VOL. 10, NO. 16, SEPTEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

BUILDING HETEROGENEOUS DISTRIBUTED EMBEDDED SYSTEMS THROUGH RS485 COMMUNICATION PROTOCOL

J. K. R. Sastry, A. Suresh and Smt J. Sasi Bhanu
Department of Computer Science and Engineering, KL University, Vaddeswaram, Guntur District, Andhra Pradesh, India
E-mail: drjkrsastry@gmail.com

ABSTRACT

RS485 based networking of distributed embedded systems is most effective as every microcontroller based embedded system has built-in serial communication interface. Establishing a network connecting heterogeneous microcontroller based systems become quite simple when RS485 based communication is used. However RS485 suffers from too much of overhead due to the need for transmitting the addresses along with Data. HUB based RS485 based communication eliminates transmission overheads. In this paper HUB based RS485 based networking of heterogeneous embedded systems that suits to a pilot project that monitors and controls the temperature within a Nuclear reactor system has been presented. Many other related inventions that include message flow architecture, method of addressing the devices on the network, and data packet design that are related to a pilot project have also been presented.

Keywords: RS485, distributed embedded systems, HUB based, communication, heterogeneous embedded systems.

1. INTRODUCTION

A distributed embedded system involves use of individual microcontroller based systems. Each microcontroller system may have built-in interfaces using which communication with other microcontrollers can be achieved. Establishing communication among various microcontrollers is essential to implement a distributed embedded application.

In a distributed embedded application both the hardware and software that comprise entire application is distributed. Communication is necessary among the microcontroller systems for exchanging of process information.

Networking of the microcontroller based systems becomes one of the most important criteria in implementing distributed embedded systems. The most critical issue that must be considered to achieve the networking of embedded systems is to address the heterogeneity issue which includes interfaces, protocols, implementation of protocols etc.

Networking of embedded systems can be achieved in many ways using protocols such as RS232C, RS485, RS422, SPI, fire wire, USB, CAN, I2C, ETHERNET, PCI, and ESA etc. Among all, bus based serial communication protocols are used for establishing a network connecting all the individual microcontroller systems. RS485 is such a protocol which is frequently used by the industry for effecting communication among individual microcontroller based systems.

One of the major problems in implementing RS485 based system is due to lack of native support within specific individual microcontroller systems. This is leading to establishing intefaces using many of the conversion devices. Using many of the conversion devices to achieve networking leads to frequent protocol conversion. Speed of communication is normally affected when communication is achieved using protocol conversion. Thus there is a requirement of finding mechanisms and methods using which RS485 based communication is used within the network of heterogeneous embedded systems without comprising on the speeds with which communication is implemented. Another problem in RS485 based communication system includes the length to which the networking can be done. Addressing various devices connected on to the bus is one of the complicated issues when it comes to implementation of RS485. The amount of data transmitted is normally more than the actual data submitted. Hub based implementation of RS485 communication systems (star topology) leads to many of the advantages especially reduction in the data transmitted there by increasing the speed of transmission.

In this paper, an RS485 based communication system that addresses the heterogeneity of the communication interfaces, protocols and the implementation of the same in respect a distributed embedded systems that monitors and controls temperature within a nuclear reactor system is presented.

2. PILOT PROJECT

Monitoring the temperatures within nuclear reactor tubes is one of the most important issues when it comes to uranium enrichment. The nuclear reactor tubes are distinctly situated and the sensors that are needed to sense the temperature within the nuclear reactor are also placed a part at long distances. The actuating mechanisms that control the temperature within nuclear reactor systems through injection of coolants are also situated at remote locations.

The sensing and actuating mechanisms are closely related and the functioning of the same is to be coordinated. The Embedded systems that form the distributed embedded system must be able to realise all the functional requirements of pilot project that monitors and controls the temperatures within the Microcontroller based systems. The configuration parameters must be fed from a remote PC which is connected to one of the embedded systems which is included into the distributed embedded system

These requirements leads to implementation of a distributed embedded systems, each embedded system designated to monitor and control either the sensing or actuating mechanisms with the need for the centralised coordination between the distributed embedded systems. Figure-1 shows various decentralised embedded systems with built-in interfaces along with an individual embedded system that provides centralised coordination. It can be seen from Figure-1 that networking should be done in such a way that one of the Microcontroller serves as a central server that regulates the flow of communication between the embedded systems. Some of the microcontrollers are used for sensing the temperatures within a Nuclear reactor tubes and some other are used for controlling the temperatures within the tubes through injecting the coolant into the tubes.

