

Design and Research of Mobile Terminal for Electric Motorcycle Based on Netty

Dong Zhang^{*}, Zhi Weihe, Yu Zeng

College of Electron Information, Hangzhoudianzi University, Hangzhou, China
zhangdonghdu@163.com

Keywords: Netty; Android; Mybatis; TCP; HTTP.

Abstract: In recent years, with the enhancing awareness of energy saving and environment protection, electric motorcycle in the consumer market have become more and more popular. Meanwhile, the rapid development of mobile Internet and Internet of things make the intelligent terminal and intelligent hardware come into our daily life. Nowadays a large number of growth in client requires develop the server which has the performance of strong structure, extensibility, customizable for data transferring and storing. Based on the request of the need, the open source framework named Netty is selected in this paper. Netty provides the support of communication between the mobile phone and the motorcycle. It's the most popular *NIO* frameworks in the industry that has already tested by hundreds of thousands of commercial projects, so that we can control the blue tooth electronic lock, real time location, password modification, authentication on the motorcycle with the help of Android application(app).

1. Introduction

This paper sketches the view based on the intelligent terminal of STM32 which as the core role of the electric motorcycle. The MCU can receive and send the information via the blue tooth communication module and network communication module. The Netty application being ran on the Ali elastic computer service system. The operate system of mobile control terminal is Android.

The main functions of the server application are sending orders to client and receiving the message from clients, Protocol Analysis and storing the data into database. Netty is the framework that support the asynchronous communication model. It has preseted the function of encoding and decoding, and supported many main protocols. Many other frameworks are short of standards and are incompetent in customization and expandability.

The instructions should be sent to control the blue tooth communication module by our intelligent terminal. The mobile phone should be operated to control the motorcycle indirectly through the Netty server. Motorcycle can communicate with server via GPRS communication module circuit. The circuit uploads the location to the server, and the server sends the new password to the motorcycle.

2. Netty Server Application Design

2.1. Create the Netty Server

Netty server was designed with reactor master-slave mode, and the connection which used to receive client's message is an independent *NIO* thread pool. Firstly, the instance should be created by *Server Bootstrap* class, and the configuration should be set up and the thread pool should be bound. There are two independent reactor thread pools, one's name is boss thread pool, other's name is worker thread pool. The former is used to receive the client connection and the latter is

used to handle high numbers of read and write on the socket channel. Secondly, the channel should be bound on the server, and the channel pipeline should be created and initialized. The channel handler should be added and set up. It mainly used for handling network event. For example, the information should be encoded and decoded, the heartbeat should be added and so on. Finally, the port should be listened at the beginning.

2.2. Constitute communication protocols

The server creates port 8000 and port 9000 at the beginning. The motorcycle connect store server via TCP protocol through the port 8000, and app connect store server via the HTTP protocol through the port 9000.

The form of the TCP transport protocol is stream in data transmission. When the server receives the message from MCU, the channel pipeline uses *Delimiter Based Frame Decoder* to decode the message by the separator automatically. It can also avoid sticking and unpacking effectively in the communication process. In order to reduce the stress of the server, the short links should be used between the server and clients. That is, when the client sends the information, and the server returns the information. After ten seconds, the server closes the channel and disconnects from the client automatically. The TCP communication protocols shown in Table 1.

Table1.TCP communication protocols

STM32 send to the Server	#(serial number),(password),(date20170419124933),(current),(volage1), (voltage 2), (voltage3), (voltage 4), (voltage 5), (voltage 6),(temperature1),(temperature2), (distance),(longitude),(latitude), (remaining electricity), (remark), #\$_
The server send to stm32	#(date20170419124955),(GPS flag), (serial number),(password), \r\n

HTTP is an application oriented object-protocol, which is suitable for distributed hypermedia information systems because of its simplicity and speed. When the server receives the message from the app, the channel pipeline uses *Http Server Codec* to decode the message. The server parses the message in the format of JSON. When the server responds to the client, the response message consists of three parts: the state line, the message header, and the response text. All of this information is encapsulated in an instance of *Full Http Response*, and then the message should be returned to the mobile phone .The server usually sends a response code to the client. The response code200is indicates that the client request is successful, 404 indicates that the requested resource does not exist;500 indicates that the server error. At last, the HTTP connection should be closed.

