

Embedded Ethernet Interface Using Arm Processor

Raghava Bhonagiri¹, Prof.K.Subba Rao²

^{1,2}(Electronics and Communication Engineering, DVR College of Engineering and Technology, Hyderabad, India)

Abstract: Currently device with microcontroller has been widely used in industrial field. However, a large number of devices don't have the network interface and the data from them can not be transmitted in network. A design of ARM processor-based embedded Ethernet interface is presented. In the design, an existing SPI serial device can be converted into a network interface peripheral to obtain compatibility with the network. The design mainly consists of SPI communication module, processor module and Ethernet interface module. In the design, embedded real time operating system $\mu C / OS-II$ is transplanted into the microcontroller LM3S8962 and the data can be transmitted between remote SPI serial devices and host computer. After the design is completed, the system is tested and the results show that Ethernet is connected between the host and ARM Cortex and the terminal data can be transmitted via Ethernet.

Keywords: ARM processor; interface; Ethernet; SPI

I. Introduction

With the development of network technology and communication technology, the need that industrial control can be completed via network has become a trend. In traditional industrial control system or data acquisition system, the structure that one host connects multiple serial devices through multiport serial cards is adopted. The task of host is to communicate with each serial device, process data and interact between the operator and computer. This structure is feasible in the case of fewer devices, lower transmission rate. But when a host needs to connect more serial devices at the same time with higher transmission rate and the data processing is more complex, the system performance is poor. In addition, these serial devices connect the same host may be geographically far and this will increase the length of wiring and drop communication quality. So a solution need be found to realize the communication between industrial control devices and Ethernet. As the embedded system itself has the performance of network and human-computer interaction, it is possible that the embedded system replaces the previous control method based on microcontroller. So an ARM processor-based embedded Ethernet interface system is designed. In the system, the introduced microcontroller LM3S8962 can communicate with serial data acquisition equipment is at the terminal through SPI interface and can transmit data to remote host computer through Ethernet interface. Compared with the system that a host is connected to many serial devices, the task of host is only to complete a single Ethernet communication and its load is lower. In the system the data can be transmitted transparently between host and serial device and the host can communicate with any serial device connected Ethernet without knowing each other's physical location[1,2]. When the serial device needs to establish communication with a new host, what you can do is just to connect the host to Ethernet.

II. System Design

The main purpose of the design is to make traditional monitoring system have the capabilities of remote monitoring or data transmission by introducing Ethernet interface it. In the design, structured modular design method is adopted and the system is mainly composed of SPI communication module, processor module and Ethernet interface module, as shown in Fig. 1.

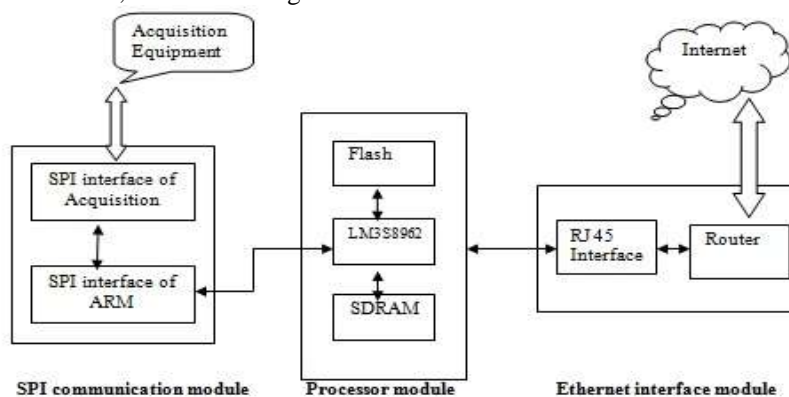


Figure 1. System structure

SPI communication module is the interface between central processing unit and RF transceiver module nRF24E1. Its work is to receive data form nRF24E1 and then transmit them to the microcontroller LM3S8962.

