Jurnal Teknologi, 41(D) Dis. 2004: 1–10

DEVELOPMENT OF A REAL-TIME EMBEDDED REMOTE TRIGGERING AND MONITORING SYSTEM WITH SC12

CHUI YEW LEONG[1] & ABDUL RAHMAN RAMLI[2]

Abstract. This paper presents a real-time embedded remote triggering and monitoring system using SC12. A protocol converter associated with data interpretation unit has been developed and implemented. In order to expand simultaneous operation channel with data interpretation unit, intelligent auto-diagnostic features has been implemented for run-time error detection purposes.

Keywords: Embedded system, triggering and monitoring system, auto-diagnostic

Abstrak. Kertas kerja ini membentangkan sistem kawalan dan pemantauan jarak jauh dengan menggunakan SC12. Satu penukar protokol dengan unit interpretasi data telah direka bentuk dan dilaksana. Untuk menambahkan saluran operasi unit interpretasi data, satu ciri auto-diagnostik pintar telah dilaksana untuk mengesan ralat.

Kata kunci: Sistem terbenam, sistem pemicuan dan pemantauan, auto-diagnostik

1.0 INTRODUCTION

Recently, the development of internet-based remote triggering and monitoring system is becoming one of the hottest research topics. Internet provides convenience in a time when everyone is desperate for easy solutions. As years pass by, people are experimenting more adventurous ways on the internet, making it even better. This makes internet-base connectivity theme so interesting. There is no limit other than imagination.

In fact, the trend of modern communication system is oriented to the standard pattern with global access, and multi-system (protocol) supported. The IP protocol is nicely defined as "system glue" to integrate multiple types of systems, services, and interfaces. Thus, the traditional system no longer compromise the demand of market to accomplish a wonderful globalization system. As a result, researchers and engineers are finding ways to overcome the limitation of distance and time, to achieve a perfect real-time remote triggering, and monitoring system. There are various types of technology to implement modern remote triggering, and monitoring system. Supervisory Control and Data Acquisition (SCADA) System, Telemetry System, MRC

[1&2] Intelligent Systems and Robotics Laboratory (ISRL), Institute Of Advanced Technology, Universiti Putra Malaysia. 43400 Serdang, Selangor. Tel: 603-89467532 Fax: 603-86566061
Email: chuiyewleong@hotmail.com, arr@eng.upm.edu.my

2 System, Distributed Control System (DCS), Wireless and Mobile Control System, and Internet/Intranet Control System are some of the widely used remote control systems which are available in the market.

Real-time embedded internet-based remote triggering and monitoring system is an advance technology that eases the remote user to manage a local system through the internet connection. The implementation of such system figures a low cost but high in efficiency, reliability, and stability solution. Figure 1 shows the block diagram of a complete internet connectivity system.

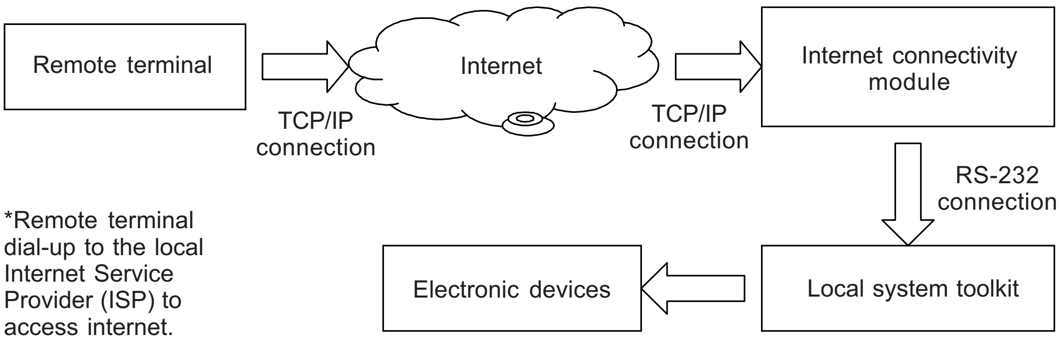

Figure 1 Block diagram of a complete internet connectivity system

Remote user can dial-up to the local Internet Service Provider (ISP) to get access to the internet connection. Through the internet connection, a remote user is able to establish a real-time connection to the Internet Connectivity Module (ICM) that is remotely located [1,2]. As a result, the remote user can nicely manage the operation of Local System Toolkit (LST) for home appliances triggering and monitoring purpose. The advantage of LST is to provide data interpretation module to increase the number of simultaneous operation channel. A run-time checking algorithm is implemented in LST as intelligent self-diagnostic module, to detect fault occurrences of the operation channels.

