Embedded System and Remotely Controlled Applications

Muneer Ahmad Dar[1], Dr. Deepshikha Jamwal[2]

[1] *Scientist-B, DOEACC Centre Srinagar/Jammu, J&K India*

[2] *Assistant Professor, Department. of Computer Science & IT*

University of Jammu, J&K India

[1]muneerdar07@gmail.com

[2]jamwal.shivani@gmail.com

***Abstract*— An embedded system is a special-purpose computer designed to perform one or a few dedicated functions, often with real time computing constraints. Embedded systems have become very important today as they control many of the common devices we use. An embedded micro controller is a chip, which has a computer processor and all its support functions, memory (Both program and data) and I/O (including bus interfaces) built within the device. The paper proposes a RCM (Remotely Controlled Model) which discusses significance of using Micro Controller to remotely control maximum number of devices so that the applications of Embedded System can be improved by connecting more devices to the Micro controller.**

***Keywords*— Embedded System, Microcontroller, RCM (Remote Controlled Model), UART (Universal Asynchronous Receiver/Transmitter), GUI (Graphical User Interface).**

***Objectives*—**

1. The basic objective of this paper is to remotely control maximum number of devices with a single 8051 Microcontroller. Efforts have been taken to control the various devices which include Stepper Motor, LCD, Keyboard and Traffic System etc

2. Through understanding of Embedded Systems, Attributes of Embedded System and challenges for an Embedded System.

1. INTRODUCTION

Embedded systems are found in a large number of applications. For example, today even low-end cars contain several dozen processing elements that take care of various elements of control, such as fuel-injection, anti-lock braking systems, temperature and seat-comfort, and navigational assistance, among other things Embedded systems include a variety of hardware and software components, which perform specific functions in host systems, for example, satellites, washing machines, robots, hand-held telephones and automobiles.

This paper is a design and implementation of "Remotely controlling Multiple Applications Controlled by a Single 8051 Micro-Controller using Real Time Operating System". 8051 Microcontroller is interfaced with multiple devices and is serially connected to the server. RS232 chip is used for serial communication between the server and the Microcontroller. RS232 is basically used for voltage conversion between PC and the microcontroller. Since many devices are used, the concept of multitasking is used. Tiny RTOS schedules and controls the operation of these devices. TCP/IP protocol is used for client/server communication. The client is first authenticated and then given control of devices. The client can control devices by simply sending commands.

Whether these benefits are in concerned with security purposes or with human's comforts, in fact these intelligent systems helped humanity in every phase of life. They have helped man in their progress by developing the new technologies [1].

1.1 ATTRIBUTES OF AN EMBEDDED SYSTEM:

As embedded systems started progressing, they started becoming more and more complex. Additionally, new attributes that got added to these systems were smart and intelligent. Not only were the embedded devices able to do their jobs but also were able to do them smartly.

1.1.1 COMPUTATIONAL POWER: These devices have some amount of computing power. A very simple 8-bit controller or a high-end 64-bit microprocessor could provide this computation power [2].

1.1.2 MEMORY: The next requirement is memory. These devices possess some amount of memory that can be used by the processor and also some to remember user data and preferences.

1.1.3 REAL TIME: All the devices have to respond to user/environmental inputs within a specified period of time.

1.1.4 COMMUNICATION: The device must be able to receive inputs given by other devices in the environment, process it and provide some solid output to the other devices or users.

1.1.5 DYNAMIC DECISIONS: The system should be able to change its next course of activity based on the change of input from its sensors or surroundings [3].

2. CHALLENGES FOR AN EMBEDDED SYSTEM

The embedded processors are getting more and more powerful. It is not uncommon to find embedded systems with powerful processors, like mobile (GSM/GPRS) handsets, high-speed routers, bridges etc. Programming for the

designing of such complex devices offers unique challenges not found in PC/workstations based applications. Some of them are listed below [4]:

2.1 LIMITED OPERATING SYSTEM SUPPORT FOR PROGRAMMING: Application programs for PCs/workstations are launched from the operating system. The tasks like memory management, scheduling, hardware abstractions and input/output from/to peripherals are delegated to OS. All these tasks are handled by the operating system in a PC environment. In embedded systems, the OS is part of application code and it closely co-ordinates with the OS to support a majority of the features that a desktop OS may provide.

