Development of a novel MLS controller based on embedded ARM

LI Xiang-jun(李向军)[1], QIN Bo(秦 波)[2]

(*1. College of Information and Navigation, Air Force Engineering University, Xi'an 710077, China;*
2. College of Automation, Xi'an University of Posts & Telecommunications, Xi'an 710061, China)

Abstract: The controller plays an important role in microwave landing system(MLS) navigation equipment. The embedded advanced reduced-instruction-set-computing(RISC) machine(ARM) is introduced to develop a new MLS controller based on the functional requirements. A design including hardware and software of an ARM system is given. The hardware design involves a ARM smallest system and its interface. μC/OS-II, a good real-time operating system (RTOS), is introduced in the software design. The task partitioning and management are also involved. The experimental results demonstrate that the embedded ARM technique is an effective way in the complex multi-task system, and it meets the high integration and real-time processing requirements of the navigation equipment.

Key words: microwave landing system (MLS); embedded reduced-instruction-set-computing(RISC); advanced RISC machine (ARM); real-time operating system(RTOS); micro operating system (μC/OS)

CLD number: TN965⁺.7 **Document code:** A
Article ID: 1674-8042(2013)02-0180-04 **doi:** 10.3969/j.issn.1674-8042.2013.02.019

0 Introduction

The microwave landing system, abbreviated MLS, is a new precision approach guidance standard adopted by International Civil Aviation Organization (ICAO)[1]. The MLS ground stations transmit signals, providing an aircraft with the azimuth, elevation and distance. Besides, a complete monitoring capability is essential. That is, the system must monitor the transmitted signals. When the system fails, or the monitoring parameter exceeds the range of allowable error, it alarms in time and automatically switches to the backup or turn off the transmitted signals to ensure reliability and integrity. In the past, these functions are assigned to different units, each of which performs a single task. Data are exchanged frequently between them by low-speed serial bus such as RS422. For example, in the MLS480, the ground stations made in Italy, six circuit boards, 230 mm × 230 mm, cooperates to achieve the above functions, on which hundreds of discrete components are mounted. Simple tasks are performed by micro control units(MCUs) such as 8031 and 8088[2]. This results in poor reliability and scalability.

Now a new embedded technology, advanced reduced-instruction-set-computing (RISC) machine (ARM), can change this situation. ARM is a 32-bit processor architecture with small size, powerful instruction set and rich interfaces[3]. It has been the dominant member in the embedded microprocessor family with high performance, high integration and low cost. Now ARM has been adopted in so many fields as industrial control, wireless communication, signal processing and network. In this paper, the functions of a new MLS controller are redefined, and then the embedded ARM is introduced to design the MLS controller.

1 MLS controller

A MLS controller is the "brain" for the MLS ground equipment[4]. Its task includes as follows.

1) Accept instructions from a portable terminal or the front panel to switch the transmitter on or off.

2) Display the device status.

3) Receive the monitoring data from some units, compare them with the threshold and decide to switch or power off in no more than 1 s.

4) Monitor the environment by sensors and battery in real time and send the alarm.

5) Synchronize between the azimuth station and the elevation one.

6) Store the user level information and the alarm records.

The second listed above is critical for the MLS. It secures the safety for an aircraft ready for landing when the MLS ground station is in fault[5]. The signal relationship between a MLS controller and other

* **Received date:** 2013-01-11
Corresponding author: LI Xiang-jun (lixiangjun@mail.nwpu.edu.cn)

units is shown in Fig. 1.

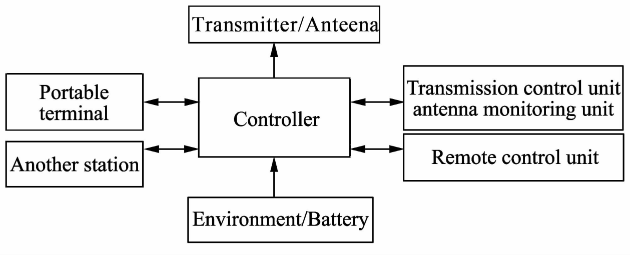

Fig. 1 Relationship of signals between MLS and other units controller

In Fig. 1, the MLS controller communicates with portable terminal, the transmission control unit, the remote control unit and another station by bus. The environmental sensors output for the environment is digital 0/1, while the battery output is analog direct current.

2 Hardware design

2.1 Bus selection

The bus among the controller and other function units affects the reliability greatly. A large amount of data on the bus is exchanged to meet the requirements of real-time data processing and reaction. So a high-speed and robust bus is necessary to the system. In this paper, controller area network (CAN) bus is introduced.