Some of the major requirements that must be met by the distributed embedded applications are shown in the Table-1.

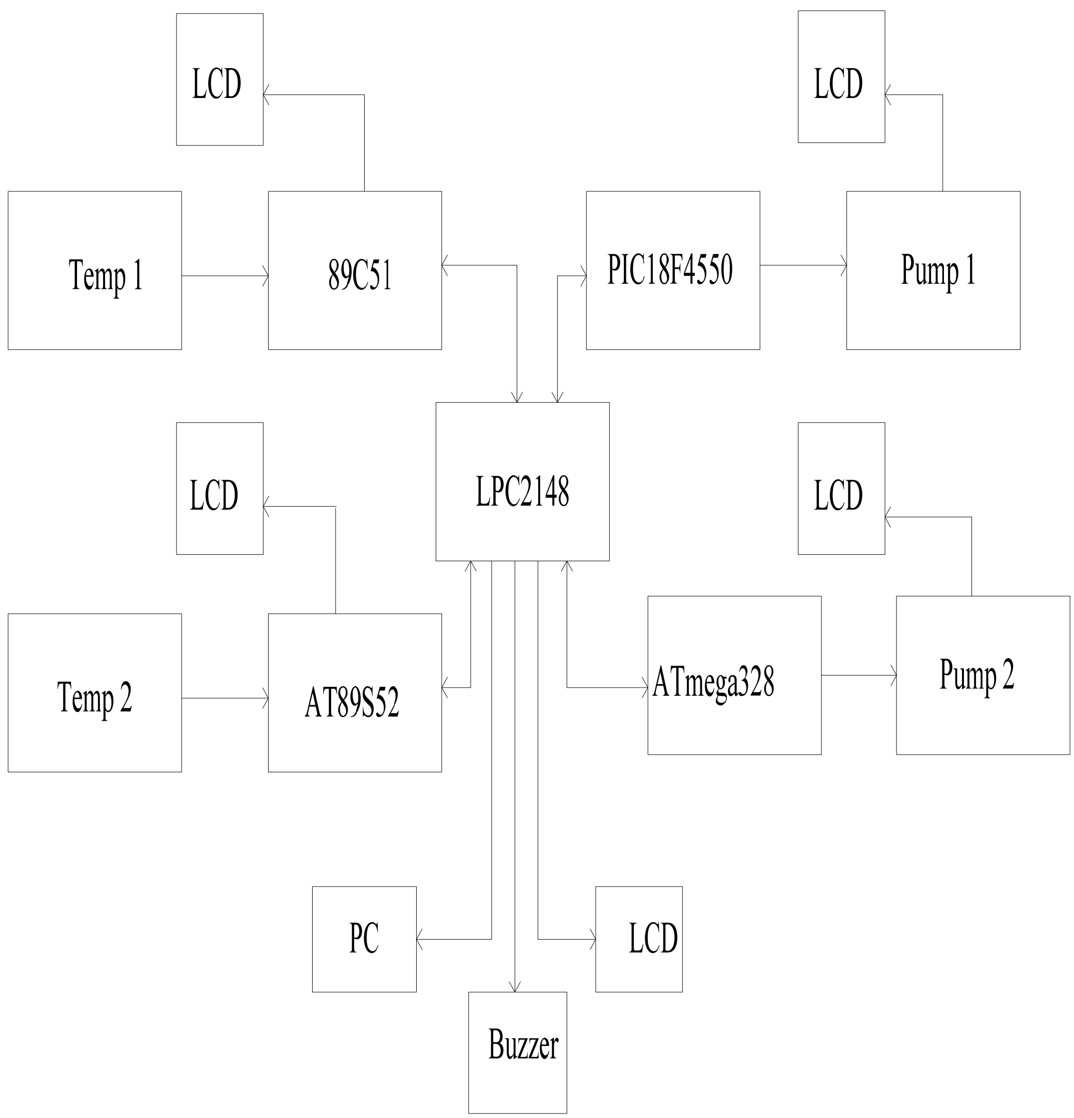

Figure-1. Top level view of a distributed embedded system.

Table-1. Requirement specification of distributed embedded application.