2.2 LIMITED SECONDARY MEMORY: Many embedded systems do not boot from a hard disk. They depend on other types of non-volatile memory like ROM, FLASH memory, instead of secondary memory devices. As systems with 16 MB flash are considered Premium, therefore our code and data sizes must be small.

2.3 LIMITED RANDOM ACCESS MEMORY: Since embedded systems inherently operate with restriction on resources (limited resource could be provided), we usually don't have the concept of swapping and virtual memory etc., in embedded systems. Therefore, while programming for embedded systems, we must be very careful about the memory leaks because these programs tend to run forever, and even a single byte leak in some path of execution will definitely bring the system to a grinding halt at a later point of time [5].

2.4 LIMITED PROCESSING TIME: We cannot afford to have a powerful processor for an embedded system because of obvious cost considerations. Therefore we may have to work with some microcontrollers with less powerful configurations. So, the code written must be efficient. We have to choose appropriate algorithms and cannot choose an algorithm with high computing requirements unnecessarily.

2.5 INTERACTION WITH HARDWARE: This factor singularly differentiates a normal application programming from embedded programming. An application programmer using the most modern operating system can develop software, by remained unaware of the underlying hardware. But, an embedded programmer usually cannot afford this level of hardware independence since its code directly interacts with the underlying hardware. Embedded programmers usually have to work with real-time operating systems that generally cannot provide such a high level of abstraction over hardware due to space and time restrictions [6].

2.6 ABSENCE OF STANDARD INPUT / OUTPUT DEVICES: A PC has standard I/O devices like keyboard, mouse and a display that can be used to display, what's happening inside our program. But many of the embedded devices do not have such I/O devices. So, a programmer has no direct way of knowing what is happening within the system. This seriously limits the amount of debugging possible on an embedded system [7].

3. APPLICATIONS OF EMBEDDED SYSTEMS:

Embedded System Applications describes the latest techniques for embedded system design in a variety of applications. This also includes some of the latest software tools for embedded system design. Some of the other examples of embedded systems are as follows:

Automatic teller machines, Computer printers, Disk drives, Cellular telephones and telephone switches, Inertial guidance systems for aircraft and missiles, Medical equipments, Video game consoles, Industrial machinery use programmable logic controllers to handle automation and monitoring, Engine control computers and antilock brake controllers for automobiles, Wristwatches, Household appliances, including microwave ovens, air conditioners, iron, and television sets, Home automation products, like thermostats, sprinkler, and security monitoring systems, Network equipment, including routers and firewalls, Traffic control (e.g. intelligent traffic lights) [8].

4. DESIGN AND IMPLEMENTATION

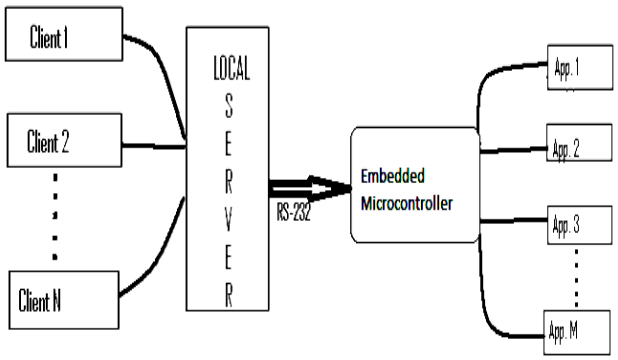

Fig 1. Implementation of Remotely Controlled Model

A software design is a model of real world systems that has many particular entities and relationships. This design is used in a number of ways. It is a basis for implementation; it serves as a communication media between the designers of subsystems; it provides system information to system maintainers about the original intentions of the system designers

The Remotely Controlled Model shown above includes N number of clients with each client using TCP/IP and can access the server for the authentication. After getting the authentication a particular client can access a particular application.

