Online Journal of Space Communication

Volume 5
Issue 9 *Global Navigation Satellite System (Winter 2006)*

Article 12

Intelligent In-Vehicle Device Based on ARM-Linux Techniques

Xiaoqing Yue

Qishan Zhang

Qing Chang

Intelligent In-Vehicle Device Based on ARM-Linux Techniques

Xiaoqing Yue, Qishan Zhang, Qing Chang

Abstract

A new Intelligent Transportation System (ITS) is introduced and some ideas about instruction trigger position (ITP) are presented. The design of terminal devices based on an embedded ARM-Linux system in vehicles is presented in detail, including the design and development of both hardware and software.

1. Introduction

With the development of a social economy along with the impact of technology, the number of vehicles has increased dramatically. As a result, city road construction and transportation infrastructure can hardly meet transportation requirements. Whether in developed or developing countries, traffic congestion gradually becomes a great social problem as growth occurs. Because of space limitations, the critical issues that should be emphasized include improving road conditions and enhancing facilities to improve transportation management capabilities. The Intelligent Transportation System (ITS) can greatly improve transportation infrastructure usage efficiency and relieve traffic pressure.

2. Patent Solution of ITS

Figure 1 gives a general description of our system design. V_V1-V4 are vehicles with intelligent devices operating under our system. They are driven from Location F to Location E. The available area of ITS is divided into different parts that are identical with base station areas. Communication between the wireless communication module of the intelligent device and the base station will determine the location or area of the vehicle. Along the road are some ITPs that are positioned around traffic signs or set by the drivers (in customized service). The in-vehicle device gets its current location information via GPS satellite technology with the real time location being compared to data output of the nearby ITPs. Additional operations are triggered when the vehicle enters the effective range of another ITP, continuing the monitoring process. All the vehicle information is transmitted to the Transportation Management Center (TMC) that has two subsystems to help deal with the vehicle and transportation data. The TMC can get all the vehicles' information, (i.e. latitude, longitude, speed, direction, electronic license and customized information) by send sampling commands to the remote technology devices. The drivers can also require guidance service by using in-vehicle communication devices. All the navigation work can be done by the TMC with the cooperation of its two subsystems. Digital mapping is included in the system described above because of the strong support of the embedded system. For proprietary reasons, detailed analysis about this system will not be presented.

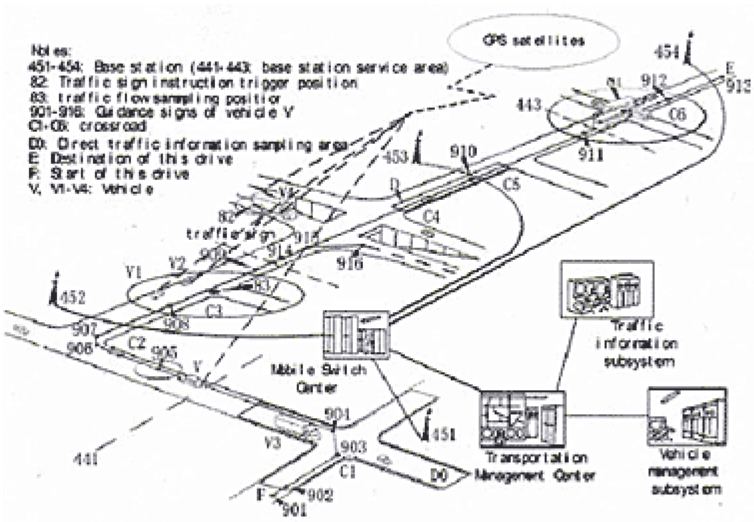

Fig. 1: System Design of ITS

Among the wide applications of ITS, location and navigation of mobile vehicles are two of the primary areas of focus. To most drivers the most important functions of an ITS are integrated information services such as road and nearby environment information, real time road status, route guiding, dynamic route navigation and related facilities data. All the functions are based on the in-vehicle intelligent device which provides support both directly and indirectly to the integrated system.

3. Intelligent Device Based on ARM-LINUX System

3.1 Design Goals

The in-vehicle intelligent device is a complex system composed of multiple sub-systems. Among its extensive application fields, each design has its specific degree of complexity according to the system function requirements and design principles. In this subject, the basic design goals are listed as follows:

Location

At least one location method supported by satellites is utilized to get the real time position information (original data of map coordinates), which can help to confirm specific vehicle location and provide the most effective transportation route available.

Map Display

A detailed digital map of the current area is included and is updated by inserting cards or downloading data from the traffic center.

ITP check and response

Matches the vehicle coordinates with the nearby traffic signs and responds correspondingly.