Requ. Number	Requirement description
1	Read Temp-1 and write to LCD
2	Effect RS485 based communication between the 89C51 (System-1) and the Central Microcontroller (System-5)
3	Read-Temp-1 and send to Central Micro Controller
4	Read Temp-1 and measure throughput
	Temperature-1 must be sensed at least 10 times per milli second
5	Effect RS485 based communication between the PIC18F4550 (System-3) and the Central Microcontroller (System-5)
	If Temp-1 > Reference Temp-1 then Pump-1 must be on
	If Temp-1 < Reference Temp-1 then Pump-1 must be off
	Compare Temp-1 > temp-2 and if true assert buzzer on
6	Read Temp-1 and make buzzer off if < Temp-2
	If Temp-1 > temp-2 then Buzzer is on
	Response time of Temp-1 must be 10µ Seconds
	If Temp-1 > Reference Temp-1 then Pump-1 must be on
	If Temp-1 > Reference Temp-1 then Pump-1 must be off
	If Temp-1 > Reference Temp-1 then Buzzer is on
7	Response between the Reading the Temp-1 and stopping the Buzzer must 10µ Seconds
	If Temp-1 > Reference Temp-1 then buzzer off
8	Read Temp-2 and write to LCD
9	Effect RS485 based communication between the AT89S52 (System-2) and the Central Micro Controller (System-5)
10	Read-Temp-2 and send to Central Microcontroller
11	Read Temp-2 and measure throughput
	Effect RS485 based communication between the ATmega328 (System-4) and the Central Microcontroller (System-5)
12	Read Temp-2 and make pump-2 on if Temp-2 > Reference Temp-2
	If Temp-2 > Reference Temp-2 Pump-2 on
13	Read Temp-2 and make pump-2 off if Temp-2 < Reference Temp-2
	If Temp-2 < Reference Temp-2 Pump-2 off
14	Read Temp-2 and make buzzer on if > Temp-1
	If Temp-2 > temp-1 Buzzer On
15	Read Temp-2 and make buzzer off if < Temp-1
	If Temp-2 > Temp-1 Buzzer On
16	Response between the Reading the Temp-2 and starting the pump-1 must be 10µ Secs
	If Temp-2 > Reference Temp-2 Pump-2 On
17	Response between the Reading the Temp-2 and stopping the pump-2 must be 10µ Secs
	If Temp-2 > Reference Temp-2 Pump-2 Off
18	The response between the Reading the Temp-2 and starting the Buzzer must be 10µ Secs
19	If Temp-2 > Reference Temp-2 Buzzer on
20	The response between the Reading the Temp-1 and stopping the Buzzer must be 10µ Secs
	If Temp-2 > Reference Temp-2 Buzzer off

All the embedded systems that are selected for implementing various independent functions are to be networked together using RS485 protocol which is accepted universally especially for implementing automobile based sensing and actuating system. However it is seen that more of device conversions and protocol conversions are required when RS485 is to be used for establishing networking of heterogeneous embedded systems.

The entire distributed embedded system is realized using five micro controller based systems which include 89C51, AT89S52, ATmega328, PIC18F4550 and LPC2148 which are networked to realize various distributed application requirements. All the requirements of the application system have been distributed to different microcontroller based systems based on the kind of functionality designated for implementation by an embedded system. Four of the micro controller systems are used for implementing respective sensing or actuating functions, one micro controller system is used as a centralized server which is primarily responsible for coordinating the functions that are implemented by the respective embedded systems.

3. RELATED WORK

Networking heterogeneous embedded systems become necessary when distributed embedded applications are to be implemented using them. Networking of heterogeneous embedded systems can be undertaken by using several of the BUS topologies which include I^2C, CAN, SPI, 485, USB and Ethernet based serial communication protocols. In this thesis the issue of using RS485 communication protocol system is explored for networking of heterogeneous systems.

Nicholas Vun C *et al*., [1] has used RS485 communication system for implementing a LAN based system for interconnecting various RS485 based rectifier systems. The network is established based on multi drop serial RS485 line and using PC as the master station for monitoring and controlling the rectifiers through a power network. In the olden days telecommunication power supply equipment are developed based on thirstier based controls. These systems are costly and less reactive. Quite recently switch mode power supplies have been invented which are highly efficient and compact. These power supply systems are provided with various aspects of intelligence to regulate the power by interfacing the same with microcontroller based telecommunication systems. The communication line length to a maximum 1200 mtrs can be supported when RS485 communication protocol system is implemented. The maximum length is further extended by using boosters.

