Implementation and Simulation of a Multiprocessor System Using 8051 Microcontroller for Industrial Applications

Toka Abdulhameed Fathi[1], Hiba Nabeel Yahya[2], Qutaiba A. Hasan[3]
{ toka.fathi@uoninevah.edu.iq, heba.yahya@uoninevah.edu.iq}

Computer & Information Engineering, Dept.Ninava University, Mosul, Iraq[1,2]
Tikrit University / Petroleum and Minerals Eng. College[3]

Abstract. The paper aims to implement and simulate a prototype of industrial multiprocessor communication systems through serial communication RS-232 protocol using an 8051 microcontroller. The created model is used for experimental purposes by the microcomputer trainer (MTS-51). Many simulators are present for the 8051 microcontrollers, many of them do not support multi microcontroller communication, or simulate all the embedded peripherals. A few of them can simulate external peripherals. The proposed simulation has been accomplished using the PROTEUS program. In this work, the RS-232 serial mode of communication using an 8051 microcontroller is used, with an emphasis on interfacing it with an LCD screen and a hex keypad, stepper motor, speaker and a seven-segment display. The operation of several microcontrollers in the master-slave configuration is explained. It is verified that both the stepper motor and speaker is controlled at each slave using the master controller. The used communication protocol is a very simplified one, highlighting how simple such communications are accomplished in high-tech industrial environments. Moreover, perfecting the use of the RS-232 for transferring small blocks of information has resulted in simple use and support of the maximum number of communicating processors.

Keywords: multiprocessor communication, 8051 microcontroller, RS-232 protocol, electronic peripherals.

1 Introduction

A typical microcontroller includes a central processing unit (CPU), input/output (I/O), and a random access memory (RAM) interfaced in one package. On-chip analogue-to-digital converters (ADC), timers, and low power consumption may also be included. These features make the microcontrollers flexible enough to be used in an extensive variety of applications.

The integration of many components in a small piece makes the microcontroller easier to use than a general processor and require less wiring and connections while doing the same function[1].

RS-422, RS-485 and RS-232 can accomplish serial communication. Among these protocols, RS-232 is used in this paper due to its simplicity and the maximum number of communicating processors that can be supported. |A central control system and sensor systems send updates to the product data terminals that might use this feature [2].

LCDs and keypads are generally used to interface microcontrollers to the external world. Low cost and simplicity of use make the 8-bit microcontroller to be widely used more than 16-bit or 32-bit microcontrollers [3].

It is known that current industrial environments use multiprocessor communications as an integral part. There are many associated pieces of research on this topic, some of which are listed below:

IMDC-SDSP 2020, June 28-30, Cyberspace

DOI 10.4108/eai.28-6-2020.2298126

In [4], the ATMEL 89C51 microcontroller is used. A simplified prototype of multiprocessor communication systems is discussed via the RS-485 Serial Communications Protocol. The authors built and simulate a model for experimental uses. The Liquid Crystal Display(LCD) interface and the Hex keypad are interfaced. They elaborate a master-slave configuration of three microcontrollers.

A simplified model of industrial multiprocessor communication systems has been discussed in [5] where an ATMEL 89C51 (8051) microcontroller has been used with RS-232 serial communication protocol. The model has been constructed and simulated for experimental purposes. ADC0804 has been connected to an LM35 Temperature Sensor and a one way DC fan then interfaced to the microcontroller, they elaborate the operation of three microcontrollers in a master-slave configuration.

Researchers in [6] focused on the design of a communication layer and application programming interface for communications among microcontrollers. A small-scale cluster computer is built using Atmel ATmega32 microcontrollers as processing nodes and an Atmega16 microcontroller for message routing. The communication library is integrated into the operating system.

The use of multiple 8051s microcontrollers is presented in [7] and parallel operation of the 8051 microcontroller chip is implied using a multiprocessor 2 configuration.