2.3. Selection of thread pool and database connection

The server needs to store the data that received from the port 8000 and port 9000 into the database. This step is time-consuming. So the *new Fixed Thread Pool* should be used to deal with this problem. When the program submits a task, a thread is created, when the number of the client on the connection reaches the maximum that allowed by the thread pool, the thread is stopped creating. The fixed length of the thread pool can be combined with the number of clients to ensure that the server receives and stores information in a normal working state.

Mybatis is a semi-automatic mapping of the framework. It's used to connect to the database. The

mapping relationship between *POJO* and SQL should be provided and matched manually. *Mybatis* is a small package, high performance, optimized, highly active Java persistence layer framework. When *Mybatis* is being used, a session should be created through the *SqlSession* interface, which is the connection resource for the database. After that, the connection should be closed and the resources should be given back to the database connection pool. During the session, the database should be operated by SQL language. Then the relevant properties of the database connection in the configuration file should be configured. Finally the mapper should be constructed. All the process is shown in Fig 1.

Mapping is consisted of Java interface and XML files, and the java interface and the XML file should be given. In this file, a space with the same name to the interface should be defined, and then the database operation should be added in this space. At last, the type should be returned and the data should be set.

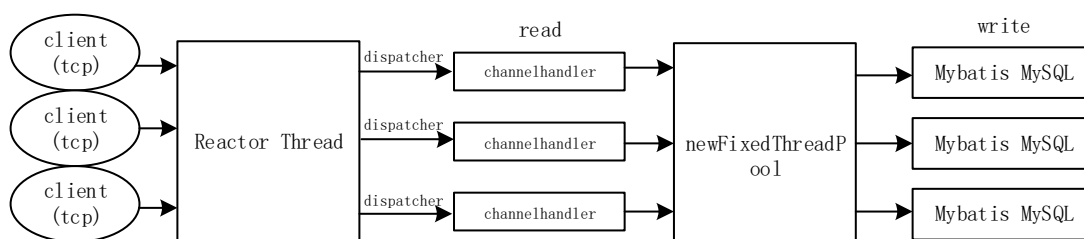


Fig.1 Integrated design of server

2.4. Exception capture.

When the client disconnects from the server, the server always generates an exception that the remote host is forced to close an existing connection. This exception is common, and the main reason is that the client has closed the connection, but the server is still reading the event. Therefore the server reads the client error. In order to solve this problem, the method called *user Event Triggered* and *exception Caught* should be rewritten in the processing of the server's link, in order to carry out exception capture and abnormal printing.

3. Server Performance Test

TCP/UDP performance testing tool is easy for us to test the server compression capability. The sending mode, target address, content, and the number of the clients should be set. After the start of the test, clients send and receive curves that can judge the performance of the server. The server applications should be ran on the Ali elastic computerservice system, and the testing tool applications should be ran on the computer. After the client sends the message, the server searches the message in the database, and then the server returns the message back to the client. Ali elastic computerservice configuration environment is shown in Table 2. At the same time, the tool provides interval and response in two ways.

Table.2 Ali ElasticComputerServiceConfiguration

CPU	2core, General-purpose n1.medium
Memory	4096M
Hard disk	150G
OS	Windows Server 2008 R2 64bit enterprise edition

3.1. Interval mode test result.

Interval mode is the mode that the client submits a request after a certain time interval. The

interval time is set to 50ms. The size of the data sent by the client is greater than the size of the received data. As is shown in theFigure2, the communication maintains about 1000 senders and 1000 receivers. There is no break between the server and the clients.After 20 seconds,the graph of sending and receiving data is much the same as above with no interruption.The volatility is small and stability.

