

REMOTE AUTOMATIC UPDATE TECHNOLOGY OF THE CLIENT PROGRAM OF THE TAXI MONITORING SYSTEM

Qi Hua^{1,2}, Yu Jun¹

¹School of Civil Engineering, Southwest Jiaotong University, Chengdu, Sichuan, China, 610031

²School of Information Science & Technology, Southwest Jiaotong University, Chengdu, Sichuan, China, 610031 – qi-3dgis@126.com

Commission IV, WG IV/5

KEY WORDS: GIS, Automation, Monitoring, Updating, Hardware, General Packet Radio Service, LPC2214, Embedded System

ABSTRACT:

This article proposes a scheme which realizes the remote update of the dispersed client program through GPRS network by making use of the IAP (In Application Programming) function of the LPC2214 microprocessor. The scheme introduces the IAP function of LPC2214 and the method of using IAP interface, designs a communications protocol, describes and explains the data form of the application layer, and finally describes in detail the realization of the program design, and gives the flow chart of the program. The scheme proposed in this article takes advantage of the IAP function of the LPC2214 processor in the client and the GPRS radio communications technology. Research proves that this scheme is effective to solve the remote update problem of dispersed taxi client.

1. INTRODUCTION

Taxi monitoring navigation system integrates GIS technology, positioning technology, embedded technology and radio communications technology, and is widely used in such fields as intellectual traffic control, vehicle dynamic monitoring and car moving GIS. Such system is composed of the client installed in the taxis and the remote server erected in the monitoring center. The client and monitoring center communicate through GPRS network.

Because of the update of the program edition in the client and the revision of the “bug” of the program, the original client program shall be updated for every a few periods of time. The traditional update scheme is to download and update client program through the interface or JTAG interface of the client in the taxi, using ISP (In System Programming) technology. This method requires the vehicles to go back to the factory or technical personnel to conduct field update in every taxi. However, the taxis are moving in the city randomly, and are widely dispersed. Therefore, the above scheme on the one hand affects the normal operation of the taxis by the method of taxis going back to the factory. On the other hand, the amount of taxis is so large that filed update is not efficient and is easy to make omission.

In view of the above problems, this article proposes a scheme to solve the remote update problem of the dispersed client program by using the IAP function of the LPC2214 processor in the client, and the GPRS remote communications technology.

2. INTRODUCTION OF THE CHARACTERISTICS OF THE LPC2214 PROCESSOR AND IAP FUNCTION

2.1 Introduction of the Characteristics of LPC2214

The microprocessor LPC2214 in the client of the taxi monitoring navigation system is one of the members in the LPC2000 microprocessor of PHILIPS corporation, with a technique of 0.18m and a work frequency of 60MHz (54 Dhrystone MIPs). It is applicable in high performance and low power consumption, and is a general-purpose processor catering to medium and low-end market. It has a 256KB inserted high-speed Flash memory and 16KB RAM, and supports the JTAG, ISP, and IAP programming methods for inserted Flash memory.

2.2 Introduction of IAP Function

IAP, namely applied programming, is a programming mode applied in Flash memory. It memorizes Flash by reading and writing in a program under the control of some certain program. In simple, under the control of applied program, it reads, erases, and writes in some memory space, which makes the program to write in itself, and change its code content by itself. IAP function is one of the bases to realize remote automatic program update. IAP function of LPC2214 makes it convenient to program inner high-speed Flash in applied program, by writing the new applied program downloaded in the outer Flash into the inner high-speed Flash in LPC 2214, covering the old applied program and realizing the update.

2.3 The Usage of the IAP Interface of LPC 2214

The IAP interface provided by LPC 2214 is a series of Thumb code, which is memorized in the Boot sector of the inner Flash. After restoration, the whole Boot sector is cast in the memory location 0x7FFFE000 ~ 0x7FFFFFFF. For IAP (in applied programming), LPC 2214 calls IAP program by having the byte pointer in register R0 pointing to the order code and parameter

in the memory (RAM). The result of the IAP order returns to the memory (RAM) pointed by register R1. The IAP interface can be called in assembler language and C language. In view of the transferability and readability of the program, C language is used in this scheme to call IAP interface.