In [8], the researchers presented a concept that implements a genetic algorithm (GA), in a multiprocessor environment. The goal of this research was to build a cooperative multiprocessor system by expanding the role of PIC microcontrollers to optimize WCDMA base station transmitter power. After implementation, results have shown that PDGA multiple microcontroller systems give better transmission power compared to conventional genetic algorithms.

A group of less capable interconnected and more specialized microcontrollers can replace the central processing unit CPU as the authors in [9] imply. These specialized microcontrollers are focused on specific tasks. Thus, unused processing hardware can simply be removed from the platform. This paper presented the results of an experimental study into the advantages of using multi microcontrollers sensor node construction.

The researchers in [10] interface a large number of peripheral devices such as ADC, LCD, Motors, Motor Relays, Keypad, RTC and seven-segment displays (FND) which is not possible with one microcontroller. The restriction of using one microcontroller are limited code memory, only limited peripheral devices can be connected, design complexity, software complexity and error probability.

The study shows that the wanted hardware configuration flexibility can be realized with relatively low overheads while supporting established sensor node programming concepts.

In this paper, Multiprocessor communication systems using an 8051 microcontroller are implemented and simulated for industrial applications via serial communication RS-232 protocol. Microcomputer trainer (MTS-51) is used to construct such a model for experimental purposes. The PROTEUS program is utilized to perform the simulation.

This paper presents the serial RS-232 communication mode concerning the use of the 8051 microcontroller. The operation of several microcontrollers in the master-slave configuration is described with an emphasis on interfacing them with an LCD screen and a hex keypad, stepper motor, speaker and a seven-segment display. The remainder of this paper is organized as follows: Section 2 includes a brief introduction to multiprocessor communication systems using an 8051 microcontroller via serial communication RS-232 protocol. Section 3 contains the external electronics peripherals that are connected to the microcontroller in different experimental projects. Section 4 consists of both hardware and software tools which are used to implement and simulate the same experimental projects. Section 5 deals with the implementation and simulation of such experimental projects. Finally, Section 6 covers the conclusions that are extracted from the experimental projects.

2 Multiprocessor Communication

The 8051 multiprocessing indicates multiple processors acting in some combined manner and associated so that data can be exchanged between them. There is generally a main or master microcontroller that guides the activities of the other microcontrollers, or slaves [2]. One of the 8051 microcontroller's numerous prevailing features is its built-in Universal Asynchronous Receiver Transmitter (UART) serial port. The 8051 microcontroller has an integrated serial port, which means that the handler may simply read and write data to the serial port. However, the user needs to set up the serial ports baud rate and operation mode. When the baud rate and operation mode is configured, all the operator has to do is write to a serial buffer (SBUF), to write something to the serial port or get the serial buffer (SBUF) to read something from the serial port. The 8051 microcontroller will be automatically known by the user when it has completed receiving or sending a byte to process it [1] [3]. Communication is through multiple 8051s using standard UART technology which assigns unique addresses to all the slaves 3 using mode. In this mode, the slaves will waste a lot of processing time rejecting data not addressed to them. Mode 2 and 3 reduce the processing time by enabling character reception based upon the state of SM2 in a slave and the state of the tenth bit in the transmitted character. A single strategy is used to enable a few slaves to receive data while the majority ignores the transmissions (see Figure1) [3].

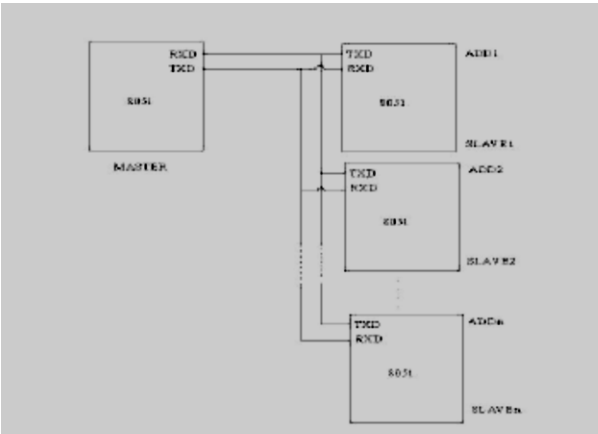

Fig. 1. **Master Slaves Communication**

3 External Electronic Peripherals

All microcontroller systems have a large number of peripheral devices which expand the microcontroller's range of application and make the process of programming testing easier. These devices are connected to microcontrollers through I/O ports. In this paper the following external peripherals are used, figure 2 shows an example of a microcontroller system with peripherals.