Data corruptions [2] which include transmission line effects, impedance imbalance in balanced pair, grounding and inadequate shielding that occur when twisted pair cable are used for transmission have been identified. Two main reasons for occurring of transmission effects are improper termination and quality of the cable. The reasons for the corruptions have been described which are primarily due to poor quality cables and improper termination. The methods for appropriate termination have been explained that avoids the data corruptions.

RS485 and fieldbus communication protocols have been used [3] to pass the process parameters and status information of field devices to monitoring computer as a mixed design. The interface between the management of control system has been achieved by ODBC (Distributed Computer Control Systems) technology. A recent trend in DCCS (Distributed Computer Control Systems) is to interconnect different distributed elements by a multipoint broadcast network wherein past point-to-point links were used. Fieldbus networks are intended to interface at the lower level of automation hierarchy for the devices like process controllers, sensors, actuators. The lowest level of industrial network in the computer communication hierarchy for both manufacturing automation and process control systems is Fieldbus. Fieldbus provides digital, bi-directional, multi-drop, and serial-bus communications with in isolated field devices like supervisory computers, controllers, robots, numerically controlled (NC) machines and programmable logic controllers, transducers, actuators, sensors, and operator stations. Merits of the fieldbus are EIS openness, interoperability, interchangeability, low cost, better performances, better maintainability, better modularity etc. The disadvantage of such a system is replacing the olden 4-2Om Analog signals. True distribution of control and control processing can be achieved through Fieldbus.

The process of data acquisition and processing has been explained [4] through using networked embedded controller systems. S3C2410A has been used as microprocessor and software development was undertaken using the platform embedded Linux, Oracle Berkeley DB, and C Language. The core board consists of CPU, memory, clock, reset, and power supply. Network communication, network detection and data retransmission, calculating the system performance have been coded into the ES application. Networking of Micro controller systems which are built using RS232C and a 485 converter on top it, has been considered. No design aspects as such are spelt out that should be considered when networking Microcontrollers based system has to be undertaken using heterogeneous embedded systems.

Daogang Peng *et al*., [5] have also used RS485 for networking several Microcontroller based systems each meant to act as data acquisition system acquiring data related to various physical parameters that include temperature, pressure etc. Some of the microcontrollers connected on to the RS485 networks are also made to communicate with remote computing stations using TCP\IP communication. Traditionally, logic controllers have been replaced by the Microcontroller based systems making it possible to acquire different physical environmental data. Issue related to heterogeneity of Microcontrollers have not been addressed that can be part of an embedded networking solution. The authors proposed dual network connectivity considering both Ethernet and RS485 thus yielding high level of redundancy leading to very high reliability. This kind of networking can however be achieved when all the ES boards used for networking have the native support of both RS485 and the Ethernet.

The design of RS485 networks involves considering various intricate parameters that include topology, cabling network matching, over voltage, transient protection, earthing, failure protection, deadlocks etc. [6]. The authors have reasoned the use of topologies and the conditions with which the topologies can be used. Many networking abrasions occur that include reflection, crosstalk, attenuation, noise in ground plane etc. that occur during data transmission as the distance increases. The signal transmission rate rises as the abrasions increases, thus limiting the communication distance. Designing the termination of RS485 communication system is one of the most important design parameter. The termination has to be done considering the impedance of the networking cable so as to avoid the reflection that may disturb the voltages used for transmitting the bits. Two methods are in use for termination using resistance and RC based impedance. Various aspects that must be considered for designing effective RS485 based networking systems have been presented. They have considered networking of Microcontroller based systems which are built using RS232C and a 485 converter on top it. No design aspects as such are spelt out that should be considered when networking Microcontrollers based systems which are heterogeneous in nature.