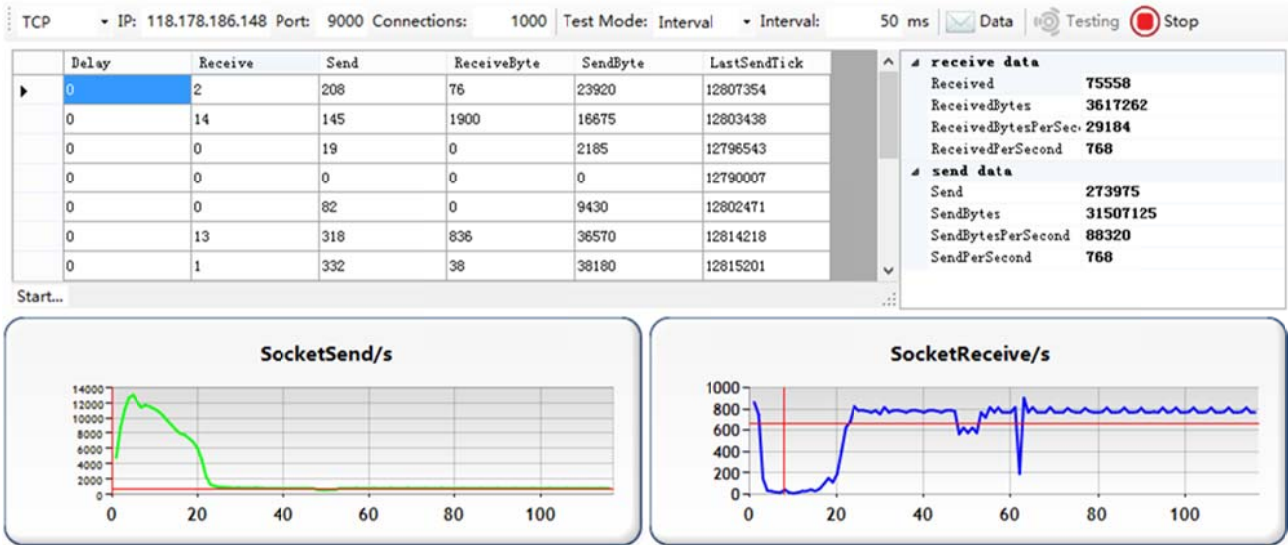


Fig.2 Test result in Interval mode

3.2. Response mode test result

As is shown in theFigure3, the response mode is the mode that the client sends the data to the server immediately after receives the data from the server. Here, when the number of clients is 1000, there are about 1800 senders and receivers. There is no client disconnects from the server and it can work steadily. It can also transmit and received the data accurately. Response mode use a question-and-answer format which uses computer resources more effectively. The number of 1800 transceivers in this model is significantly higher than that in the interval mode.

Based on the comparison of two different result, it proves that the Nettyserver has a very high concurrency performance. It can be connected to thousands of clients.