Processor module is the core part of the design, in which the ARM chip LM3S8962 is used to complete the complex operations and receive a lot of data from nRF24E1 as a slave. In the module, data link between SPI port and Ethernet is established, SPI data stream format is specified, the transmission rate between serial data stream and IP data packets is controlled and IP packet is received or sent through reading or writing Ethernet interface module.

In the Ethernet interface module, the collected data are uploaded to a PC via Ethernet interface by using the fuctons of OS μ C / OS-II and the commands from the host computer are received commands to control the data acquisition system.

III. Communication Module Design

A. SPI Interface

SPI interface is used to realize synchronous serial data transmission between CPU and low-speed peripheral devices by way of full-duplex communication. Its data transfer rate is up to several Mbps. SPI interface works in master-slave mode and it includes four signals: *SCLK*, *MOSI*, *MISO*, and */SS*. *SCLK* is the common clock in the entire SPI bus, *MOSI* is the master output, slave input, *MISO* is the host input, slave output and */SS* is used to mark slave. In two devices which communicate with each other via SPI bus, the slave is low level and host is high level in */SS* pin. In the design, because the front-end RF transceiver module nRF24E1 can only be used as host in SPI communication, the microcontroller LM3S8962 is used as a slave and the mode of a single master and multi-slave are adopted, which is shown in Fig. 2

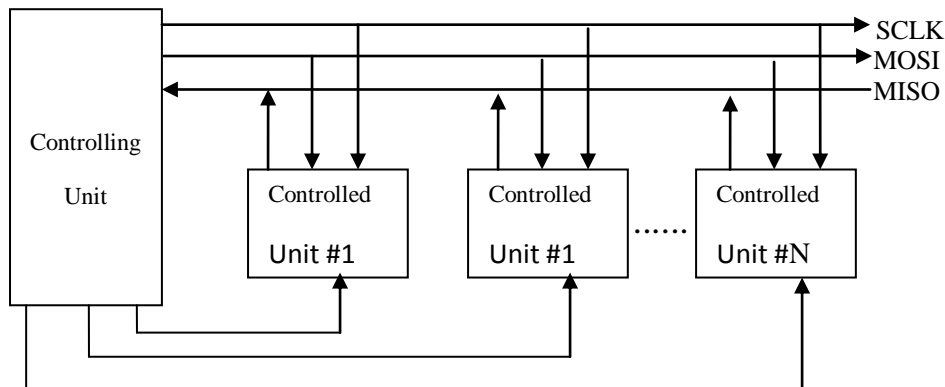


Figure 2. SPI bus structure

The interface circuit of LM3S8962 and nRF24E1 is shown in Fig. 3. Through the SPI interface, the maximum transfer rate achieves 2Mbps which meets the majority of applications. As nRF24E1 reads input data at rising edge of *SCLK* and reads output data at falling edge of *SCLK*, the 00 pattern of LM3S8962 is selected. So the data outputted from nRF24E1 is inputted into LM3S8962 after half a clock cycle in the contion of synchronization status.

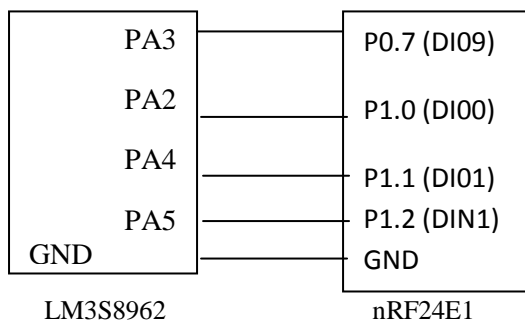


Figure 3. SPI interface circuit between LM3S8962 and nRF24E1

B. Ethernet Interface

In the Ethernet interface module, Ethernet controller are divided into two layers according to its functions and it is shown in Fig. 4. One is media access controller (MAC) layer and the other is network physical (PHY) layer. They correspond to Layer 2 and Layer 1 in ISO model. MAC layer provides the treatment

on data sending and receiving. It also provides an interface to PHY through an internal medium independent interface.