Testing of electronic devices and equipment in large scale are most sought out requirements in the Industry. RS485 based master salve networking [7] have been used for undertaking testing in large scale, PC is used as HOST and several distributed Single chip microcontrollers (SCM) are used as slave processors. The SCM are used for testing the rectifiers and the code running on PC performs data display, counting, storing, printing and other test functions. In order to support long distance transmission outdoors and to increase anti-interference performance and to resist lightning storms, voltage level conversion between SCMs' TTL and RS485 is performed by a N75LBC184 chip which can be both anti-lightning and can bear 8KV static electricity shock. Applications showed that the newly designed system has the characteristics of reliability, obviously high working efficiency, and decreasing the rectifier bridge production cost

In a stereo garage, there are many devices which need to communicate with PLC, the main controller of stereo system. The devices include such as LED screen, ID card readers, keyboard, MP3 player and so on. The protocols of the devices are so different that PLC can't communicate with the devices directly. Therefore, a network is needed to combine devices and PLC together. In order to integrate stereo garage devices in a network and realize information sharing, a communication network based on RS485 is designed [8]. The network is simple but stable and useful. Both hardware design and software design are introduced. In hardware design, some primary schematics are proposed. In software design, communication protocols and data structures are discussed. At last, the service condition is mentioned as well.

Interconnecting a PC based network (High performance network) with Microcontroller based (Single Chip based) has become a necessity for implementing applications that need high performance computing and networked low performance computing. A system design of PC and more practical master-slave control system was introduced; using Multi-computer and PC complete the communication [8]. The hardware and software system design were given. The experiment was carried out, and the result shows that this system has advanced, practical and good reliability.

Technical training and experimentation requires an automation system, which requires a communication platform which support multiple protocols [10]. The communication platform is built around ARM Cortex-M3 microcontroller and owns abundant peripheral circuit and Interfaces. The system is primarily supported through implementing a file transfer protocol. TCP/IP is protected on Cortex board for effecting File transfer. A PC is connected on to the network through RS485. The host resident application is implemented through C# using which the files can either be uploaded or downloaded.

Clean and inexhaustible power of sunlight will be the most promising resource in mankind's quest to develop sustainable energy in the 21st century and beyond". The solar energy has several advantages for instance, it is clean, unlimited, and has potential to provides sustainable electricity in the area not served by the conventional grid power, and for this reason solar energy is the most suitable renewable energy source that will be used in entire world. The main focus is to design and simulate a DC to DC converter [11] used for supplying power to a distributed sensing system based on a multi-drop sensor network with RS485 interface. This system is used to measure temperature, voltage and current in the automotive or navy application.

Data about the process taking place within a power plant is undertaken through an embedded system. The data is processed at a centralized location and the processes results are used to control various actuators that control the process itself. A design platform has been proposed based on 32-bit ARM Coretx-MO microprocessor as its core. The overall design scheme [12] of the system and the content of Modbus Communication Protocol are introduced while the circuit interface is achieved through RS485. The communication between master station and slave station based on Modbus RTU communication protocol, and embedded real-time operating system µCOS implements and discusses the generation of Cyclic Redundancy Check in the Modbus Communication Protocol.

4. INVESTIGATION AND FINDINGS

4.1 Networking heterogeneous embedded systems through RS485

A, RS485 based networking system has to be established for connecting several distributed embedded systems into a network. The central microcontroller based system is expected to be in a remote location. As per the description of the functional requirements the central micro controller shall have to act like a single master and the rest as slaves.

The communication between the master and the slave requires a speed of 100 kbps which allows the signals to be driven to a distance of more than 1200 Mtrs which is sufficient considering the application in view. The networking diagram connecting the heterogeneous embedded system using RS485 is shown in the Figure-2 the networking in this case is achieved through RS485 bus. In the case of bus based networking every device must be identified with an address and every communication must have address data padded with the actual data meant for transmission. This adds very high overheads.

Every microcontroller based system generally has inbuilt serial communication interface and the interconnectivity can be achieved through MAX485 converter. As such networking heterogeneous embedded systems can be achieved straight forward without any complication using RS485. Even in the case that RS232C port is non-existent any other native port can be used for networking by using the appropriate converters

The communication can be effected straight forward by implementing the application interface using RS232C based send and receive communication protocol.

However the networking of the heterogeneous embedded systems can also be achieved through use of a HUB which is built through a Microcontroller based system in which multiple RS232C interfaces have been supported. The networking using a HUB is shown in the Figure-3. The hub can be codded to transmit message based on port identification which need not be transmitted over the entire communication line and the data to be transmitted is minimal as data to be used for referring to port could be of the order of a byte due to the absence of the BUS.

