include is as follows:

1. Vehicle trajectory tracking, baidu Map API is used for this function.
2. Vehicle information watching, TCP/IP socket protocol is used for requesting certain service from the Server center and the received information are cast to the view page.
3. Orders, the vehicle owner could send some certain orders to the vehicle through the app.
4. Connection, a connection state can be viewed in the view page and when the app is not connected to the server, a reconnect request message is sent automatically until connection is established successfully.

Communication protocol.

In order to ensure the Information transferring efficiency and safety, a set of specific communication protocol should be developed, which can be divided to three part:

1. The communication protocol between on-board device and the vehicle.
2. The communication protocol between the on-board device and the server center.
3. The communication protocol between the Android App and the server center.

As a node of the vehicle CAN bus, the on-board device gains vehicle information mainly from a few of CAN frames with different identifiers, which is defined in the protocol file. The protocol specify the meaning of each byte of the message, here we get four CAN messages with different identifiers, in which three for vehicle data and one for orders from remote controller.

After received the vehicle state data in the form of CAN frame, the on-board device would organize the data to a buffer to be sent to the server center through 3G DUT module according to a specific protocol, which would ensure the server center would receive and process the data correctly. As the protocol defined, a frame head and a frame end needed and the data is stuffed in the middle.

Table 1 Communication Protocol

Frame Head				data			Frame End	
1	2	3	4	5	6	...	n-1	n

In Table 1, byte 1 and byte 2 is the frame head, byte n-1 and byte n is the frame end. And through byte 3 and byte 4, the total length of the frame can be calculated.

$$L = \text{Value (byte 3)} * 256 + \text{Value (byte 4)} + 4$$

After processed in the sever, the data would be transferred to the specific App according the account ID, and the communication protocol above in the table 1 is also applicable to the communication between Server center and Android app.

Conversely, orders from android app can also be transferred to the on-board device based on the agreed communication protocol above. While a specific point should be noted is that, when orders from android app preparing to be transferred to the vehicle in the form of CAN frame, the sending priority level of the frame should be the highest among the information processing programs, which would reduce the time delay of the order transferring to the lowest level.

Conclusion.

This proposal gives a solution of the remote mobile controlled and monitored vehicle system^[5], which can basically fulfill the task of controlling and monitoring the vehicle state by the vehicle owner. The system works in a module of one to one or one to many, which is organized in C/S architecture and quite flexible for users to control one or multiple vehicles.

The solution is not perfect, but at least it would give a relative complete guideline for further research, not only for the remote controlling and monitoring field but for much more internet car applications in the future as the conception of internet car is more and more clear and mature, which means much more closer to coming truth.

References

- [1] Md.Nazmul Hasan, S.M Didar-AI-Alam; Sikder Rezwatul Huq (2011): *Intelligent Car Control for a Smart Car*, International Journal of computer Applications 2011-01.
- [2] Robert N.Charette (2009): This Car Runs on Code, IEEE Spectrum 2015-03-01.
- [3] Alexander Gepperth, Benjamin Dittes, Michael Garcia Ortiz (2012): *The contribution of context information: A case study of object recognition in an intelligent car*, Neurocomputing 94 (2012) 77-86.
- [4] Basnayake C; Mezentsev O; Lachapelle G; Cannon M (2004): *A Portable Vehicular Navigation System Using High Sensitivity GPS Augmented with Inertial Sensors and Map-Matching*, SAE Paper 2004-01-0748.
- [5] Dai Jinbo; Wang Shaokun; Zhao Hongwei; Wang Xu (2012): *The Design and Implementation of the Wireless Remote Control Car*, Department of Computer Science and Technology , Department of Software, Jilin University; ICCEE 2012-11-16.