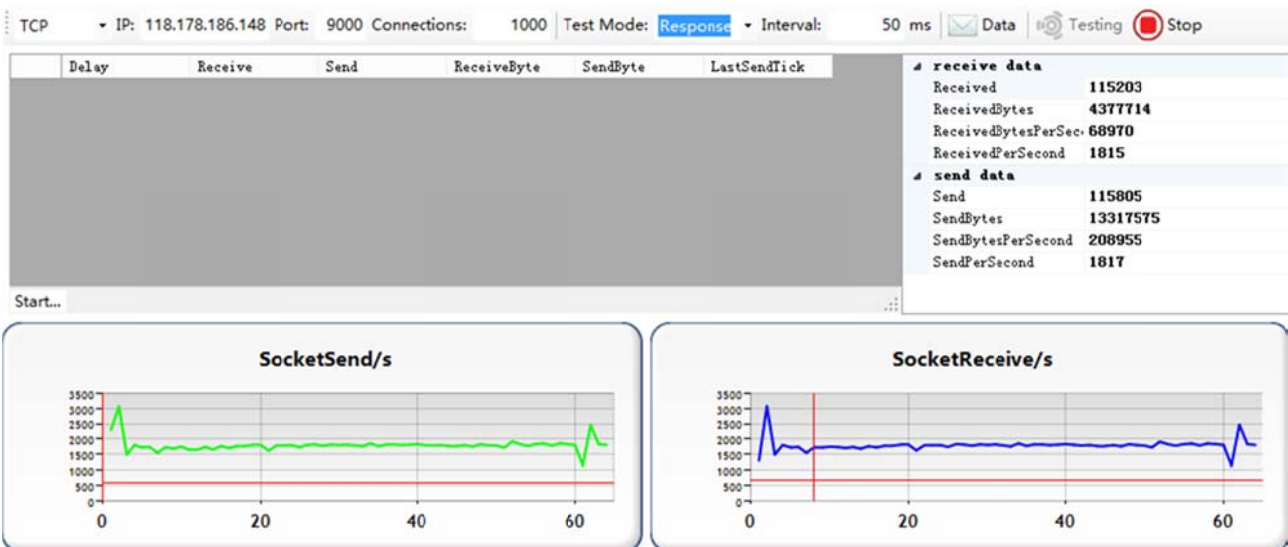


Fig.3 Test result in response mode

4. Android Application Design

4.1. Bluetooth lock function.

Bluetooth4.0 is a low power wireless transmission. Here we use the open source code BluetoothLeClass.java which is provided by Google official. It defines several interfaces and provides several important Bluetooth communication methods.

The app determines whether the phone supports Bluetooth 4.0. Firstly the *mBLE* is instantiated by *BluetoothLeClass* object, and the service callback and data callback operations are performed with the *mBLE*. If the *mBluetoothAdapter* in the *mBLE* is null, it means this phone doesn't support Bluetooth4.0. As is shown in the Figure 4, the flag bit is created which named *mScanning*, if the state of *mScanning* is true, the app's activity jump into the bluetooth list interface. The name and the address of the bluetooth client are added to the *mLeDeviceAdapter*. One client in the *mLeDeviceAdapter* is chosen, then the app finds the bluetooth service and the service's eigenvalue. If the value of UUID equals the value of *UUID_KEY* ("0000ffe-0000-1000-8000-00805f9b34fb"), and then the client can be operated by *gattChar*.

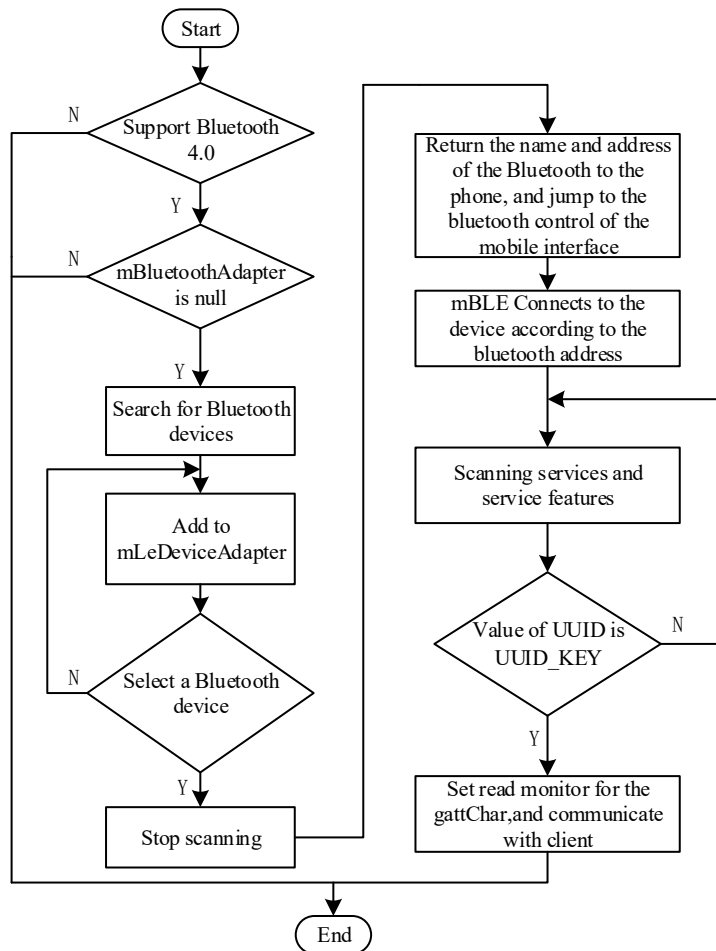


Fig.4 Bluetooth communication flow chart

4.2. Authentication function.

The authentication function is based on the serial number of the motorcycle and the password of the client. In this process, the motorcycle serial number and password should be inputted into the text box, and transmitted the message to the server. The server searches the information according to

the serial number in the database, if the password matched with the data stored in the database, the information returned the validation which is true, and otherwise the validation which is fail.