CAN bus is a device interconnection control network. Compared with other buses, CAN bus is considered to be one of the most promising field bus with outstanding reliability, timeliness and flexibility. Its fastest speed can be 1 Mbps within the distance of 40 m. It can work at point-to-point, point-to-multipoint and broadcast mode[6]. It can meet the needs of the design based on its characteristics.

For CAN bus application, CAN controller and transceiver should be considered. Now many types of MCUs including ARM integrated CAN controller. LPC2290, one of the NXP ARM described in the next section, is an example. For CAN transceiver, TJA1050 is a good choice.

2.2 ARM selection

ARM is available in different cores, such as ARM7, ARM9, etc. ARM7TDMI is enough for general applications. Its good processing capacity can be no less than 60 Mbps, and meet most of real-time requirements. Also, its architecture facilitates debugging[7].

So in our design the NXP ARM LPC2290 is selected, which is based on a 16/32 bit ARM7TDMI-S CPU with real-time simulation and embedded trace support. It embeds many resources and interfaces, such as 16 KB static RAM, 8-channel 10-bit A/D converter, two 16C550 compatible UARTs and fast I^2C. Moreover, LPC2290 integrates two CAN controllers compatible with CAN2.0B, which can be accessed by 32-bit registers and RAM[8]. Because there is no Flash program memory and no enough RAM for data cache in LPC2290, generally external memory expansion is necessary[5]. The ARM minimum system with 2 Mb Flash (SST39VF1601) and 512 Kb RAM (ISSI IS61LV25616AL) are designed in application.

2.3 Overall design

The novel MLS controller consists of a ARM core board, power and peripheral interface circuit in hardware. The signal conditioning circuit, two serial port level translators, two CAN transceivers and JTAG interface compose the peripherals. The hardware scheme is shown in Fig. 2.

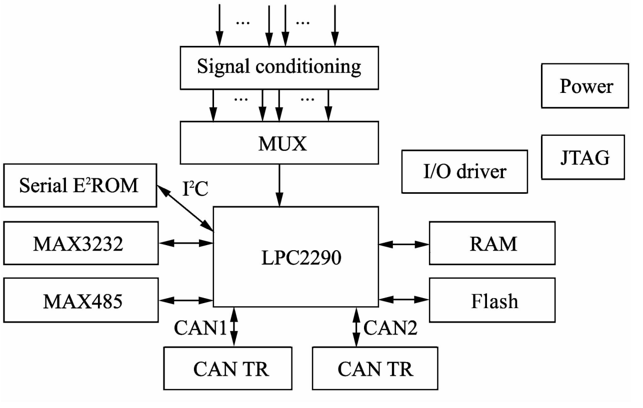

Fig. 2 Hardware structure diagram

In Fig. 2, MAX3232, level conversion chip from TTL to RS232, is used for UART0. In the way the controller can communicate with a portable terminal. For UART1, the bus is converted into isolated RS485 for station synchronization. CAN1 is used to communicate with other units at the speed of 500 Kbps, while the isolated CAN2 is used to communicate with a remote controller at a low speed of 50 Kbps. Serial E^2ROM (AT24C256) can store user login information to verify the identity of the user. It also stores alarm records, such as time, alarm source and data. Because the analog signals like battery levels can not be connected to A/D converter in the ARM directly, they must be conditioned to 0 – 3.3 V. There are Fourteen analog signals should be conditioned, and a MUX (ADG706) is adopted. A precision reference (TL431) is reserved for the fifteen channels of the MUX in order to test the A/D converter. The signal conditioning circuit, MUX and the A/D converter embedded in the ARM thus form the data acquisition system. Because I/O of the ARM can afford enough current to drive the transmitters and the antennas, driving circuit like TLP521 is necessary.

Fig.3 shows the real controller. The part in the red rectangle is the ARM core board, including LPC2290, RAM, flash memory and so on.

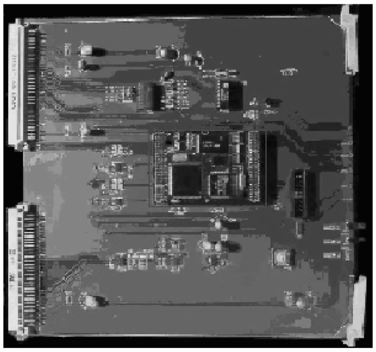

Fig.3 Real controller

3 Software design

3.1 RTOS

The MLS controller is a multi-task system, and it requires real-time processing and reaction. The traditional design based on the foreground/background systems would be too complex, and real-time processing can not be guaranteed. Thus real-time operating system (RTOS) is a good way.

Among the matured RTOS, micro operating system (μC/OS-Ⅱ) is an open source code embedded operating system, whose kernel is pre-emptive. In addition, the code of μC/OS-Ⅱ, written in C language mostly, is short and easy for porting based on different hardware platform such as ARM and digital signal processor (DSP)[9].

