Design and Implementation of Multiprocessor Communication In Embedded Processors for Real Time and Industrial Automation

Dr.Vasuki

Department of Electrical and Electronics Engineering

Achariya College of Engineering Technology, Puducherry.

Abstract- Multiprocessor implementation is gaining considerable significance in modern day embedded systems. In systems where more than one processor is used the need for multiprocessor communication occurs. The various motivations for using multiprocessor communication are

- Increased performance
- Increased reliability
- Functional modularity
- Physical distribution of processing power

In this paper, multiprocessor approach is implemented for a real time boiler plant control application. A simple prototype for industrial multiprocessor communication system using ATMEL 89C51 microcontroller via serial communication RS-232 protocol is discussed. In this communication scheme failure in communication link is detected by using microprocessor-fault detection protocol.

I.INTRODUCTION

Multiprocessing is traditionally known as the use of multiple concurrent processes in a system as opposed to a single process at any one instant. A critical problem in multiprocessors is managing communication between the processors in the system.

Individual processor performance is important, but it alone is not sufficient to achieve scalable performance. In an industrial stetting where there are two or more interrelated processor system, the need for multiprocessor communication is indispensible.

The microprocessors or microcontrollers may communicate via many protocols e.g.RS-232, RS- 422, RS-485. 8 bit microcontrollers are used as they are simple and easier to use, also due to their low cost. To interface the device to the outside world, input and output devices like the keypad and LCD are used.

In this paper, the multiprocessor approach where RS232 mode of communication using the 8051 microcontroller is the main concern. In the boiler plant control application three slaves and a Master controller are used. The temperature of the boiler is monitored by the first slave. The Master processor queries the slave1 processor for information and analyzes the temperature data. If the temperature of the boiler exceeds specific limit then the Master processor sends appropriate command to Slave3 for switching off the Heater through a relay circuit. On the other hand if the temperature of the boiler falls below the lower limit then the Master processor sends command to Slave3 to Switch on the heater. Another slave, slave2 is used to measure the pressure in the boiler.

II. RS-232 PROTOCOL

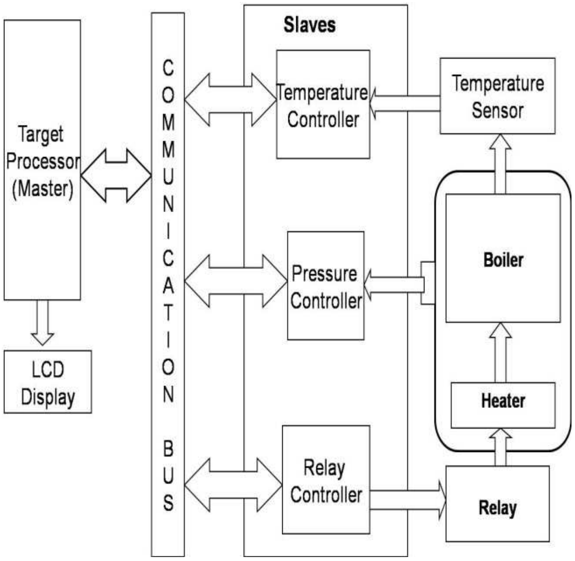
RS-232 is a serial port which is simple, universal, well understood and supported. The standards to 256kbps or less and line lengths of 15M (50 ft) or less. The Serial Port is harder to interface than the Parallel Port. In most cases, any device connected to the serial port will need the serial transmission converted back to parallel so that it can be used. This can be done using a UART. Details of character format and transmission bit rate are controlled by the serial port hardware, often a single integrated circuit called a UART that converts data from parallel to asynchronous start-stop serial form. The voltage levels, slew rate, and short-circuit behavior are typically controlled by a line-driver that converts from the UART's logic levels to RS-232 compatible signal levels, and a receiver that converts from RS-232 compatible signal levels to the UART's logic levels. Devices which use serial cables for their communication are split into two categories. These are DCE (Data Communications Equipment) and DTE (Data Terminal Equipment).