The code required for managing the heterogeneity is distributed and needs to be managed independently. The distribution of middleware from a central server as such in not being supported as on date due to various reasons. The HUB based implementation allows the use of middleware in a single Microcontroller based system and thus will become maintainable.

The Middleware can address each of the port in a round-robin fashion through software polling leading to avoidance of any un-related data to be transmitted to a related port.

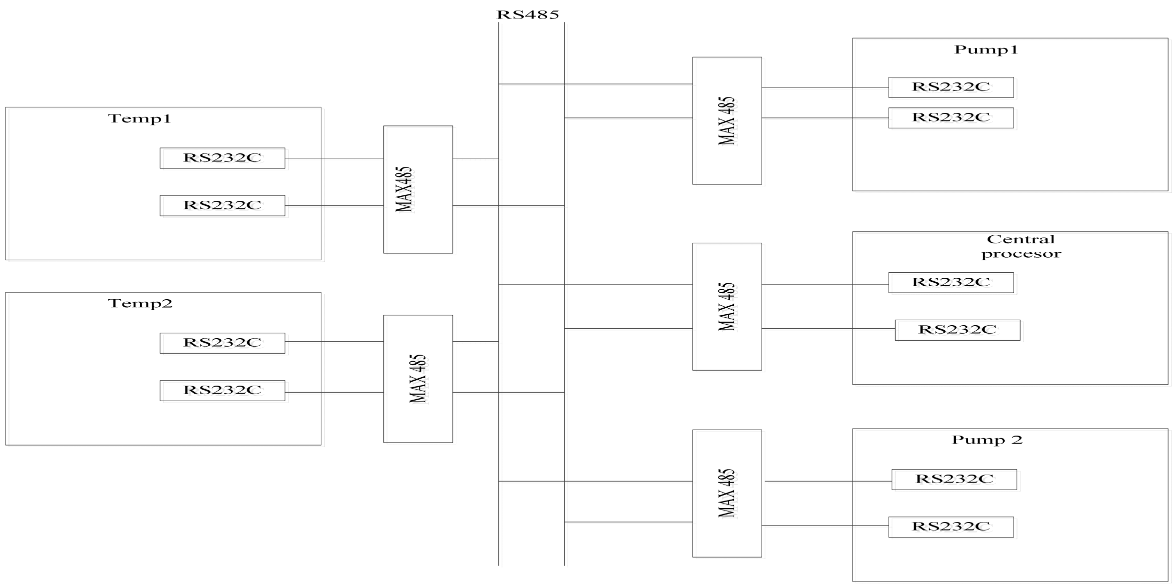

Figure-2. Networking heterogeneous embedded systems through BUS based RS485.