3.2 Software hierarchy

In RTOS, software hierarchy is shown in Fig.4.

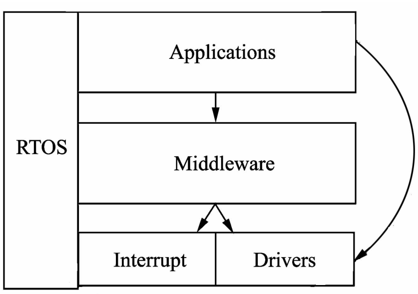

Fig.4 Software hierarchy

The software mainly consists of hardware drivers, interrupts, middleware and applications.

Porting the μC/OS-Ⅱ C based on LPC2290 is the first step. Hardware drivers refer to the basic register accessing, such as UART, I^2C, CAN, A/D aod IO. Interrupts includes UART, I^2C and CAN. Middleware is the mature function library. In μC/OS-Ⅱ, application means task partitioning and management. Each task is assigned different RAM and priority. Tasks can be synchronized by semaphore, message box or message queue[10].

Table 1 lists the tasks and their characteristics. The smaller the number of the priority level is, the higher priority the task occupies.

Table 1 Tasks partitioning

Task name	Priority	RAM allocation (bytes)
Serial 0(RS232) RX	2	1 024
Serial 0(RS232) data processing	3	1 024
CAN1 RX	4	4 096
CAN1 data processing	5	512
Serial 1(RS485) RX	6	512
A/D sampling	7	512
CAN2 RX	8	512

4 Experimental results

After hardware debugging and software programming are finished, the controller, together with other units, is assembled in cabinet for experimental test[11]. The layout of the cabinet is shown in Fig.5. A simplified CAN bus application layer protocol is customized, and it performs effectively[12].

Fig.5 Controller (the first board inner) in the cabinet

Fig.6 gives an example of the software interface in the portable terminal communicating with the controller. More importantly, the fact that the controller responses in no more than one second when a fault occurs shows that the embedded ARM has high performance.

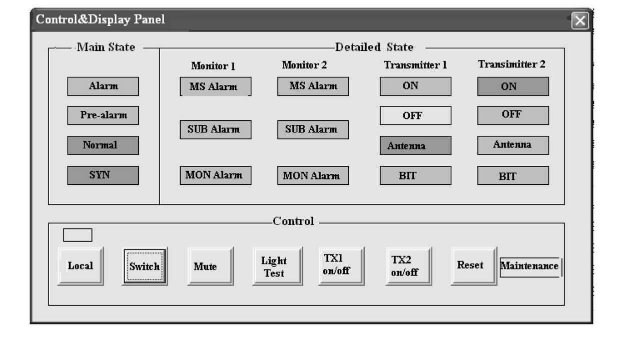

Fig.6 Virtual panel communicating with controller

5 Conclusion

In this paper, the embedded ARM is introduced to design a new MLS controller for the first time. Besides, CAN bus is introduced in the controller to exchange data with other units. Embedded hardware platform based on μC/OS-II RTOS facilitate system development. The novel MLS controller with small size (230 mm×230 mm) can perform the same functions of those six circuit boards. It is proved to be more stable and reliable in experiments. It also provides an effective way to design navigation system. In fact, the platform based on the embedded ARM has been also adopted in new navigation equipment.

References

[1] Miller F P, Vandome A F. Microwave landing system. VDM Publishing House, New York, 2010.
[2] MLS-480 Technical Specification, 2006.
[3] LIU Kai. Basis of ARM embedded application technology. Beijing: Tsinghua University Press, 2009: 60-65.
[4] WANG Wei, ZHANG Bin. Design of MLS signal simulator. Telecommunication Engineering, 2011(7): 31-35.
[5] SHAO Weng-jian, GUO Wan-you. Design of MLS signal processing system based on DSP. Electronic Science and Technology, 2009, (1): 20-23.
[6] RAO Yun-tao. Theory and application on field bus CAN. Beijing: Beihang University Press, 2007: 130-141.
[7] DU Chun-lei. ARM architecture and programming. Beijing: Tsinghua University Press, 2003: 70-82.
[8] PHILIPS. LPC2210/2290/2292/2294 user manual, 2004.
[9] Labrosse J J. Micro-C/OS-II: the real-time kernel. Beijing: Beihang University Press, 2003: 220-270.
[10] ZHOU Li-gong. Basis and experiments of ARM microprocessor. Beijing: Beihang University Press, 2005: 102-125.
[11] PENG Xue-mei, ZHANG Hui. MLS signal simulation and application. Electronic Science and Technology, 2008, (6): 24-27.
[12] ZHOU Li-gong. Experimental course of ARM embedded system , 3rd editon. Beijing: Beihang University Press, 2006: 226-235.