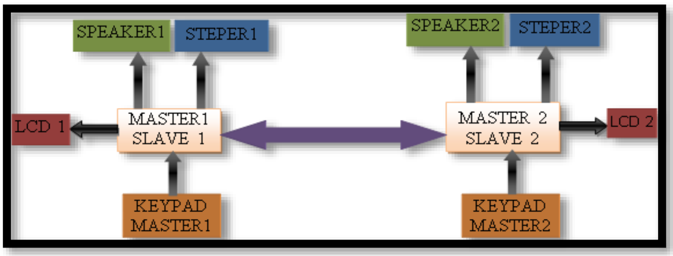

Fig. 2. Block diagram of master/slave with peripheral

3.1 Seven Segment Display

A seven-segment display is one of the most elementary electronic display devices that can display number digits from 0-9 [11].

3.2 LCD Display

The LCD screen is a very commonly used output device to display alphanumeric characters. It is very helpful in providing user interface as well as for debugging purpose. It offers high flexibility to the user, as it can display the required data [13].

3.3 Stepper Motor

A stepper Motor translates electrical signals into mechanical motion. In many applications such as dot matrix printers, disk drives, and robotics, the stepper motor controls the position [14].

3.4 Microcomputer Trainer

A microcomputer trainer (MTS 51) is a self-contained system designed for the study of single-chip microcomputers. A variety of I/O devices are suited for control 5 applications. Microcomputer trainer features include a serial port connector and RS232 interface for multiprocessor communication and communication with PCs for programming by using a flash program and debugging using a µ-vision program. In addition, it is equipped with a 7-segment display, an LCD, hex keypad, speaker, stepper motor and other I/O devices [16].

3.5 PROTEUS Program

The PROTEUS program allows engineers to emulate real designs of a range of simulator models for common microcontrollers and a set of animated models for related peripheral devices such as keypads, LCDs, LEDs, RS232 terminals and more. A complete microcontroller design can be simulated and its software can be developed without building a physical prototype. Using PROTEUS, the time for the design building and testing can be efficiently reduced [17].

4 Simulation and implementation of Multiprocessor System

In this paper, the experimental projects are implemented using the microcomputer trainer (MTS 51) and simulated using the PROTEUS program. The following subsections will be **mentioned in experimental projects.**

4.1 Single Master / Single Slave Communications

To create a multiprocessor communication system in a master-slave configuration, and to increase the number of I/O devices that can be controlled by the 8051 in multiprocessor mode, two 8051 microcontrollers in master-slave configurations are connected and this means the direction flow can be from one to other.

In this experimental project, two 8051s are connected and each one has an LCD screen to display a specific message, a stepper motor, a speaker as a controlled device, and hex keypad to assign which peripheral will be controlled. At each 8051, if key 0 is pressed, the current 8051 will be the master and running a stepper motor on its board, and its LCD screen, in this case, will show the following message “master 1 run step”. If key 4 is pressed, the master will send a byte of data that tells the slave 8051 to run the slave’s stepper motor and display on the master’s LCD the following message, “master1 send step”. Finally, if key 8 is pressed, the master will send a byte of data that makes the slave 8051 to run the slave’s speaker and on the master’s LCD the following message will be displayed “master1 send speaker”.

If there are no keys pressed, the 8051 will act as a slave and wait for the RI signal. If this signal is detected, it will become a slave. After verifying its address, the slave 8051 checks the received data byte to see if it is 4 or 8 value. If the data byte is 4, the slave will run its stepper motor and its LCD1 will display “slave 1 received step”. If the received data byte is 8, the slave will run its speaker and its LCD1 will display “slave 1 received speaker”.