The system is defined as real-time system because connection-oriented protocol (TCP) is applied to accomplish the connection between remote terminal and ICM [3]. As a result, any operation from the remote terminal will affect the status of the LST immediately.

2.0 NETWORK EMBEDDED SYSTEM

Network embedded system is a system in a single chip that has functional protocol stack installed on a complete set of Real-Time-Operating-System (RTOS). The network embedded system can be defined as a small size personal computer which provides customize network solution for specific application purposes. Network embedded system integrates all features needed for modern Ethernet and Internet applications. Not only does the system hardware supply features such as Ethernet, serial ports, and programmable I/O pins, the software to use the interface is also readily integrated into the system.

SC12 from Beck-IPC is used to implement the system-on-chip [4]. SC12 micro-processor is part of the Beck-IPC family of system-on-chip based on the ×86 architecture. The chip is the ideal upgrade for 80C186/188 designs requiring 80C186/188 compatibility, increase performance, serial communications, Ethernet communications, direct bus interface, and more than 64 K memory. SC12 integrates up to 512 KB DRAM with increased performance, and up to 512 KB flash storage in reducing memory subsystem cost. It also integrates the functions of CPU, multiplexed address bus, three timers, watchdog timer, chip selects, interrupt controller, two DMA controllers, two asynchronous serial ports, programmable bus sizing, and programmable I/O pins on one chip. It is a highly integrated design that provides all Media Access Control (MAC) and Encode-Decode (ENDEC) functions in accordance with the IEEE 802.3 standard. Network interface of SC12 includes one 10Base-T with RJ-45 connector.

3.0 MECHANISM REVIEW

Significant operation mechanism of internet connectivity embedded remote triggering and monitoring system is concentrated in signal translation of network embedded system, digital signal switching control in local system toolkit, and error recovery capability of coding method.

3.1 Signal Translation

Signal translation is the most significant operation mechanism for system integration. Different unit of operation block requires different type of input data format for further processing. In this case, signal converter is necessary for signal translation [5]. The task of signal converter is to translate/convert input data format into another type of customize data format for expanded operation unit. Error checking and fault detection is embedded as supported functions of signal converter to enhance signal precision.

3.2 Digital Signal Switching Control

The process of demultiplexing recovers raw data from the network packet. Local System Toolkit will interpret the raw data to perform signal-switching control. Different incoming signal/data activates appropriate channel to trigger target switch or relay [6]. The mechanism of signal switching control is decided by the rules of program in the embedded system. Any custom operation can be defined in the embedded system for custom usage. In the case of local connectivity system toolkit, microcontroller will analyze the incoming data to decide, which relay to be triggered.

3.3 Error Recovery

Error recovery provides a way to optimize the data distortion of the system. The importance of error recovery is to ensure the possibility of fault-tolerance is optimized. Strong coding method provides higher resistance to the ambient noise or distortion signal. Combination of strong coding selection and error recovering enhance the reliability of data communication through a high risk or noisy medium. In the case of network communication technology, efficiency of error recovery is decided by protocol implementation [7]. The general case of TCP/IP uses its built-in function/mechanism to ensure the data packet to be transferred from source to destination in the most efficient way.

4.0 IMPLEMENTATION

The operation mechanism of real-time connection-oriented internet connectivity embedded triggering and monitoring system is illustrated in block-diagram as shown in Figure 2. The system implementation and development is divided into three parts: server, connectivity toolkit, and client. The implementation of server is concentrated in the application of network embedded system, Real Time Operating System, and network application programming. The development of local connectivity system toolkit is mainly focusing on circuit design and microcontroller programming. Client-side development involved Graphic User Interface (GUI) design, object-oriented programming, and Active-X object control. The implementation of network embedded