The following C language code could be used to call the IAP of LPC 2214:

```

    • Define the inlet location of the IAP program.
    # define IAP_LOCATION 0x7FFFFFFF1
    • Define the data structure for transferring IAP order list and results to IAP program.
    unsigned long Command[5];
    unsigned long Result [2];
    • Define the function type pointer. The function contains two parameters and no return value.
    typedef void ( * IAP) ( unsigned int [ ] ,
    unsigned int [ ] );
    IAP iap_entry ; // Define function pointer.
    • Set up function pointer.
    iap_entry = (IAP) IAP_LOCATION ;
    • Call IAP by the following sentence.
    iap_entry (command , result) ;
    • The typical RamToFlash() function calling IAP interface is shown as follows:
    void RamToFlash(uint32 dst, uint32 src, uint32 no)
    { paramin[0] = 51; // Set up order byte
    paramin[1] = dst; // Set up parameters
    paramin[2] = src;
    paramin[3] = no;
    paramin[4] = Fosc/1000; // When PLL function is not used,
    Fcclk=Fosc
    iap_entry(paramin, paramout); // Call IAP service program
    }
    
```

The function of RamToFlash() is to copy the data in RAM in Flash, and the order code is 51. Its inletparameter is: dst means target location, which is the start location of FlashH and is demarcated by 512 bytes; src means source location, which is the location of RAM (the bytes of the location must align); no means the number of bytes copied, which could only be one of 512/1024/4096/8192. Its outlet parameters; IAP return value and it is written in the buffer district of the paramout.

Other functions operating inner Flash could follow the above example, which could be realized by transferring different order codes and parameters to IAP interface. Common order codes are shown in table 1. Attention should be given in the following aspects when IAP function is used: the Flash memory could not be visited when writing or erasing; the space of 32 bytes on the top of inner RAM is occupied when IAP orders are executed, so the user program shall not use this space; PLL and MAM parts shall be cut off before IAP function is called; correct system clock frequency shall be set up when the order of erasing the sector and the order of copying the contents of RAM in Flash are called.

IAP order	Order code
Preparing the programming sector	50
Copying the contents of RAM in Flash	51
Erasing the sector	52
Checking the available space in the sector	53

Reading the ID of the apparatus	54
Reading boot code edition	55
Comparison	56

Table 1. Common IAP Order Codes

3. THE DESIGN OF COMMUNICATIONS PROTOCOL

3.1 The Layers and Structures of the System Communication Protocol

In order to transfer data between the client and server through GPRS, there shall be an internet communications protocol in the client. The bottom to top of the communications protocol in this scheme is respectively: interface drive layer, GPRS module drive layer, PPP protocol layer, IP protocol layer, TCP protocol layer and application layer. The GPRS module used in the system is the TC45 produced by SIEMENS. It internally installs a TCP/IP protocol, which could realize the transparent transfer of data; The protocol in other layers is realized by the codes written in C language. The functions and data form in each layer is realized in accordance with the contents of the protocol. The realization of the top function shall make use of the bottom function and data form. The task of the bottom function is to serve the top function. The cooperation of the protocols at all the layers finally fulfill the task of the application layer, that is transferring data.

3.2 The Design of the Data Form in the Application Layer

In order to transfer data correctly and completely, and to realize the function of transfer resuming at the breakpoints at the same time, this scheme designs the following data form in the application layer.

3.2.1 The Form of the Data Transferred from the Server to the Client

The form of the data transferred from the server to the client is shown in table 2

1 byte	1 byte	1 byte	512 bytes	2 bytes
Cephalon mark	Identification code	Frame number	Data section	Check code

Table 2. The form of the data transferred from the Server to the Client

The data sections are defined as follows:

(1) Cephalon mark: 1 byte. Its value is 01H reflecting the start of the frame 头.

(2) Identification code: 1 byte. The LPC2214 in the client examines it to differentiate data and orders. When it is 00H, those transferred are data, and when it is 01H, those transferred are orders.

(3) Frame number: 1 byte. It represents the number of this frame. When a confirmed frame is transferred, the number of the frame shall add one. The number increases progressively from 1 to 255. After the number reaches 255, it starts again from 1.

(4) Data section: This part is the core of the packet. Its length is 512 bytes. In order to assure the consistency of the size of the data packets, those smaller than 512 bytes shall be supplemented by 0xFF.

(5) Check code: 2 bytes. Before transferring data, the data ready to be transferred shall be operated first to become check sum of 2 bytes. The check sum and data are transferred altogether in a packet, in order to make it convenient for the receiver to check the correctness of the data.