Data Communications Equipment is devices such as modem, plotter etc while Data Terminal Equipment is the Computer or Terminal. The RS- 232 standard defines the voltage levels that

Correspond to logical one and logical zero levels. Valid signals are plus or minus 3 to 15 volts. The range near zero volts is not a valid RS-232 level; logic one is defined as a negative voltage, the signal condition is called marking, and has the functional significance of OFF. Logic zero is positive; the signal condition is spacing, and has the function ON. The standard specifies a maximum open-circuit voltage of 25 volts. RS-232 drivers and receivers must be able to withstand indefinite short circuit to ground or to any voltage level up to ±25 volts. Use of a common ground is one weakness of RS-232 slave microcontrollers AT89C2051 is connected. The main interest is to perform two way communications between masters – slave configuration. The master microcontroller is interfaced to the display device, the liquid crystal display (LCD). The slave1 is interfaced in parallel with the analog to digital converter ADC804 which in turn is connected to the temperature sensor LM35. The slave 2 on the other hand is interring faced also interfaced in parallel with the ADC804 which is connected to the pressure sensor. The slave 3 on the other hand is interfaced to the relay which is connected to the heater. The heater is used to heat the substance in the boiler and the temperature sensor records the temperature of the boiler, this electrical signal from the temperature sensor is converted into the digital form by the ADC and then fed to the slave1. The master processor then displays the present temperature of the boiler on the LCD. Depending on the temperature of the boiler the master controller gives a command to the slave3. If the temperature is high then the master controller commands the slave3 to switch OFF the relay, which will in turn switch OFF the heater. When the temperature is low then the reverse occurs where the master controller commands the slave 3 to switch ON the relay, hence the heater is switched ON and the temperature in the boiler is increased.

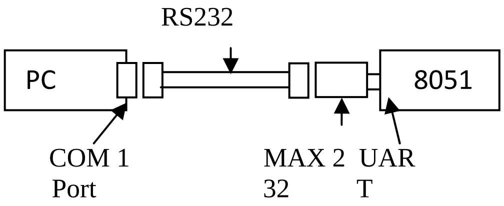
III.SYSTEM IMPLEMENTATION

To establish multiprocessor communication system in master slave configuration, one AT89C52 master controller and three slave microcontrollers AT89C2051 are connected. The main interest is to perform two way communications between masters – slave configuration. The master microcontroller is interfaced to the display device, the liquid crystal display (LCD). The slave1 is interfaced in parallel with the analog to digital converter ADC804 which in turn is connected to the temperature sensor LM35. The slave 2 on the other hand is interring faced also interfaced in parallel with the ADC804 which is connected to the pressure sensor. The slave 3 on the other hand is interfaced to the relay which is connected to the heater. The heater is used to heat the substance in the boiler and the temperature sensor records the temperature of the boiler, this electrical signal from the temperature sensor is converted into the digital form by the ADC and then fed to the slave1. The master processor then displays the present temperature of the boiler on the LCD.



Depending on the temperature of the boiler the master controller gives a command to the slave3. If the temperature is high then the master controller commands the slave3 to switch OFF the relay, which will in turn switch OFF the heater. When the temperature is low then the reverse occurs where the master controller commands the slave 3 to switch ON the relay, hence the heater is switched ON and the temperature in the boiler is increased.

IV SOFTWARE PROTOCOL



The 8051 module connects to PC by using S232. RS232 is a protocol which supports half-duplex, Synchronous / asynchronous serial communication. The Registers used in serial transfer circuit are SBUF, SCON and PCON. The communication between the master and the slave begins with the SFR in the 8051 microcontroller. The special function register (SFR) are areas of memory that control specific functionality of the 8051. While programming, SFRs have the illusion of being internal RAM because the SFRs addresses are 0FFh and below just like internal RAM. The SFR is a Serial data 8-bit register. For a byte of data to be transferred via the TxD line, it must be placed in the SBUF.

SBUF holds the byte of data when it is received by the 8051 RxD line. The serial control register
(SCON) in the SFR is used for serial communication. SCON is bit addressable.