The designing of each module shows how each of the modules is to be carried out to satisfy all the conditions. Every module has specific functions to be carried out for its proper operation. To carry out all the operations proper designing of each module is essential before the implementation. The design of each module can easily be understood by data flow diagrams, which will show step-by-step procedure in which each module is to be carried out.

4.1 GUI BASED AUTHENTICATION

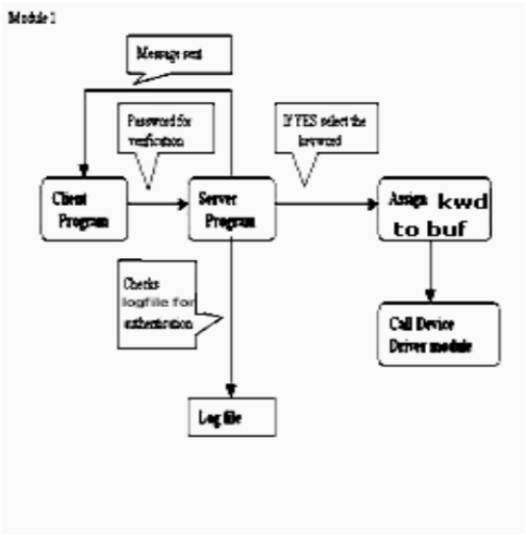

Fig 2. Implementation of GUI based authentication.

The basic function is to provide authentication remotely. The remote authentication should eliminate the requirement of the authenticated persons to be present at the physical location where controlling operation is required.

4.2 CONFIGURING THE UART OF PC AND MICROCONTROLLER:

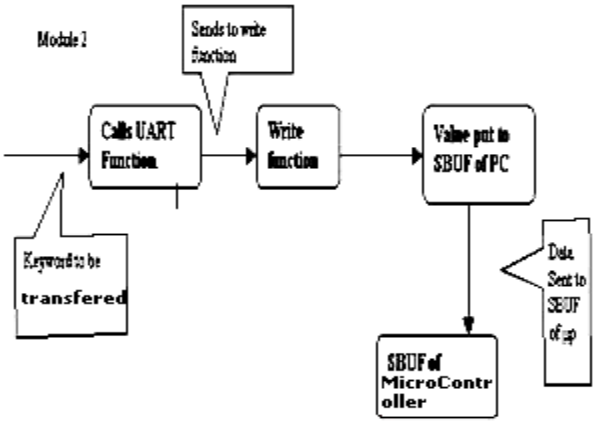

Fig 3. Configuring the UART of PC & Microcontroller

This module's basic function is to provide communication link between the PC and the Microcontroller. This is enabled by Serial communication.

4.3 THE 8051 FUNCTIONING UNIT AND HARDWARE INTERFACE:

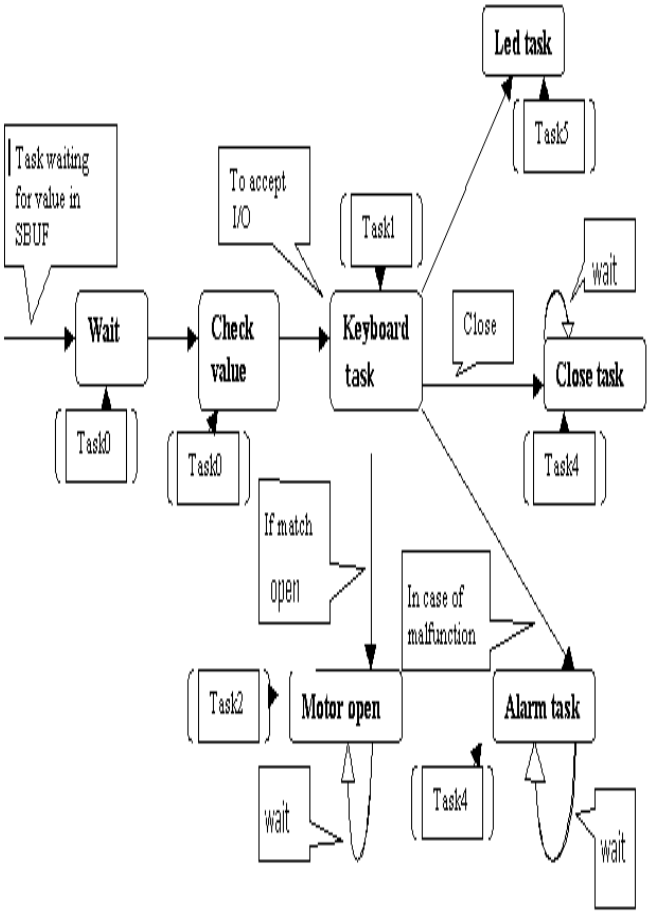

Fig 4. Functioning unit and Hardware Interface.