The flowchart of this experimental project is shown in Figure 3.

Figure 4 shows a circuit diagram of master/slave with peripheral, while Figure 5 shows the experimental project of master/ slave with peripherals.

4.2 Single Master / Multi Slaves Communications

This experimental project is similar to the previous project, but the master 1 can control a stepper motor and speaker in both slave 2 and slave 3, where this depends on the state of a pressed key. The visual 6 display of the communication scheme is shown in Figure 6. So, if key 0 is pressed master1 will run its stepper motor and the LCD1 will display “master1 run step”. If key 1 is pressed, master 1 will run its speaker, while the LCD in this case, will show this message “master1 run SPK”. If key 2 is pressed, the master1 will send a data byte to slave2 telling it to run its stepper motor. Then the master’s LCD1 will display “master 1 send step to slave2”. If key3 is pressed, master1 will send a byte to slave 2 to run slave’s speaker, then the LCD1 will show “master1 send SPK to slave2”. Though keys 4 and 5 are used to send data to slave3, slave1 can receive data from other slaves and check if it is used to run its stepper motor or speaker. Slave 2 and slave 3 receive the first byte sent by master1 and they also check it to constrain which device should run, however, these two slaves can only send data to master1. Both slave’s LCDs will show a message that declares which operation should be done by 8051. The flowchart of master/ multi slaves with peripherals is shown in Figure 7.