Audio Function

The intelligent device includes information broadcasts and gives audio instructions when the intelligent device is triggered into the instruction mode.

Wireless Communication

A wireless communication module or similar communication interface is included to communicate with the TMC or similar type of navigation center.

Human-Machine Conversation Interface

Some interactive interfaces dealing with local information such as keyboards, infrared controllers, or a touch display should also be provided.

Advanced Interface

Some interface options are also being considered for further development of the current system (i.e. enhancement of the black box).

Currently, there are many application problems associated with intelligent vehicle devices in the Chinese market (i.e. vulnerable in terms of reliability, dependence on a general GIS platform, complex hardware, complex operation, system breakdowns). These types of challenges can oftentimes lead to incorrect use of the system and result in high development costs.

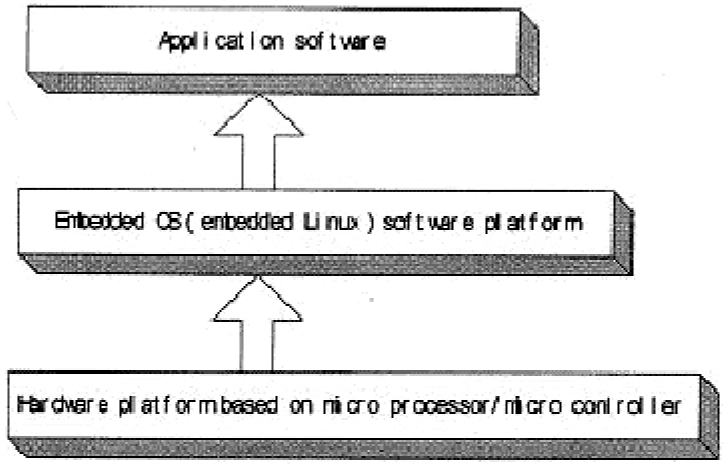

Fig. 2: Hierarchy of Embedded System

3.2 Terminal Hardware Design

The hardware design is based on a 32-bit arm-core processor, which has the following advantages. First, it strongly supports real time tasks and is able to complete multiple tasks in short response time. This can cut down on the execution time, enhancing efficiency and the timeliness of data. Second, the process has an excellent ability to protect information which can help to avoid the cross effect errors that sometimes occur between various types of modularized software. This can also lead to the advantage of beneficial software diagnoses capabilities. Third, the processor has an expandable structure which allows it to basically expand into a highly performance embedded microprocessor with low cost and low power consumption. The processor is widely used in electronic products, wireless communication, network communication and various other electronic fields.

The main intelligent device includes a central processor module, location module, mobile communication module, audio synthesizing module, sensor signal input/output devices and LCD (Liquid Crystal Display) touching screen module. The central processor module focuses on dealing with the vehicle's real time information while the location module is responsible for receiving location information from satellites and sending it to the central module. Differential Global Positioning Systems (DGPS) are used to calibrate GPS location information by signals from satellite stations while the mobile communication module is used to provide the vehicle exchange information with the TMC and the base station. Fig. 2 depicts the relationship between the main modules.

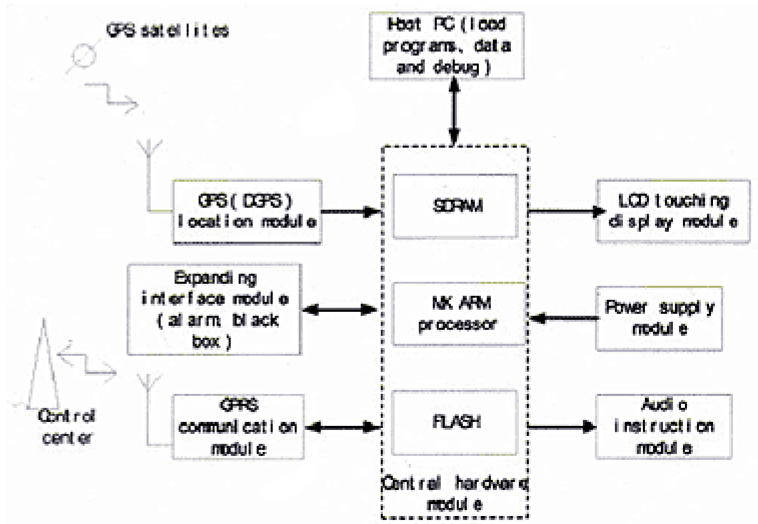

Fig. 3: Schematic Diagram of the In-Vehicle Intelligent Device

The central processor module receives information from the location module and uses the GPRS module to communicate with the TMC. It's complete functions include receiving the information from the TMC, decoding the coded information, extracting the instructions, controlling the touch screen to display the communicated information (i.e. the command, triggered instructions, current coordinates (latitude, longitude, speed, time and the auto-direction, etc.) in the digital map. It can also set up personal system services from the touch screen providing helpful information and providing additional communication with the user. Furthermore, the advanced input/output module makes the system fully prepared for the development of an alarm system and the vehicle black box.