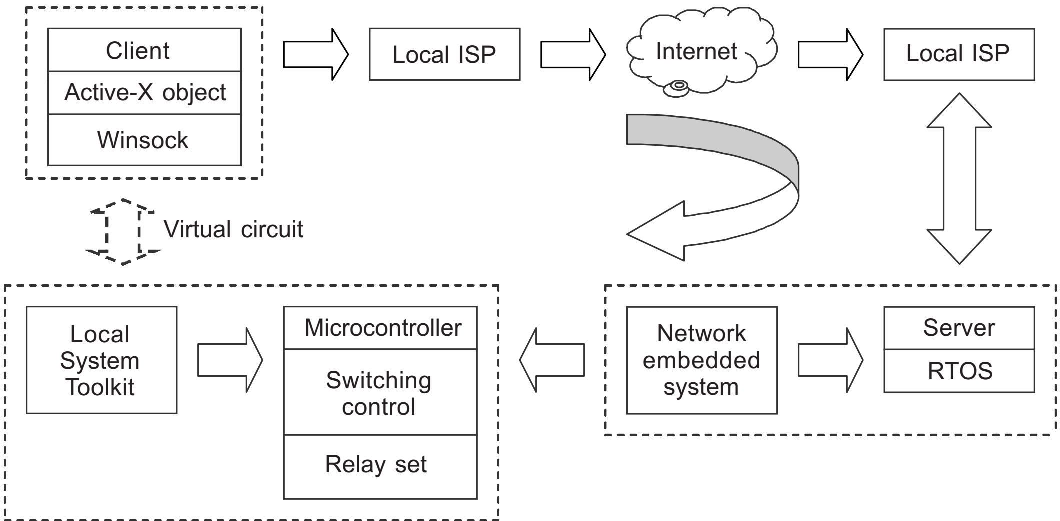

Figure 2 Operation of real-time connection-oriented internet connectivity embedded triggering and monitoring system

system is significant for the system design. SC12 from BECK-IPC provides most of the significant features to accomplish internet connectivity implementation.

Pin terminology of SC12 includes address/data bus, programmable I/O pins, programmable chip select, system interrupts, timer, 10-base-T interface, asynchronous serial ports, DMA, synchronous peripheral interface (SPI), I^2C bus, and reset/power fail generator. System configuration of SC12 is based on the ISA system architecture with some changes to meet embedded system requirements. The changes are: no ISA-bus, no Video-Interface, no Keyboard Interface, programming of Serial Ports, and DMA support. Figure 3 shows the system memory configuration and the basic I/O of SC12.

The task of RTOS in managing hardware and software resources is very important for the attention of I/O, Data/Address Bus, Interrupts, Programmable Chip Select, and CPU. RTOS maintains an abstraction layer of code to interact with hardware configurations. The implementation of RTOS provides a consistent application interface, which allows structural software development with consistent application program interface (API) for high-level software development, regardless of the hardware process management.

Consistent API allows the task of software development to be performed above the layer of RTOS. As a result, application programming is possible regardless of the consideration of hardware process management. The implementation of network application requires protocol API. Protocol API is a group of network function interface

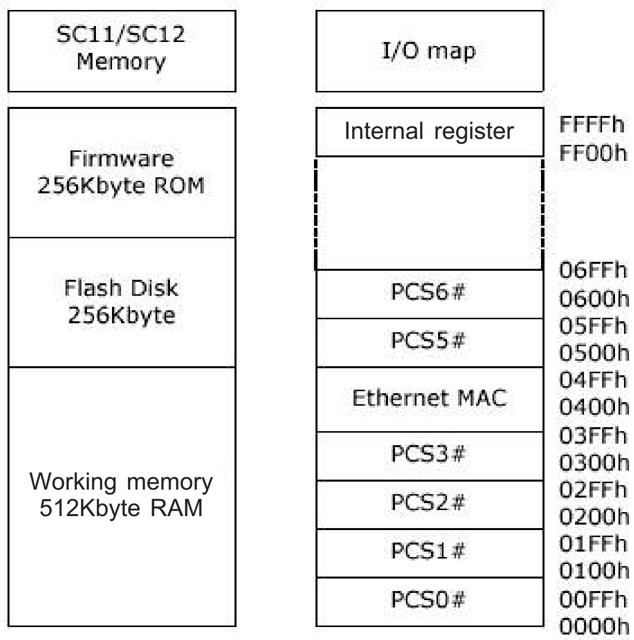

Figure 3 System memory configuration and the basic I/O of SC12

for network embedded system. Through the protocol API, network application can be developed to achieve desired network solution. Some examples of protocol API are shown in Table 1.