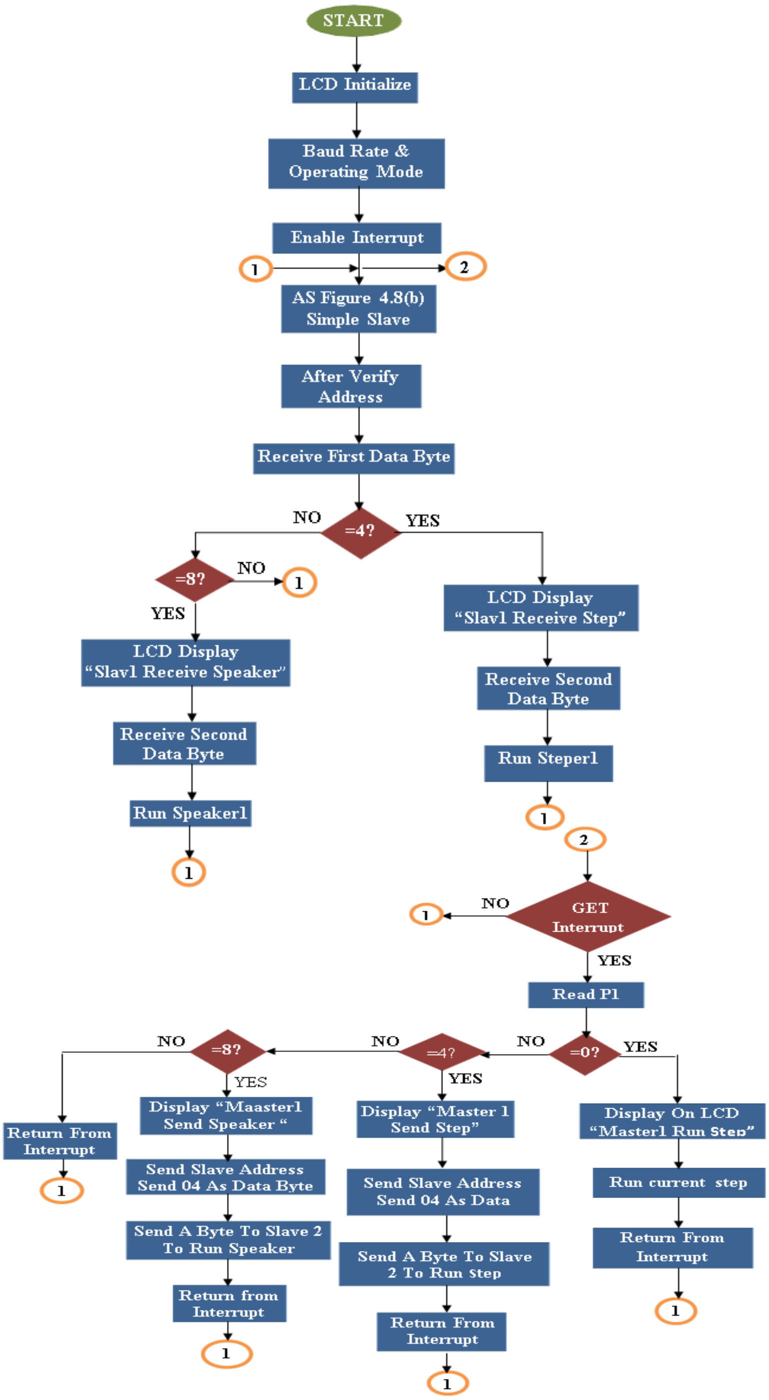

Fig. 3. The flowchart of this experimental project

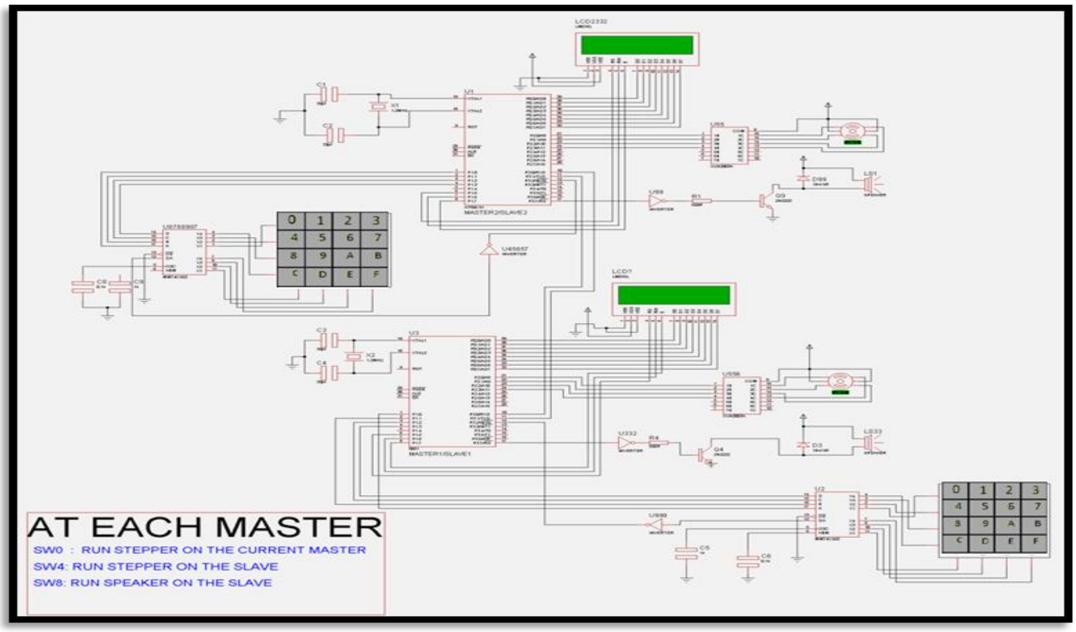

Fig. 4. **the circuit diagram of the master-slave**

Fig. 5. **Experimental project of master/ slave with peripherals**

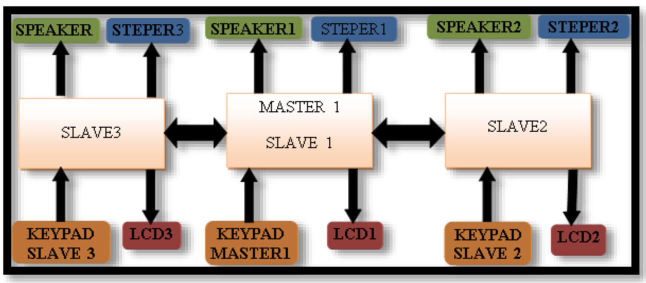

Figure 6. Block diagram of master/multi slaves with peripherals

