Multiprocessing Communication Protocol Development
Miss. Neelam K. Chaudhari\textsuperscript{a}, Dr. R.K. Gupta\textsuperscript{b}, Prof. R.R. Yawale\textsuperscript{c}
\textsuperscript{a,M.E student, \textsuperscript{b}\textsuperscript{Professor, \textsuperscript{c}\textsuperscript{Assist.Proffesor}}
\textsuperscript{a,b,c}\textit{Dept. Electronics and Telecommunication, Terna Engineering College, Nerul, Navi-Mumbai}
\textsuperscript{a}\text{neelam.chaudhari27@gmail.com, \textsuperscript{b}rajivmind@gmail.com, \textsuperscript{c}rupaliyawale@gmail.com}

Abstract — This is a project to designing system and developing protocol to enable communication within system environment. Embedded systems are the brains of today's most digital and industrial control systems. In systems where more than one processor is incorporated, the need for multiprocessor communication often arises. It fully utilizes microcontroller features & embedded technology concept to minimize the complications of digital gates, size and cost too.

Keywords: Multiprocessing, Microcontroller, PC interface, Communication protocol.

1. INTRODUCTION
1.1 What is Multi processing?
Multiprocessing, as generally defined, it is two or more central processing units (CPUs) within a single computer system. The term also refers to the ability of a system to support more than one processor or the ability to allocate tasks between them. There are many variations on this basic theme, and the definition of multiprocessing can vary with context, mostly as a function of how CPUs are defined. In present project Multi processing refers to use of multiple peripheral devices.

1.2 What is communication Protocol?
A communications protocol is the set of standard rules for data representation, signaling, authentication and error detection required to send information over a communications channel. An example of a simple communications protocol adapted the voice communication is the case of a radio dispatcher talking to mobile stations. The communication protocols for digital computer network communication have many features intended to ensure reliable interchange of data over an imperfect communication channel. Communication protocol is basically following certain rules so that the system works properly.

2. SYSTEM OVERVIEW
2.1 Existing System
The solution of above defined Problem is to use more than one controllers or multi controllers in a single project. Now communication in different controllers can be done mainly by two ways i.e. by using I2C protocol and SPI protocol.

2.2 Limitations and Drawbacks of Existing system
Limitation of i2c and spi protocols
Synchronization of the clock is required. Data is transferred serially hence the system becomes slow. Master slave configuration is strictly maintained. Slaves cannot transfer data directly among each other as data transfer has to take place through master itself.

2.3 The System
- We had used 8-bit parallel data lines to transfer the data and 3 bit parallel line for handshaking signal.
- There is no master Slave configuration.
- Each controller can communicate or transfer data independently.
• Priorities are set by the controllers ID. The algorithm is developed by which up to 255 microcontrollers can transfer their data to desire destination microcontroller independently.
• Data transfer means that any microcontroller can get data from any controller but with its permission. Similarly any microcontroller can send data from itself to any destination controller.
• The clock frequencies of these controllers can may or may not be same but the protocol will work efficiently.
• The same program will be burn in all the controllers without any modification.
• Addition and removal of any controller will not affect the protocol.

2.4 Features of the System
The various features to be developed along with various Technologies are:-
• Embedded Software Developing in ANSI C for the Microcontroller.
• μC Architecture & System Designing.
• μC Based Hardware Control.
• PCB Making & Assembling.
• Serial protocol.
• 230 V Power Supply Designing
• Multiprocessor Communication Protocol development.

3. BLOCK DIAGRAM

![Block Diagram]

Figure 1. Block Diagram

3.1 The Microcontroller
This unit is the heart of the complete system. It is responsible for all the process being executed. An embedded microcontroller is a chip which has a computer processor with all its support functions (clock & reset), memory (both program and data), and I/O (including bus interface) built into the device. These built in functions minimize the need for external circuits and devices to be designed in the final application. The job of this is to continuously scan the data from PC, to display the messages on the LCD, to communicate with the SD card etc. In short we can say that the complete intelligence of the project resides in the software code embedded in the Microcontroller.
3.2 PC Interface Unit
This unit provides Interfacing between the Microcontroller which itself works on TTL Logic to the Computer which works on RS232 Logic.