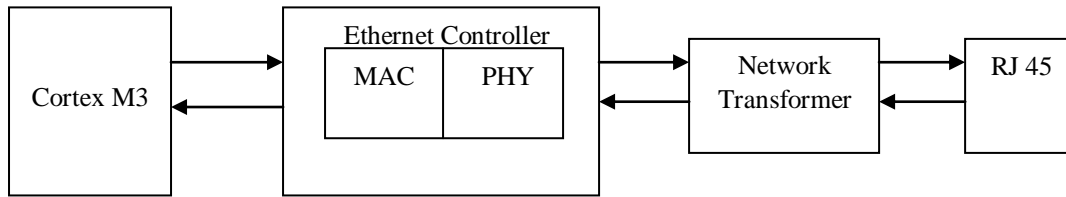


Figure 4. Ethernet controller and Ethernet interface

In this design, LM3S8962 supports Ethernet interface and network transformer HR601680 which is connected between Microcontroller LM3S8962 and connector RJ45 is used to improve the signal anti-interference capability. HR601680 is a 1:1 transformer with a smaller package and it supports 10M/100M Ethernet.

III. Software Design And System Testing

A. μ C/OS-II Transplanting

For the transplanting of OS μ C / OS-II, its main task is to re-write the codes which are associated with the processor and compiler[3]. The codes include the following: re-definition of compiler-related data type, interrupt-switching code, stack growth direction, the initialization function for task stack, task-switching function and so on. The transplanting is completed in the following environments: the compiler tool is IAR FOR ARM and EasyARM8962 development board is used as target board. In order to transplant μ C / OS-II into the ARM processor LM3S8962, three ARM architecture-related documents: *OS_CPU.H*, *OS_CPU_A.ASM* and *OS_CPU_C.C* are modified.

1) Configuration of the file *OS_CPU_A.ASM*:

The constants, macros and data types which are required by μ C / OS-II are re-defined. The ARM processor-related contents are modified. For example, the stack of ARM Cortex-M3 is configured from high addresses to low addresses in the growth direction.

2) Configuration of the file *OS_CPU_A.ASM*:

Five simple functions are programmed by assembly language in the transplanting process. *OS_ENTER_CRITICAL()* is to close the interrupt source. *OS_EXIT_CRITICAL()* is to re-open the interrupt source. *OSStartHighRdy()* is to run the current highest priority task. *OSIntCtxSw()* is used to switch interrupt task and it will be called in process of exiting the interrupt service function *OSIntExit()*. *OSCtxSw()* will be called if a task wants to give up the use right of CPU.

3) Configuration of the file *OS_CPU_C.C*:

In the functions defined by *OS_CPU_C.C*, *OSTaskStkInit()* is associated with CPU and it needs to be modified in μ C / OS-II transplanting. It is called to initialize the stack in the task initialization phase.

After the work above, three files: *Startup.S*, *Target.C*, *Target.H* in *Target* directory need to be programmed according to the actual situation in target board[4]. And then μ C / OS-II can be run on the microcontroller LM3S8962.

B. Data Transmission

In order to transmit the data from SPI serial to Ethernet, two system tasks are established in OS μ C / OS-II. One is to receive front-end data through SPI interface and the other is to transmit data to Ethernet.

1) SPI receiving task:

For the case that the data are transmit to Ethernet, the data which have been arrived at SPI port are stored into SPI sending buffer and packaged according to TCP/IP protocol and then are added IP and UDP message head on the condition that the PC with SPI interface is set to SPI slave mode and the SPI interface is enabled. At last, the converted data are sent to the host through the corresponding UDP port. This process is shown in Fig. 5(a).