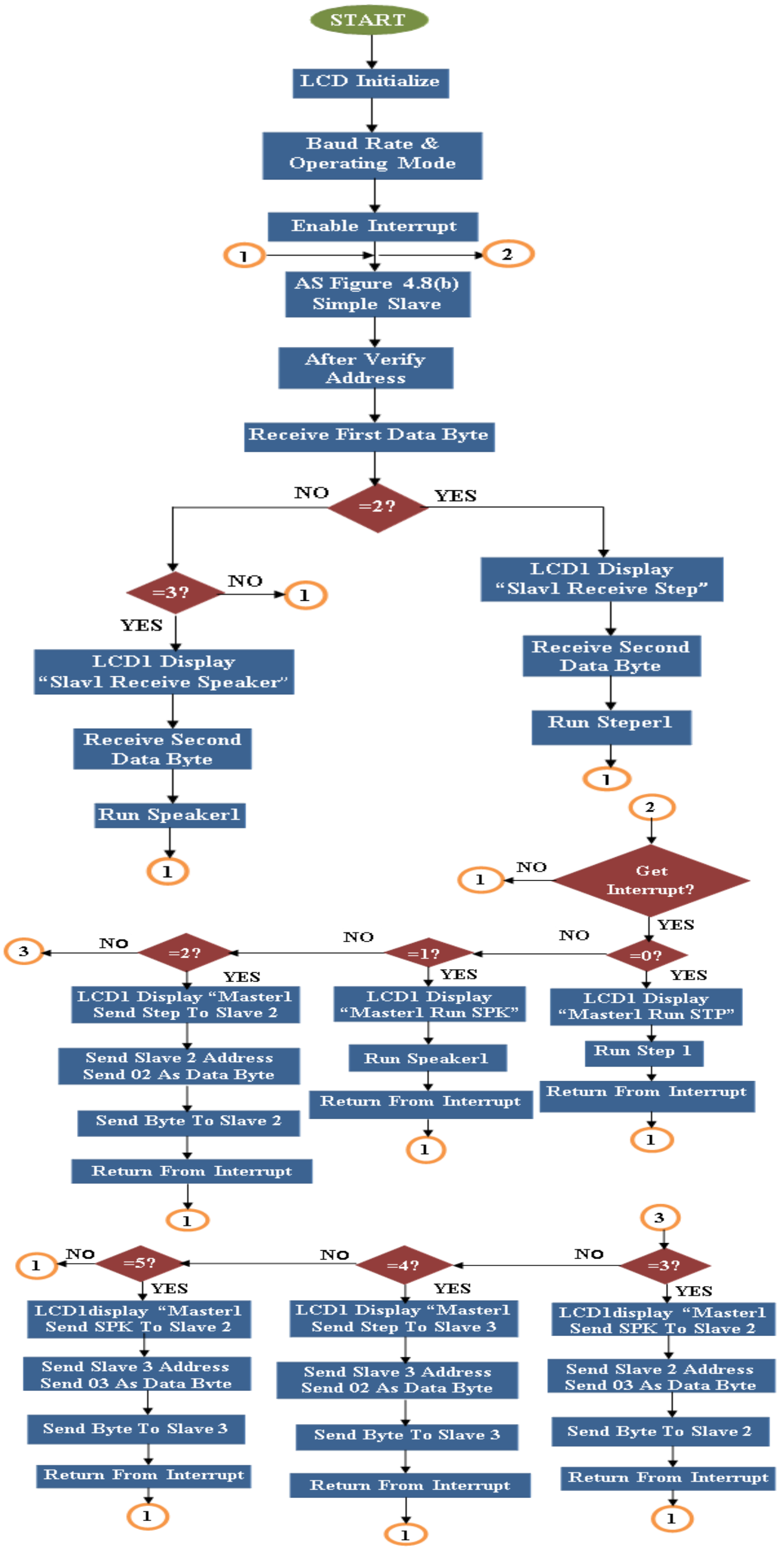

Figure 7. The flowchart of of master/ multi slaves with peripherals

Conclusions

Communication between multiprocessors is an important part of industrial communications. A prototype of an industrial multiprocessor communication system through serial communication RS-232 protocol using an 8051 microcontroller has been built.