3.3 ADC
This unit is to convert the analog values coming from different Sensors like LDR, Humidity sensor, Temperature sensor etc. into digital values which can be stored into the SD card.

3.4 LCD
This unit is used to display all the system messages.

3.5 Keypad
This unit is used by the user to enter the required amount.

3.6 Power Supply Unit
This unit will supply the various voltage requirements of each unit. This unit will be consists of transformer, rectifier, filter and regulator.

4. COMMUNICATION PROTOCOL

8 bit system bus is used for address bus as well as data bus. In addition to this there are three extra control signals. Those are CS, SACK, and RACK. This makes the communication faster. For example consider device A wants to transfer data to device B. So A will check status of control signal CS. If it is 0, it indicates bus is busy. So device A will wait till the bus becomes free. When the device gets the signal 1 from CS, it will send SACK signal to the device B and put data into data bus. Device B will read data and send RACK signal to device A. Accordingly action will be taken by microcontroller. The action taken by the microcontroller is dependent on the program stored in ROM.

There will be 8 bit parallel Data lines that are connected to all controllers; it is called ‘Bus’. There are 3 bit handshaking or control lines which are defined below.

Figure 2. Routing Protocol Development
First signal line (CS) indicates the status of the 8-bit Data bus. 
If CS=0  bus is busy 
If CS=1  bus is free 
Second signal line (SACK) sender’s acknowledgment 
SACK =1 initialized 
SACK =0 data send 
Third signal line (RACK) received acknowledgment 
S3=1 initialized 
S3 =0 data received 
Hence there will be 11 lines only, for providing communication between up to 255 microcontrollers.

5. ALGORITHM

5.1 Algorithm for Send Packet Function
1. Initialize the pins
2. Check for CS pin which indicates that if communication is already being done by other device 
If CS=0  bus is busy, if CS=0 is found then return (0) 
If CS=1  bus is free, indicating failure of comm. 
3. Capture the Bus my making CS =0
4. Capture Destination address to the bus /receiver 
5. Make SACK =0 indicating Data add has been sent on bus 
6. Wait until Rx gives Acknowledgement or time out happens if timeout – then communication establishment failure
7. Settle SACK
8. Wait for Rx settlement
9. Send Sender Address on Bus
10. Wait for H/S settlement
11. Send Packet length 
12. Wait for H/S settlement
13. Send all the bytes in the packet & wait for H/S settlement 
14. Release the bus by making CS=1

5.2 Algorithm for Receive Packet Function
1. Initialize the pins 
2. Check for CS pin which indicates that if he Data is present on the Bus or not 
   If CS=0  Data is present 
   If CS=1  bus is free 
3. Wait till CS=0 
4. Wait till SACK=0 
5. Check whether Data on Bus= my address 
6. Send RACK
7. Wait for H/S settlement i.e. communication started. 
8. Receive the sender’s address from the Bus 
10. Receive the packet length from the Bus. 
12. Receive the packets by executing a for loop (from 1 to packet length).  
13. Communication completed.
5.3 Flowchart for µC

5.4 Flowchart for PC
6. CONCLUSION
In this paper we had used 8-bit parallel data lines to transfer the data and 3 bit parallel line for handshaking signal. There will be no master Slave configuration. Each controller can communicate or transfer data independently. Priorities can be set by the controllers IDs. We had developed that algorithm by which up to 255 microcontrollers can transfer their data to desire destination microcontroller independently. Data transfer here means that any microcontroller can get data from any controller but with its permission. Similarly any microcontroller can send data from itself to any destination controller. The clock frequencies of this controller may or may not be same but the protocol will work efficiently. The same program will be burn in all the controllers without any modification. Addition and removal of any controller will not affect the protocol.

REFERENCES
[1] ISP Foundation, 9390 Research Blvd. Ste. 1-100, Austin. TX 78759
[6] Arnold Millsap, Mark Lowden, Michael Folkerts, Siegfried Dais and Jan UnNh "Mapping SAE 11850 Messages into CAN Version 2 0" SAE In-ternational Congress and position 1993 SAE 930437