The implementation of network application involved PPP client and TCP server. The PPP client is responsible to perform modem dial-up to local ISP, and establish internet connection. TCP server is loaded automatically into the network embedded system and opens a port for listening mode. The TCP server is ready to accept connection request from the remote terminal.

Table 1 Some of the protocol API from the network embedded system

Interrupt	Function	Description
0 × AC	0 × 01	API OPENSOCKET, open a socket.
0 × AC	0 × 02	API CLOSESOCKET, close a socket.
0 × AC	0 × 03	API BIND, bind TCP or UDP server socket.
0 × AC	0 × 04	API CONNECT, connect to another socket.
0 × AC	0 × 05	API RECVFROM, receive message.
0 × AC	0 × 06	API SENDTO, transmit a datagram.

4.1 Local Connectivity Toolkit

Local connectivity toolkit is connected to the network embedded system through RS-232 connection. The functions of the connectivity toolkit are: listen to the incoming data, analyze the data, and perform appropriate operation. The connectivity toolkit is controlled by a micro controller to enable data processing mechanism. The implementation of system architecture for connectivity toolkit is quite straightforward. The operation block of connectivity toolkit is illustrated in Figure 4.

A system program determines the operation of a microcontroller. The implementation of a system program is a significant task to make the system work. Assembly language is used to implement a system program. Instead of using C language, assembly language is more flexible in creating a control purpose system program. The system software of the connectivity toolkit should be able to detect and prevent any unexpected dead-loop, to ensure the system stability and reliability. A timer is set to recover original instruction pointer, once idle time has expired. Process dead-loop caused the idle time to expire. This is very important to ensure that system operation can be recovered, once system failure is found.

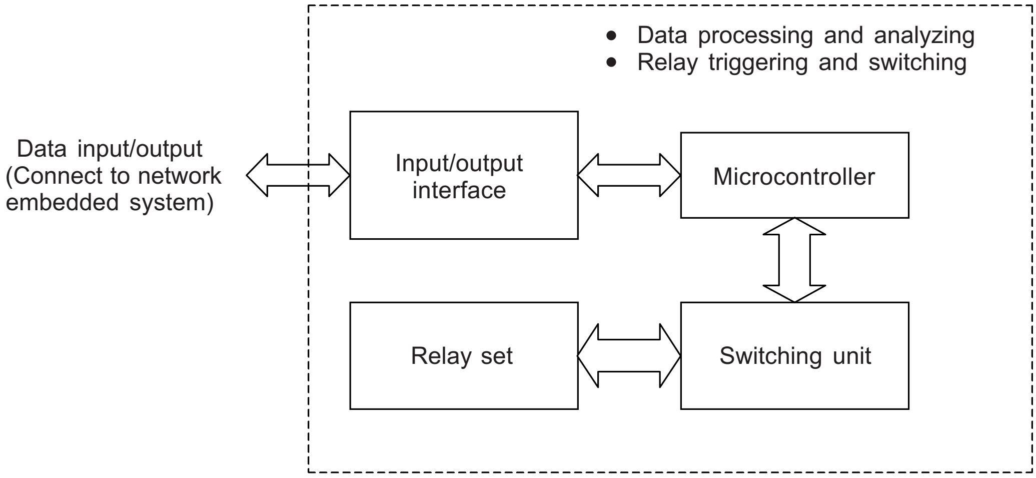

Figure 4 System architecture block diagram for connectivity toolkit

4.2 Client-Side Development

When two computers are communicating over a network, the one that waits for the connection is acting as a server, and the one that initiates the connection is a client [8]. High efficiency is determined by the transfer speed of the control signal from the client-side to the network embedded system. In the case of real-time control system, TCP should be the best choice for the client-server connection.

Graphic User Interface (GUI) is implemented to ensure the usability and user-friendly of the client-side. Windows socket enable the client-side development in the window environment. To access the windows socket, a number of ActiveX objects is used. Visual Basic provides a very-easy to-use IDE for users to program appropriate codes to be used in different applications. Winsock, one of the Active-X components that comes with the Visual Basic, is integrated into the network program. Through the Winsock component, a user can customize the remote IP address, port number, and the protocol type to be used, to configure the client-side. As a result, the client-side is able to initialize connection, authorize session, and launch data transfer.