The experimental projects have been simulated using the PROTEUS program installed on an Intel Core i3, 2.27GHz CPU and implemented on a microcomputer trainer (MT51). It has been found that both the stepper motor and the speaker can be controlled at each slave using the master controller. The used communication protocol is a very simplified one highlighting how simple such communications can be accomplished in high-tech industrial environments. Moreover, perfecting the use of the RS-232 for transferring small blocks of information has resulted in the simple use and support of the maximum number of communicating processors.

Multiprocessor communications are an essential part of recent industrial environments as nowadays communications are of much more important than ever. The developed hardware is a prototype of the communication models used in manufacturing environments, with master controllers representing the control from where commands and instructions are sent to different parts of the industrial setting, and slaves representing the role of places where these commands are sent to be executed.

References

[1] K.J.Ayala, “The 8051 Microcontroller Architecture, Programming, and Applications”, West Publishing Company, 1991.

[2] D. Calcutt, F. Cowan, and H. Parchizadeh, "8051 Microcontroller An Applications-Based Introduction", Newnes, 2004.

[3] Muhammad Ali Mazidi, Janice Gillispie Mazidi, and Rolin D. McKinlay, “The 8051 Microcontroller and Embedded Systems”, Prentice-Hall Inc., 2009.

[4] Kashif Altaf, and Javaid Iqbal, "Multiprocessor Communication using 8051 Microcontroller and RS-485 Line Driver", 2003.

[5] Mohsin Murad and Uzair Kamal, "Multiprocessor Communication using 8051 Microcontroller and RS-232 Interface", 2006.

[6] Kalim Moghul, “Design of a Message Passing Interface for Multiprocessing with Atmel Microcontrollers”, Project Report, Master of Electrical Engineering Program, Cornell University, May 2006.

[7] Ammar Ibrahim Majeed, Husam Juma’a Neamaa, Riyadh Ali AL-Helali, Majid Eddan, “Multiprocessor Configuration of 8051microcontroller Chip”, Electrical Engineering Dept., ALMustansiriyah University, 2007.

[8] K. Prajindra Sankar, S.K. Tiong, and S.P. Johnny Koh, “Parallel Distributed Computational Microcontroller System for Adaptive Antenna Downlink Transmitter Power Optimization”, International Journal of Electrical and Computer Engineering 4:11, Electronic and Communication Engineering Dept., University Tenaga Nasional, Malaysia, 2009.

[9] Utz Roedig, Sarah Rutlidge, James Brown, Andrew Scott, “Towards Multiprocessor Sensor Nodes”, Lancaster University, Killarney, Ireland, June 2010.

[10] SM. Kalyankar1, SD. Sawarkar, “Design and Development of Reconfigurable Multiprocessor Architecture for Embedded Systems “, IJSRNSC Volume-5, Issue-1, April-2017.

[11] http://en.wikipedia.org/wiki/Display_examples

[12] http://www.datasheetarchive.com/4x4% 20numeric%20keypad-datasheet.html

[13] http://www.8051projects.net/lcdpinout/i
ntroduction .php.

[14] http://what-whenhow. com/8051microcontroller/steppermotor- interfacing.

[15] http://www.scribd.com/doc/72229934/Speaker.

[16] “MTS-51 microcomputer trainer”, Web-Site: www.kandh,com.tw.

[17] Sani Irwan Bin Md Salim Fkekk, "Proteus VSM 6.9 Schematic Simulation Quick Guide", UTeM, http://www.readbag.com/utemmy- myweb-sani-pdf-quickguideproteus, Jan. 2008.