3.2.2 The Form of the Document Information Transferred from the server to the Client

The form of the document information transferred from the server to the client is shown in table 3.

1 byte	1 byte	2 bytes	2 bytes	4 bytes
Cephalon mark	Identification code	Edition number	Total frame amounts of the document	Document size

Table 3 The form of the document information transferred from the Server to the Client

The data sections are defined as follows:

(1) Cephalon mark: 1 byte, 01H. It represents the start of the frame.

(2) Identification code: 1 byte, 10H. It shows that it is the frame transferring documents.

(3) Edition number: 2 bytes. The first byte represents the major edition number, and the second byte represents the supporting edition number. For example, if the two bytes are 01H and 02H respectively, the edition number is 1.2.

(4) Total frame number needed to transfer a document: The program documents are packed in the unit of 512 bytes into data frame before transfer, so the total number of frame tells the client how many data frame to be received.

(5) Total bytes of the document: the size of the document. After the client receives the size of the document, the available space of the inner Flash is checked to see if the document could be loaded.

4. THE REALIZATION OF THE REMOTE UPDATE OF THE CLIENT PROGRAM

4.1 The Overall Design of the Remote Update of the Client Program

The client program consists of two parts: client guidance program and client application program. The client guidance program (which is different from Boot loading program in that the former is designed by the user, and the latter is designed and solidified by the chip producer) is stored in the 0 sector of the inner Flash of LPC214. Its start location is 0x0. Therefore, guidance program is first operated after the system is electrified and restoration. The guidance program completes the initialization of the system, guides the program to move to the inlet of the client application program, and turns the control power to the application program. The client application program is the program realizing the functions of the client, and is the core of the client program. The application program is saved in the unlosable memory in the system. The application program of this system is stored in the inner Flash of LPC2214. It is this part of program that needs to be updated. Guidance program does not need to be updated, because if the update fails, after the LPC2214 restoration, the guidance program could still

guide the start of the system and continue the failed update, preventing the collapse of the system and increasing the reliability of the update program.

The update of the client program requires the cooperation of the guidance program and application program. First of all, the client application program downloads a new application program and saves it in the outer Flash through its downloading module. Then the system restarts, and the client operates the guidance program. The loading module in the guidance program loads the new application in the assigned location in the inner Flash, which replaces the original application program. The update of the client program is completed.

4.2 The Realization of the Design of the Downloading module of the Client Application Program

After the client receives the update order from the server, the downloading module of the client application program is started. First, the client sends a request to the server to obtain the size and edition number of the program documents to be updated. The client program is stored in the inner LPC2214. There are 18 sectors with the size of 256K in the inner Flash of LPC 2214, among which the client guidance program and firmware occupies 2 sectors with the size of 16K. Therefore, the application program shall be no larger than 240K when it is designed. The sector arrangement of the Flash is described in reference [2] in detail. After comparing the document size and the edition number, if the edition number of the document is newer than the present application program, the application program is downloaded, and otherwise the downloading module program is quitted. The system adopts GPRS network to transfer data, therefore, in view of the instability of the radio network, the downloading module has been designed with the function of transfer resuming at breakpoints. Before transferring data, the server shall packet the client program document ready to be transferred into several data frame including data of 512K, and number the data frame in successive sequence. When the client begins downloading, it first checks if the data frame of the new client program has already been downloaded. If so, the next data frame number is sent to the server, telling the server to transfer from this data frame. When the server receives the reply signal from the client, it transfers the data frame according to the data frame number required by the client. The client checks the data frame after receiving it. If the data frame received is correct, the data in the correct data frame is collected and saved in the outer Flash, and the client asks the server to transfer the next data frame; if the data frame received is wrong, the client would ask the server to retransfer this frame. If the correct data frame could not be received for 5 times continuously, the internet connection is regarded as poor, and the downloading program is quitted. The data frame would be downloaded after the internet connection turns better. After the data frame of the client program documents are received correctly and saved in the outer Flash, the program edition number is revised and the update mark is set as "1", for the convenience of informing the guidance program that the new client program has been downloaded and update is needed. The task of the downloading module of the application program is fulfilled. The flow chart of the downloading module is seen in figure 1.