4.3 System Integration and Optimization

System integration involves the combination of network embedded system, connectivity toolkit, and remote terminal. Network embedded system and connectivity toolkit are integrated locally but the remote terminal resides elsewhere. For testing purposes, a cross-link cable was used to establish a logical linking between the remote terminal and the network embedded system. Logical linking between network embedded system and connectivity toolkit was performed by RS-232 connection.

System optimization is significant to maintain the working performance of the system. In order to optimize real-time connection-oriented embedded remote triggering and monitoring system, there are several points which should be emphasized:-

(1) Power supply
(2) Heat problem
(3) Software design

Power supply is necessary to put into consideration in detail. Unstable power supply will cause any unexpected problem which is hard to be traced and debugged. For connectivity toolkit, a high quality switching power supply is used to make sure that the DC output is stable and free of any unwanted ripple, that will severely affect the operation performance. Head problem should be noticed as precaution for system design and implementation. Selection of operation voltage is important to make sure it is optimized. Operation voltage that is located out of the range of optimized voltage will cause system disable or overhead. For connectivity toolkit, 24 VDC is the most suitable to be applied for the best performance. System optimization of software design involves code optimization and error checking. Code optimization is necessary to make sure that instructions can be ran in the fastest time.

5.0 RESULT

Once the network cable is plugged into the RJ-45 socket, the pre-defined network configurations automatically launch the network service. These configurations include IP address, subnet mask, gateway, and DNS server. As a result, the unit network module of the internet connectivity embedded triggering and monitoring system provides a transparent interface of the local system to the global network. Remote terminal to be connected to the network embedded system can be launched in any platform or operating system; on condition that TCP/IP is applied to be the default network protocol.

The client-side enable user to establish connection to the server (network embedded system) by using IP address, and appropriate port number. In this case, the IP address of network embedded system is 10.6.11.3 and the application specific TCP server service is mapping to the port number 100.

Once the "*connect*" button is pressed, requesting for connection, an indicator prompts the status of connection. Once the connection is established, the user is allowed to manage the remote system as well as monitor its status. As a result, the status of the remote system is transparent to the user. If the first "*on*" button is pressed, the client-side will send a specific code to the remote network embedded system. Once the network embedded system received the code, translation process initialized to process the code before transferring to the local connectivity system toolkit. As the system toolkit received the code, the microprocessor performed appropriate operations to trigger corresponding relay that resided in the system toolkit. Figure 5 shows the operation of triggering first relay in the remote system toolkit. The operation is similar to trigger on any of the relays. The same mechanism is applied to trigger off the relays.