2) **Ethernet receiving task:**

In the Ethernet task, in order to receive the data from Ethernet in the system, the local IP address and subnet mask must be set firstly, and the appropriate UDP port is opened to monitor whether there are data in UDP port. As UDP packet, the data which have been reached the UDP port, are analysed according to TCP/IP protocol and then stored into SPI receiving buffer. At last, the analysed data are sent to the SPI serial device through SPI interface driver. In the process above, an infinite loop is used and the process is shown in Fig. 5(b).

After the software and hardware designs have been completed, the ultimate generated codes are compiled and downloaded to the target system for testing. The codes include hardware system startup code, μ C / OS-II kernel, TCP/IP protocol stack, SPI port driver and part of the application. In the testing, PC is used as SPI port device which is connected with nRF24E1 and the host respectively. The crossover cable is used to connect the host and RJ45 in garget system. IP address and subnet mask are set in both computers to ensure that the communication between them is in the same network. When the IP address of host is inputted by ping command the results show that the Ethernet is connected between the computer and ARM Cortex.

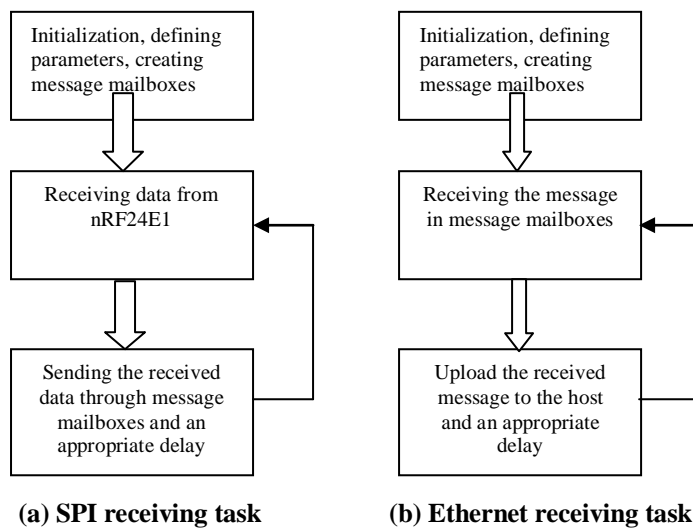
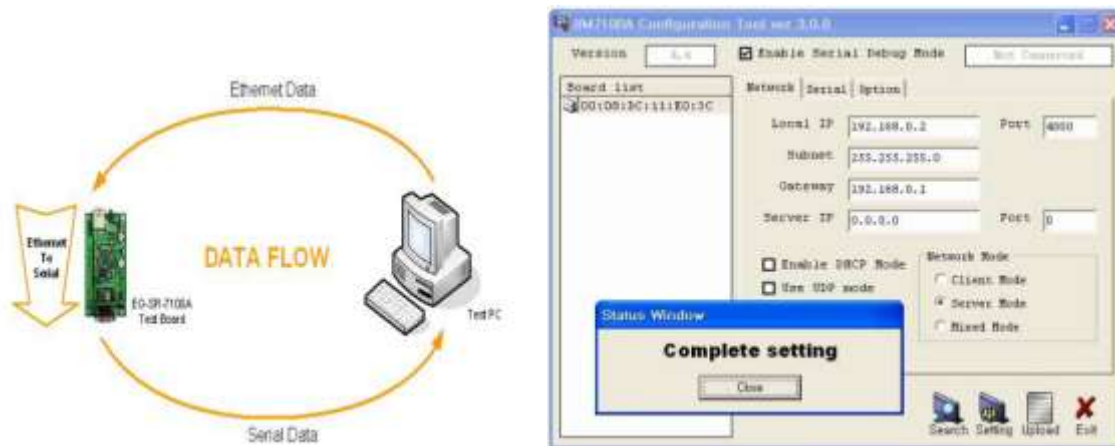