SMD	SM1	SM2	REN	TB8	RB8	T1	R1

SM0, **SM1** Serial port mode specifier
REN - (Receive enable) set/cleared by software to enable/disable reception.
TI - Transmit interrupt flag.
RI - Receive interrupt flag.
SM2 = **TB8** = **RB8** =0 (not widely used)

SM1 and SM0 determine the framing of data SCON.6 (SM1) and SCON.7 (SM0). Only mode 1 is compatible with COM port of PC.

SM1	SM0	Mode	Operating Mode
0	0	0	Shift register
0	1	1	8-bit UART
1	0	2	9-bit UART
1	1	3	9-bit UART

❖ RI (Receive Interrupt) SCON.0
Receive interrupt flag. Set by hardware halfway through the stop bit time in mode1. Must be cleared by software. When the 8051 receives data serially via RxD, it gets rid of the start and stop bits and place the byte in the SBUF register. Then 8051 rises RI to indicate that a byte. RI is raised at the beginning of the stop bit.

❖ TI (Transmit Interrupt Flag) SCON1.
When the 8051 finishes the transfer of the 8-bit character, it raises the TI flag. TI is raised by hardware at the beginning of the stop bit in mode
1. Must be cleared by software.

❖ RB8 (Receive Bit 8) SCON.2
In serial mode 1, RB8 gets a copy of the stop bit when an 8-bit data is received.

❖ TB8 (Transfer Bit 8) SCON.3
TB8 is used for serial modes 2 and 3.
The 9th bit that will be transmitted in mode 2 & 3.Set/Cleared by software.

❖ REN (Receive Enable) SCON4
Set/cleared by software to enable/disable reception.
When REN=1 It enables the 8051 to receive data on the RxD pin of the 8051**.**

❖ Transfer Data with the TI flag
The following sequence is the steps that the 8051 goes through in transmitting a character via TxD:

- The byte character to be transmitted is written into the SBUF
- It transfers the start bit.
- The 8-bit character is transferred one bit at a time.
- The stop bit is transferred.

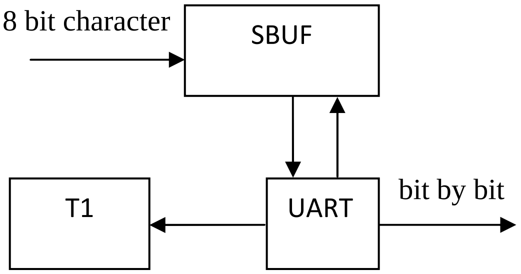
❖ Transfer Data with the RI flag

• During the transfer of the stop bit, the 8051 raises the TI flag, indicating that the last character was transmitted and it is ready to transfer the next character.

• By monitoring the **TI** flag, we know whether or not the 8051 is ready to transfer another byte.

– The SBUF register will not overloaded.

– If another byte is written into the SBUF before TI is raised, the un transmitted portion of the previous byte will be lost



❖ Receive Data with the RI flag

The following sequence is the steps that the 8051goes through in receiving a character via RxD:

• 8051 receives the start bit indicating that the next bit is the first bit of the character to be received.

• The 8-bit character is received one bit at a time. When the last bit is received, a byte is formed and placed in SBUF.

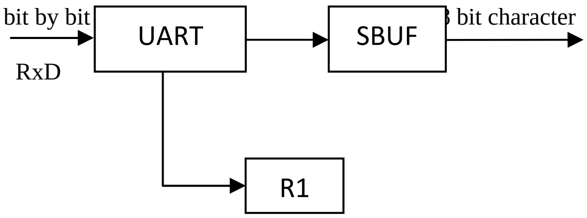
❖ Receive Data with the TI flag

• The stop bit is received. During receiving the stop bit, the 8051 makes RI=1, indicating that an entire character was been received and must be picked up before it gets overwritten by an incoming character.