3.2.1 Central Processor Module

The central processor module is composed of the CPU chip and peripheral memory devices. The embedded Linux is stored in a system known as FLASH. It is stored as a software platform and SDRAM (synchronous dynamic random memory) is used for running the application software. The memory devices are chosen to meet developing and updating requirements. A total of 64M SDRAM and 32M FLASH and a MMC (Multimedia Controller) card are selected by considering that at least 12M memory space is needed for downloading area information.

In this system, an MX series CPU from Motorola was selected for its superiority which included:

1. High performance and low power consumption. The ARM920T microprocessor works with a frequency of up to 200MHz. Such strong CPU processing ability and high calculation speed can meet various needs of mobile environments, dealing with data instantly.

2. Versatile interfaces are included. General interfaces, such as an LCD/SDRAM/USB controller, A/D converter and MMC/SD main controller module are integrated into the central processor, and many peripheral interfaces are provided to make the expanding of the application possible (i.e. multi-media function). This feature can fully satisfy the demanding interface requirements of the intelligent device.
3. With low power consumption and high integrity the hardware used can be minimized.

Two SDRAM chips are chosen and used simultaneously to realize the 64M bytes of required memory space. The parallel bus technology is used to expand the width of the data bus. Two 16M_16-bit chips which use the same control signals including chip select, write, and read signals plus identical internal chip addresses combine to constitute the 32-bit sub storage device module. The sub module is always considered as a whole component to be operated while storing different data. Thus the total data and the speed would theoretically be twice that of using one single chip.

The selected SDRAM have the same working frequency as the CPU external frequency, using the same clock to exchange data without delaying or waiting. It not only leads to better system performance, but also simplified design, improved data transmission speed, and a satisfactory data access rate for multiple instructions in the application software.

The parallel connection of 32-bit wide data is also used in FLASH connection to realize a high data transmission rate. The embedded Linux OS (Operating System) is loaded to the 32M FLASH connection with a capacity of 256M bits and is separated from data space. This protects the OS and disks from virus intrusions and ensures the system remains stable and durable.

3.2.2 Main Peripheral Modules

1. Display module: The display module consists of a color LCD display (640_240mm touch screen) and an LCD controller. The touch screen technology is a new way of human-machine conversation. Compared with the conventional facilities, such as keyboard and mouse, the touch screen function is more direct. The writing input function can be realized after installing recognition software.

   Related literal information is displayed on the LCD while receiving direct route or audio instructions. Customers can then manage personal service through the touch screen.

2. Location Module: A GPS receiver, antenna, and DGPS constitute the module. The location module receives real time location information from the satellites. The data frames are then analyzed and stored in the intelligent device and transmitted to the MTC if asked.

The GPS module consists of frequency converter, signal path, a microprocessor and memory unit. It sends location information to the central processor via a serial port and the central processor can also send setup instructions to the GPS module to control the working status and mode. A specified antenna is also needed to help receive satellite signals. Generally signals from more than 3 satellites are needed for accurate location identification. Setting the antenna on top of the vehicle can acquire better location data, and the location resolution can be improved from receiving correction signals from satellite stations by adding a DGPS module.

3. GPRS (General Packer Radio Service) communication module: The main task of this module is to receive instructions or information from the TMC and transmit the status information of the host vehicle.

   The GPRS mobile module is selected to complete communication with the advantage of a high data transmission rate, the efficient utilization of wireless channel resources, and excellent performance bursting data transmission. The low cost and flexible adaptability to data flow is also welcomed.

3.3 Software Design

3.3.1 Software Structure

The software part can be divided into 3 layers including the Operating System (OS), a Graphic User Interface (GUI), and application programs. The OS and GUI are critical components that constitute the foundation or platform of the entire system. The embedded Linux OS was selected for its many features. First, it has integrated a process by which the programmer can easily set up the developmental environment and cross-compiler, and avoid the problems associated with using an emulator (ICE) in an embedded system. Second, all the key files are open which makes the design and development of a real time hardware system possible. The development of a real time software system is also made possible with the use of this system. Additionally it provides strong support for network protocol that can be utilized to develop the PPP and TCP/IP network protocols which both directly support the wireless communications portion of the system.