Figure 5. System tasks for transmitting data

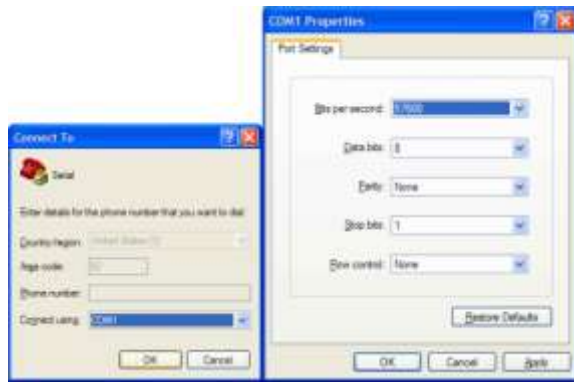
When TCP & UDP test software and UDP communication program are run in the host, the test results also show that the communications from both sides are successful.

C. System Simulation Results



Data Flow about Ethernet-to-Serial Result

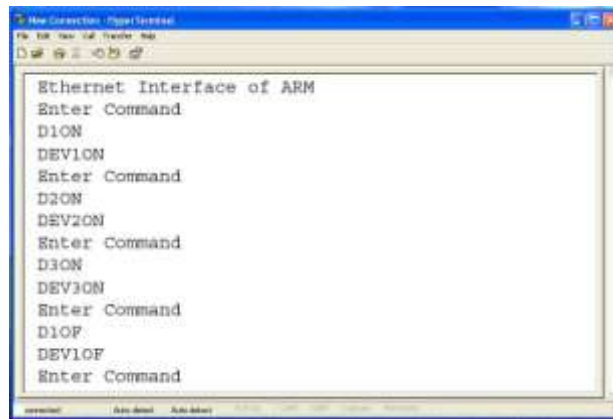
Window after Running “Setting”



Serial Terminal Program Configuration



Network Terminal Program Configuration



Verifying the output in HyperTerminal by giving Commands

IV. Conclusion

In order to transmit the data from an existing device with SPI interface to network, a embedded Ethernet interface based on ARM processor is designed. On the basis of hardware platform, real time OS μ C / OS-II is transplanted into the microcontroller LM3S8962 and the SPI serial data and Ethernet data packets can be converted to each other by software programming. For traditional serial devices, the design provides an effective implementation to connect Ethernet and the remote devices with SPI serial interface can be changed into the shared resources in network. This design can be used widely in remote data acquisition and control system in industry.

References

- [1] CHEN Guo-ju, "Design of a monitoring system based on ARM and Ethernet applied" 2009. YU Cheng-bo, LIU Jie, and TAO Hong-yan, "Research on remote monitor technology of equipment," 2002.
- [2] YU Cheng-bo, LIU Jie, and TAO Hong-yan, "Research on remote monitor technology of equipment," 2002.
- [3] LIU Hong-li, "The Research and Experiment of the Embedded System μ C / OS-II on PC," 2009.
- [4] ZHANG Shi, DONG Jianwei, SHE Lihuang, "Design and development of ECG monitor's software system," 2007.

Biographies



Raghava Bhonagiri, presently working as Assistant Professor in ECE department at DVR College of engineering and technology, Hyderabad. He has got B.tech degree from Swami Ramanantha thirtha Institute of science and Technology Nalgonda, A.P. He has got M.tech (Embedded Systems) from same Swami Ramanantha Thirtha Institute of science and Technology Nalgonda, A.P.



Prof. K. Subba Rao presently working as HOD in ECE department at DVR College of engineering and technology, Hyderabad. He has got B.tech degree from JNTU Hyderabad in 1984. He has got M.tech degree from JNTU Hyderabad in 1987. He has got PhD from IIT Kharagpur in 1994. Previously he worked in CMC Hyderabad as Systems Engineer from 1984-1985. He worked as HOD in several colleges like Amaravathi University, Amaravathi, Auroras Engineering College, CVSR Engineering College, DRKIST in Hyderabad from 1988-2012.