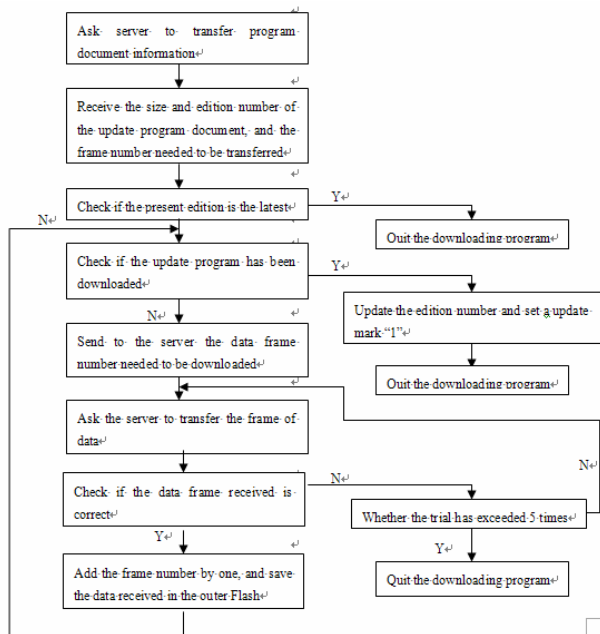


Figure 1. The flow chart of the downloading module of the application program

4.3 The Realization of the Design of the Loading module of the Client Guidance Program

After the client guidance program is electrified and restoration, it starts automatically. The guidance program first initializes the system, and then reads the edition number and download mark of the program, determining whether the update of the client program is needed. If update is not needed, it starts normally, moves to the inlet of the application program to operate the original application program; if update is needed, the loading module is called to load the new client application program. The loading module of the guidance program is realized by the IAP function provided by LPC2214. LPC2214 reads the downloaded application program document in the outer Flash, and calls the IAP interface function to write in location storing the client application program in the inner Flash. After the program is written, and the update process completes, the guidance program moves to the inlet of the application program and starts the new application program, and till now the task of the loading module of the guidance program is fulfilled. The flow chart of the loading module of the guidance program is seen in figure 2.

5. CONCLUDING REMARKS

This scheme has already been practically used in the monitoring system of taxis. After the car client of the system receives the update order, it downloads the program from the server and updates automatically. The vehicles need not to go back to the factory to update, thus not affecting the normal operation of the taxis. The client program is transferred to each client through GPRS network. The working efficiency is greatly improved, and the updating condition of each taxi is recorded in the server. This scheme could also be used in the remote update of the client program of the monitoring system with dispersed clients based on GPRS network.

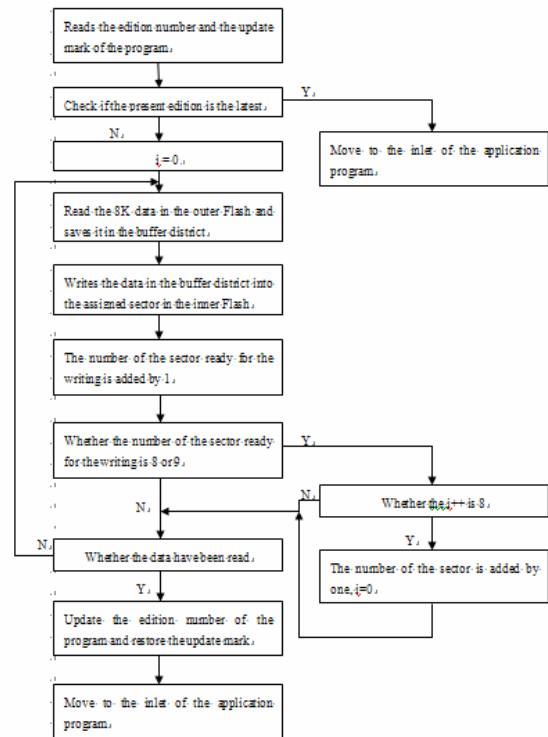


Figure 2. The flow chart of the loading module of the guidance program

REFERENCES

- HU Jingjing, 2005, Wireless Long-distance IAP Function Base on GPRS, Microcontrollers & Embedded Systems, 5(6): pp. 22-25.
- LI Youchi and ZHANG Xianqi, 2005, AT 89S51 Serial Programming and its Read-write for Serial EEPROM, Moletron Communication, 23(3): pp. 30-32.
- LABROSSEJ J., 2003, uC/ OS-II Embedded Real Time Operating System, BEI HANG UNIVERSITY Publishing Company.
- ZHO Ligong and etc., 2005, Foundation of Micro-controller, BEI HANG UNIVERSITY Publishing Company.