The whole process of verification is time-consuming. During this time, the mobile phone have to wait for the server's feedback, so asynchronous message processing in the Android system should be used, and the abstract methods provided by the tool for asynchronous message processing with *AsyncTask* toolscan work. As is shown in Fig 5. The password modification function is the same as the authentication function.

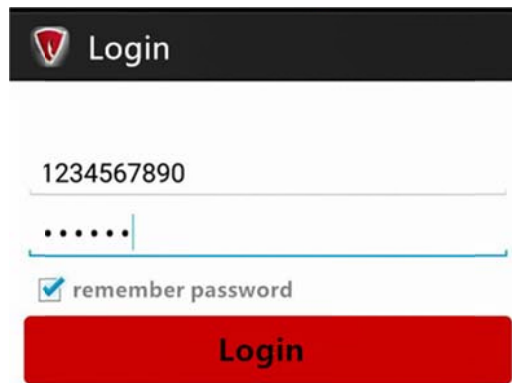


Fig.5 Authentication function chart

4.3. Map location track display function

Baidu map provides a rich interface of API in the field of mobile phone for mobile application developers. Geocoding is the process of converting geographic information into coordinates. Anti-geocoding changes the geographical coordinates into specific address information through the Baidu map coordinate positioning engine. The coordinates of the object corresponding to the administrative divisions, street news can be shown. In our application, when the button is triggered in the location page, app sends an asynchronous request message to the server, the server responds the latest longitude and latitude location information to the app, and the format of the information is JSON. App stores the latitude and longitude location information in the linked list. The track service and the entity listener of the Baidu map should be opened, while creating a track refresh thread to ensure the real-time trajectory of the receiver. The real-time roadmap can be drawn and displayed on the screen.

4. Conclusion

The mobile terminal control of the electric motorcycle and the functions of the Netty server has been designed and implemented in this paper. With using the open source framework to design the concept, the customer needs have been met, and the electric motorcycle becomes more intelligent and networked. Besides, the design uses universal interface, which greatly facilitates the development of new features in the future.

Although the system has already met the needs of users, there are still some deficiencies. For example, the server can't push the message to the app, and app can't show the information about the battery of the motorcycle on the screen. The performance of the server and the app should be improved in the future.

References

[1] Norman Maurer. Netty in Action [M]. 5th Edition. Connecticut: Manning Publications Co., 2013.

- [2]Ying Li Wang,Zhi Jiang Xu,Si Si Li. Based on HTTP Protocol Android Phones to Access the Web Server[J]. Applied Mechanics and Materials, 2014, 3082(543).
- [3]Na Xin Peng,YuQiang Chen. Research and Implementation of Interceptor Based General Physical Pagination Component of Mybatis[J]. Applied Mechanics and Materials, 2014, 2987(513).
- [4]Clarence Ho. Using MyBatis in spring [M].Apress:2012.
- [5]Anonymous. Cacheonix Releases Distributed MyBatisCache [J]. EN, 2011.
- [6]ZhiGuoZhang,MingHao Tang. Research and Implementation of Bluetooth Data Transmission Algorithm Based on Android [J]. Applied Mechanics and Materials, 2013, 2418(325).
- [7]Peter Papcun,IvetaZolotova,KarimTafsi. Control and Teleoperation of Robot Khepera via Android Mobile Device through Bluetooth and WiFi[J]. IFAC PapersOnLine, 2016, 49(25).
- [8]Tie Bin Liu. Design and Implement an Application Development Platform Based on Open Source Technology [J]. Advanced Materials Research, 2014, 3326(989).
- [9]Shi Feng Wu,LuoZhong,Man Li Hu,FanZhou,Hua Zhu Song. Struts2-MVC Based Design of Mobile Supervision System [J]. Advanced Materials Research, 2014, 3593(1049).