GUI is a important part of the embedded system because it provides the application software with the standard interface functions of drawing point, lines, a dialog box, listing controls, and the message driving system. The embedded GUI systems such as Xwindow, MicroWindow, and Qt/Embedded, are widely used and available. However, they are not always suitable for vehicle intelligent devices because of their large size and structure or their low execution speed. The MiniGUI system is comparatively simple and is chosen in our design for its small size, low requirements of system resources, excellent performance and reliability, and flexible configuration.

As for storage and access of location and traffic data, the embedded database MiniDragon was selected. All the data are managed in areas divided by information from the base stations through the GPRS communication system. Each area is mapped to a data chart storing the time segment of service, instruction position trigger information, transportation status collection information, and temporary broadcasting service information. Some temporary charts are also set to store non-public information.

3.3.2 OS Repotting

The OS platform consists of bootloader, a system program process and an FS (file system).

1. Bootloader: The program has not been loaded since the circuit is electrified so at this point in the process the initialization is done by the bootloader. The main task of bootloader is to initialize the running environment for the OS, including setting abnormal vector charts and abnormal data processing functions, initializing memory devices, configuring data stacks under different ARM modes, enabling abnormal interrupts, and completing the handover of processor modes and status results.
2. OS: Standard Linux source code can be downloaded from an authorized web site. Related files about the CPU chip and peripheral devices should then be calibrated, including mainly clock set-up, interruption establishment, memory allocation, and some other configurations regarding the registers. The necessary device drivers and protocols should be added to the OS.
3. FS: The operation of an embedded Linux system needs the support of management through user spaces, configuration files, booting scripts, and the running library as well as the OS image. MiniGUI and MiniDragon are also compiled in the root file system image.

When related object files are created the switches are adjusted to make the target turn to the bootstrap mode. The target device is then used by serial port to communicate with the host computer, receive the bootloader image, and load it to the flash process. Then the OS and root FS images can be loaded to FLASH with the support of bootloader. After the initialization of the system, a channel connecting the host computer and the target is set up to print the debug information directly on the screen of the host PC. The serial port is used as the debugging channel for the system. After this process is complete a switch to the OS booting mode and then the Linux OS result in the system running on the ARM platform as depicted in the following flow chart in Fig. 4.

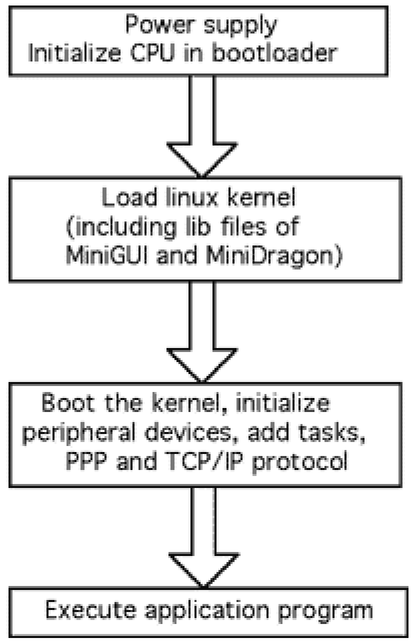

Fig. 4: Execution Process of the Software System

When the circuit board is electrified, the system task process jumps to the Address 0x00000000 where bootloader is loaded. Then the OS image is copied from flash to SDRAM and the booting code of the Linux OS is executed to load the appropriate information. Then the Linux OS gets control of the system to schedule application software including setting up the graphic management window, displaying the navigation instructions, completing the personal service management by the touch display, communicating with the transportation center, and turning on the alarm system.

After the above steps are complete the software platform of the intelligent device has been set up. The application software receives real time GPS data, changes them into coordinates and triggers the related operations. Another process is created for communication with the TMC through wireless radio or broadcasting to complete the task of location, navigation, and other configured services.

4. Conclusion

The embedded Linux has been successfully applied in intelligent in-vehicle devices and the ITP conception also works well in the ITS system. This shows that a well-designed, intelligent device matched with other advanced devices and suitable ITS design can greatly improve the efficiency of the whole system.

REFERENCES

- Yundong Li "Application of Embedded System in Vehicle Navigator Design," Journal of Telemetry, Tracking, and Command vol. 5, 2003.
- Kebin Huang "The Realization of Booting System based on ARM£ʃuCLinux," Application of Electronics Technique, vol. 3 2004.
- Chunlei Du, "Structure of ARM and Related Programming," Beijing, Tsinghua University Press, 2003.