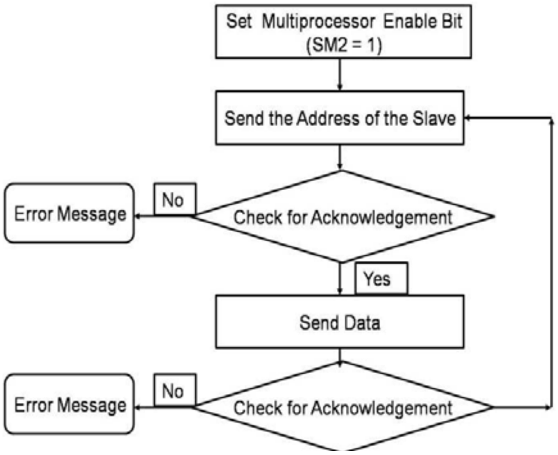
• By monitoring the RI flag, we know whether or not the 8051 has received a character byte.

– If the SBUF is not copied into a safe place, then there is a risk of losing the received byte

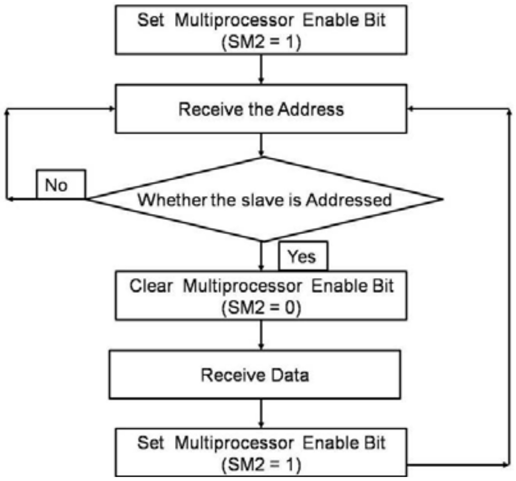
• After SBUF is copied into a safe place, the RI flag bit must be cleared by the programmer.



Multi Processor Protocol (Master)

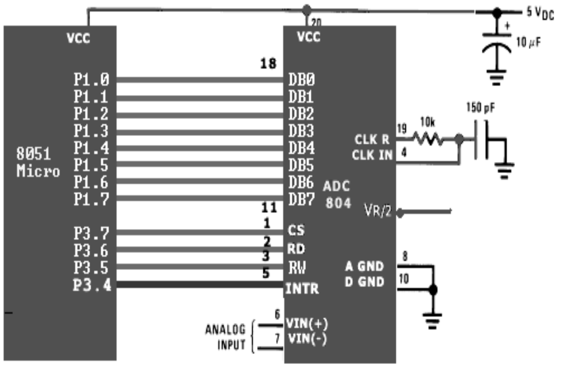

Set Multiprocessor Enable Bit (SM2 = 1)
Send the Address of the Slave
Error Message
No
Check for Acknowledgement
Yes
Send Data
Error Message
No
Check for Acknowledgement

Multi Processor Protocol (Slave)


Set Multiprocessor Enable Bit (SM2 = 1)
Receive the Address
No
Whether the slave is Addressed
Yes
Clear Multiprocessor Enable Bit (SM2 = 0)
Receive Data
Set Multiprocessor Enable Bit (SM2 = 1)

V ADC INTERFACING

It is an 8 bit converter with 0 to 5v analog input voltage.100 microsecond conversion time. Signals to be interfaced (on the ADC804) are D0 – D7, RD, WR, CS, INTR. Mapping can be both memory mapping and IO mapping.