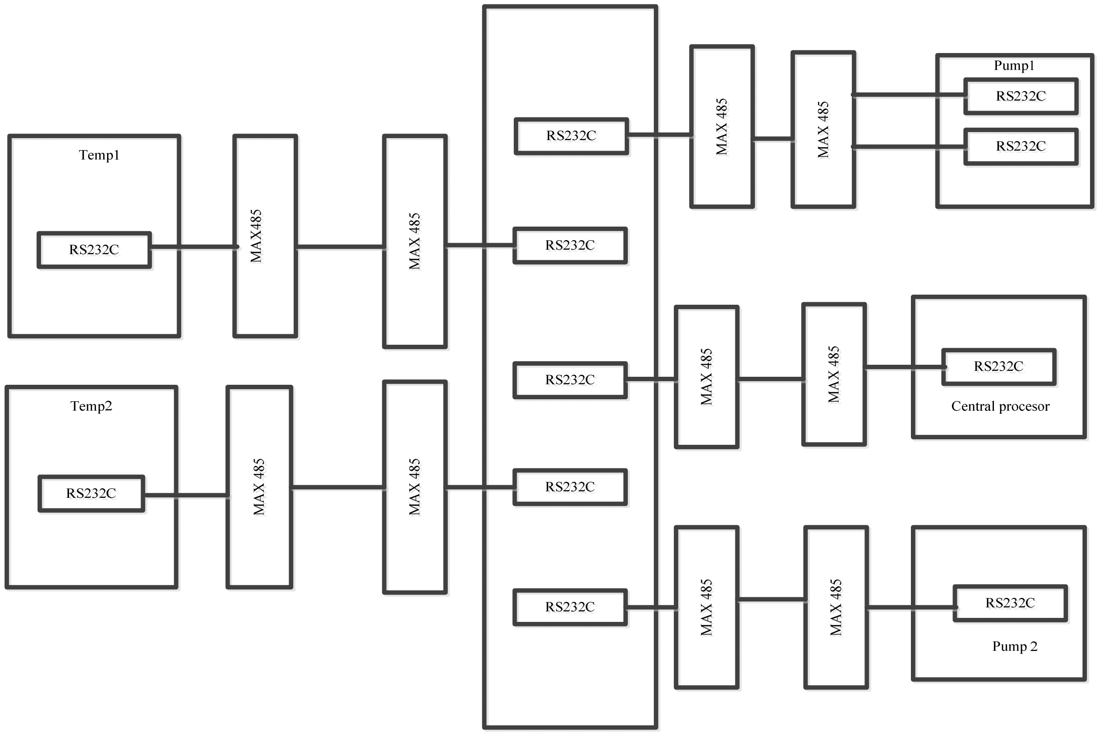

Figure-3. Hub based networking of heterogeneous systems using RS485.

4.2 Architectural design for monitoring and controlling the message flow system

The networking diagram shown in the Figure-3 shows the interfacing of the various heterogeneous microcontrollers based systems which are interconnected through a RS485 based protocol system using a HUB. However communication software resident in different microcontroller based systems is required for achieving application specific messaging through the middleware ported on to a Microcontroller based system. All the heterogeneous issues are handle within the middleware.

The communication has to be initiated by the master by using RTR (Remote transmission request) for want of Temperature-1 and Temperature-2 to be transmitted by 89c51 and AT89s52 in that sequence. The throughput, sequencing and timing of receipt of the temperatures are designed and developed into master device. The applications on 89C51 and AT89S52 will have software components to receive the master requests through the middleware and transmit the data to the master device through the middleware. No addresses as such are required for effecting the communication. The communication components implements RS232C serial communication system for transmitting and receiving the temperature data as RS485 signals.

The master device at the start-up receives the reference temperatures from PC which is connected to the master through RS232C serial communication system. The connection between the PC and the master device is point to point and not routed through the HUB. The communication between the distributed embedded systems and the master device is achieved through the HUB.

The sensed temperatures are compared with the reference temperatures and in the event that the sensed temperatures are more than the reference temperature, a message is sent to the Microcontroller based systems that operate the pumps to be on or off through the HUB based microcontroller based system. On the master side, two individual software components for each of the pump controller system are in place for transmission of the commands and reception of acknowledgement that the intended pump operation has been achieved successfully or otherwise. The communication in this case is achieved through use of RS485 interface. The software components that are designed for effecting the communication between the master and the pump control slave devices is achieved through implementation of RS485 protocol.

The master also is provided with a component that computes the temperature gradient and asserts a buzzer or otherwise if the temperature gradient is beyond the prescribed limits. This function as such requires no communication as the entire functioning is implemented within the master device.

The software architecture that depicts the application specific communication is shown in the Figure-4.