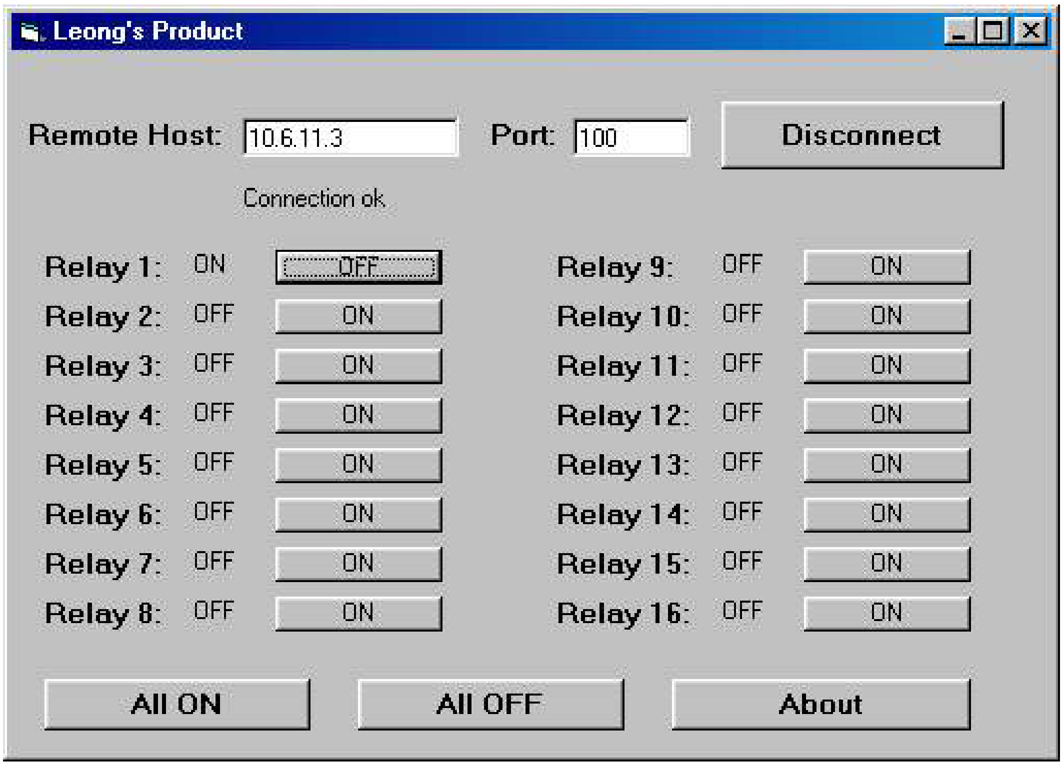

Figure 5 The operation of triggering first relay in the remote system toolkit

5.1 Respond Time

For effective remote triggering and monitoring purposes, the respond time is significant to enable real-time application. There are various types of connection method to allow transformation from source to destination. Each type of connection method owns its proprietary transfer rate according to the physical features of the transfer medium. Although TCP is applied for client-server implementation, the respond time still existed for the remote system. An analytical result in Table 2 shows a list of respond time according to the type of network connection. In LAN environment, 100-Base-T shows the best performance. However, 384 kbps ADSL shows the best environment in WAN environment.

Table 2 Respond time according to the type of network connection

Network	Connection	Respond Time (s)					
		1	2	3	4	5	Average
LAN	10-BaseT	0.2	0.3	0.2	0.1	0.2	0.2
	100-BaseT	0.2	0.2	0.2	0.2	0.2	0.2
WAN	56k dial-up	2.3	2.6	2.4	2.5	2.8	2.5
	64k ISDN	2.0	2.2	2.0	1.9	1.8	2.0
	128k ISDN	1.2	1.3	1.2	1.2	1.2	1.2
	384k ADSL	0.4	0.5	0.5	0.5	0.4	0.5

6.0 CONCLUSION

The capabilities of internet enabled embedded system for remote triggering and monitoring were demonstrated. This system provides three significant contributions: remote service activation through PPP, up to 16-operation channels per module, and integrated run-time error detection feature to enhance the capability and quality of the system operation. Data transformation mechanism was also proved with comprehensive result bi-directionally at run-time operation.

ACKNOWLEDGEMENT

The authors would like to thank the Institute of Advanced Technology, Universiti Putra Malaysia, and all the staff in Intelligent Systems and Robotics Laboratory, especially Mrs Rosiah Osman, Mr. Muhamad Saufi Mohd Kassim, Mrs Juraina Md Yusof, and Mr. Mohd Hussein.

REFERENCES

[1] Agranat, I. 1998. *Embedded Web Servers in Network Devices Communication System Design.* July 2004. 30–36.
[2] Case, J., M. Fedor, M. Schoffstall, and C. Davin C. 1990. *The Simple Network Management Protocol (SNMP).* RFC 1157. May.
[3] Taylor, E. D. 1998. *TCP/IP Complete*. McGraw-Hill.
[4] BECK-IPC "*Embedded Web Server*" http://www.beck-ipc.com Accessed on October 2004.
[5] Howard, D. Reforming. 1998. Data Acquisition Over the Internet. Sensor Magazine. January 1998.
[6] Franklin, G. F., J. D. Power, and M. Workman. 1997. *Digital Control of Dynamic System*. Addison Wesley Publishing.
[7] Martin-Flatin, J. P. 2000. Web-Based Management of IP Network and Systems. *Ph.D. thesis*. Swiss Fed. Inst. of Technology, Lausanne (EPFL). October.
[8] James, G. 1997. *Local Area Network: A Client/Server Approach.* John Wiley & Sons, Inc.