This module's basic function is to handle the interfaced devices for operation through multitasking. The devices interfaced are the keypad, the Motor, the Elevator, the LCD display, the Traffic system and the A.C. Bulb. These devices are multitasked. Once the second module activates the 8051 hardware unit, the keypad will provide codeword authentication. Once authenticated all the devices will be activated which will perform the running and stopping of the motor and the LED display. The Microcontroller will act, as the real time system once the application is web enabled. The software, which will handle its operation, should be able to handle time critical tasks. Hence RTOS is the software used.

5. Conclusion

As embedded systems play an important role in our daily life work so the author looked into the details of the embedded systems and discussed its characteristics, its components, the challenges programmer faces while designing such systems, its practical implementations, its importance in daily life and its future prospects.

In this paper, we presented Remote Controlling strategy as lower-cost, stable and simple technique for controlling multiple applications. We introduced the design and implementation of the RCM, a Remotely Controlled Model which can improve the performance of Microcontroller and in turn in the Embedded System applications.

6. Future Prospects

In future, the RCM (Remotely Controlled Model) proposed by the authors can be improved by connecting more Microcontrollers to the server and more applications can be controlled remotely.

Acknowledgement

This work is partially been supported by DOEACC Centre Srinagar and University of Jammu. The authors would like to extend their thanks to IT companies and professionals for providing data, with the help of which the paper is framed.

References

[1]. Ohhoon Kwon et.al, "Swapping Strategy to Improve I/O Performance of Mobile Embedded Systems Using Compressed File Systems", ISSN : 1533-2306 , ISBN: 978-0-7695-3349-0, IEEE, 2008.

[2]. Suarez, F.J. et.al, "Performance debugging of parallel and distributed embedded systems", ISBN: 0-7695-0634-8, page(s): 135 – 149, IEEE, 2000.

[3]. Jen-Hao Teng et.al, "Integration of networked embedded systems into power equipment remote control and monitoring", ISBN: 0-7803-8560-8, IEEE, 2004.

[4]. Tan, T.K. et.al, "EMSIM: an energy simulation framework for an embedded operating system", ISBN: 0-7803-7448-7, IEEE, 2002.

[5]. Ramamritham, K et.al, "System support for embedded applications", ISSN: 1063-9667, ISBN: 0-7695-1868-0, IEEE, 2003.

[6]. Hall, T.S. et.al, "A Novel Approach to an Embedded Systems Curriculum", ISSN: 0190-5848, ISBN: 1-4244-0256-5, IEEE, 2006.

[7]. Stephanie White, Mack Alford & Julian Hotlzman, "Systems Engineering of Computer-Based Systems." In: Lawson (ed.), Proceedings 1994 Tutorial and Workshop on Systems Engineering of Computer-Based Systems, IEEE Computer Society, Los Alamitos CA, 1994, pp. 18-29.

[8]. Payne, William (19). Embedded Controller Forth for the 8051 Family (hardcover) (in English), Elsevier, 528. ISBN 978-0125475709.

AUTHORS

1. Muneer Ahmad Dar (Bsc, MCA, M.Phil) currently working as Scientist-B in DOEACC Centre Srinagar/Jammu, Srinagar (Formerly CEDTI). Having development and teaching experience of around six years.

2. Dr. Deepshikha Jamwal, (Diploma In Electronics & Communication engineering, Bsc. BCA, MCA, M.Phil, Ph.D), currently working as Assistant Professor. in Department of Computer Science & IT, University of Jammu, having teaching experience around four years. Number of publications is around seventy-five in various reputed journals, International conferences & National Conferences.