CS – Chip Select, active low

RD – Read Digital data from ADC, H-L edge triggered

WR -- Start conversion, L-H pulse edge triggered

INTR -- end of conversion, goes low to indicate conversion Data bits -- D0-D7

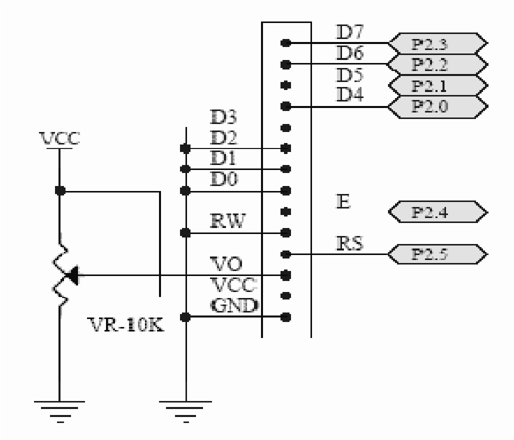
CLK IN is an input pin connected to an external clock source when an external clock is used for timing. However, ADC804 has an internal clock generator. To use the internal clock generator of the ADC804, the CLK IN and CLK R pins are connected to a capacitor and a resistor. D0-D7 of ADC804 is connected to the data bus of the 8051 system. CS of ADC80 is connected to an appropriate address decoder output .And the INTR of ADC804 connect to an external interrupt Pin on the 8051 (INT0 or INT1).

VI LCD INTERFACE

An 8051 program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an 8051 is an LCD display. Some of the most common LCDs connected to the 8051 are 16x2 and 20x2 displays. This means 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively. The LCD used here is the LM 44780.

The 44780 standard requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If an 8-bit data bus is used the LCD will require a total of 11data line (3 control lines plus the 8 lines for the data bus). The three control lines are referred to as **EN**, **RS**, and **RW.** The **EN** line is called "Enable." This control line is used to tell the LCD that they are data is sent. To send data to the LCD, program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, **EN** high (1) and wait for the minimum amount of time required by the LCD, and end by bringing it low (0) again. The **RS** line is the "Register Select“line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen.

The **RW** line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is read commands all others are write command hence RW will almost always be low. Most of the LCDs demand a high level of precision and care in handling, requiring particular amount of time, called the idle time IT), between two consecutive instructions. It takes a certain amount of time for each instruction to be executed by the LCD. The delay varies depending on the frequency of the crystal attached to the oscillator input of the 4780 as well as the instruction which is being executed. Hence code for the wait state, which waits for a specific amount of time to allow the LCD to execute instructions, this method of waiting" is not very flexible. If the crystal frequency is changed, the software will need to be modified. By using the “Get LCD Status” commands which will return the two tidbits of information; the information that is useful. When the “Get LCD Status” command is given the LCD will immediately raise the corresponding port it's still busy executing a command or lower to indicate that the LCD is no longer occupied.



VII CONCLUSION

Thus the to meet the increased communication processing requirements of high-speed networks, a multiprocessing network interface is considered for processing multiple layers of a communication protocol stack. The approach taken is to process different task in parallel. Multiprocessor systems increases the performance beyond that possible with a single processor with a single processor with modest cost and complexity. Multiprocessor systems can be designed for high system reliability. By interconnecting multiple processors together, failure in one can be detected by the other. Multiprocessor architecture opens up the possibility of distributed processing rather than concentrating all the processing for an application in a single processor, multiprocessor systems divide the tasks into groups that can be handled by separate processor. Processing load is thus distributed. In functionally partitioned multiprocessor systems, each processor is assigned a relatively independent task- this breaks the system into isolated modules.

VII REFRENCE

[1].Multiprocessor Communication using 8051 Microcontroller and RS-485 Line Driver *Kashif Altaf, Javaid Iqbal , Department of Mechatronics, College of E&ME , National University of Science and Technology Rawalpindi, Pakistan*

[2]. Real-Time Systems for Multi-Processor Architectures* *Éric Piel, Philippe Marquet, Julien Soula, Jean-Luc Dekeyser*

[3]. Efficient Technique for Clustering And Scheduling Onto Embedded Multiprocessors. *Vida Kianzad et al IEEE vol 17, no 7 July 2006*

[4]. The Feasibility Analysis of Multiprocessor Real-Time Systems * *Sanjoy Baruah Nathan FisherThe University of North Carolina at Chapel Hill*

[5]. Optimal Real Time Scheduling Algorithm For multiprocessors *Hyeonjoong Cho IEEE July 2006*