VOL. 10, NO. 16, SEPTEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

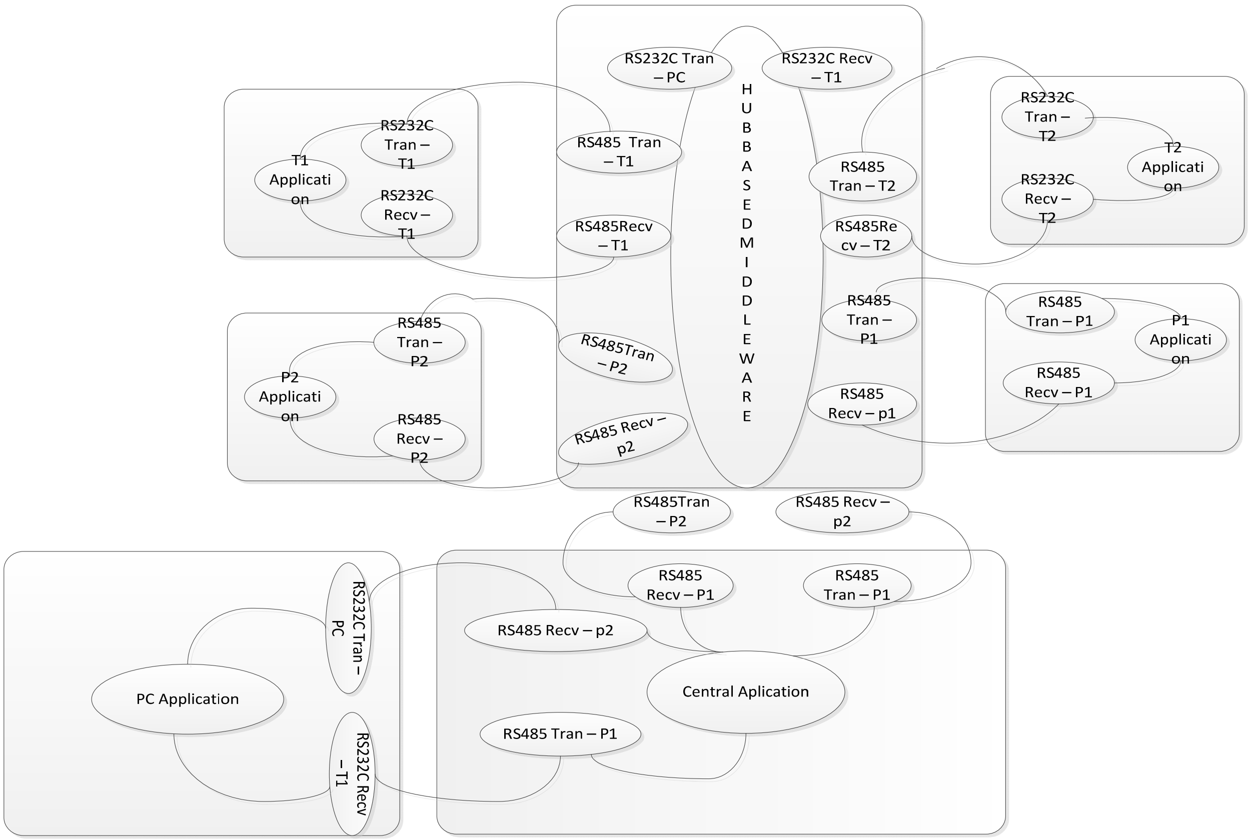

Figure-4. Communication system Architecture based on RS485 protocol.

It can be seen from the Fig. that the HUB based microcontroller based system has been provided with necessary interfaces for effecting RS485 based communication with other microcontrollers based systems.

4.3 Addressing the devices on the Network

In RS485 based communication, one of the connected device will be the master and the others are slaves. Every master and the slave are identified with a port number. The middleware maps the port numbers with the physical ports addresses. Every device will transmit the data to the HUB enclosing the port number with data that must be transmitted. The middleware after receiving the data fetches the port number and its related physical address with the help of table maintained by it. The data received is transmitted to the respective device.

\

Every communication is initiated from the master. Every slave is assigned with a port number which is built into it. The middleware maps the port number with the physical port to which the device is connected. In this case, only one slave device will respond when requested by the master. The response from the salve could be an acknowledgment followed by the actual data requested by the master through a data packet which contains the details of the data the master is expecting.

The port numbers are assigned based on the sequence in which the messages must flow keeping the distributed embedded application in view. The allocation of port numbers to the devices is done on the PC side and the same are transmitted to the Central Microcontroller based system to be stored in it. The port numbers thus are used as priority numbers for message flow. A typical port allocation initiated from the PC is shown in the Table-2

VOL. 10, NO. 16, SEPTEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Table-2. Address allocation algorithm.

Serial Number of device	Type of device	Device model number	Port Number on the HUB	Reason for assigning the priority
1.	Master	LPC2148	1	Master has the priority over the salves
2.	Slave-1	89C51	2	Temp-1 flow before other messages
3.	Slave--2	AT89S52	3	Temp-2 must follow temp-1 in a fraction of 10µsec
4.	Slave-3	PIC18F4550	4	Message to pump-1 must follow temp-2 within 20µsec
5.	Slave-4	ATmega328	5	Message to pump-2 must follow the message to pump-1 within 10µsec

However the messages from the slaves can be of different patterns and the same are to be handled as per the priorities attached to those messages. The communication software running on the master, will post a message along with its port number to a queue and a queue handler will dispatch the messages as per the priorities attached to the messages. The working of the priority based dispatching system for effecting the flow of control of messages as required by the distributed embedded application is shown in the Figure-5.

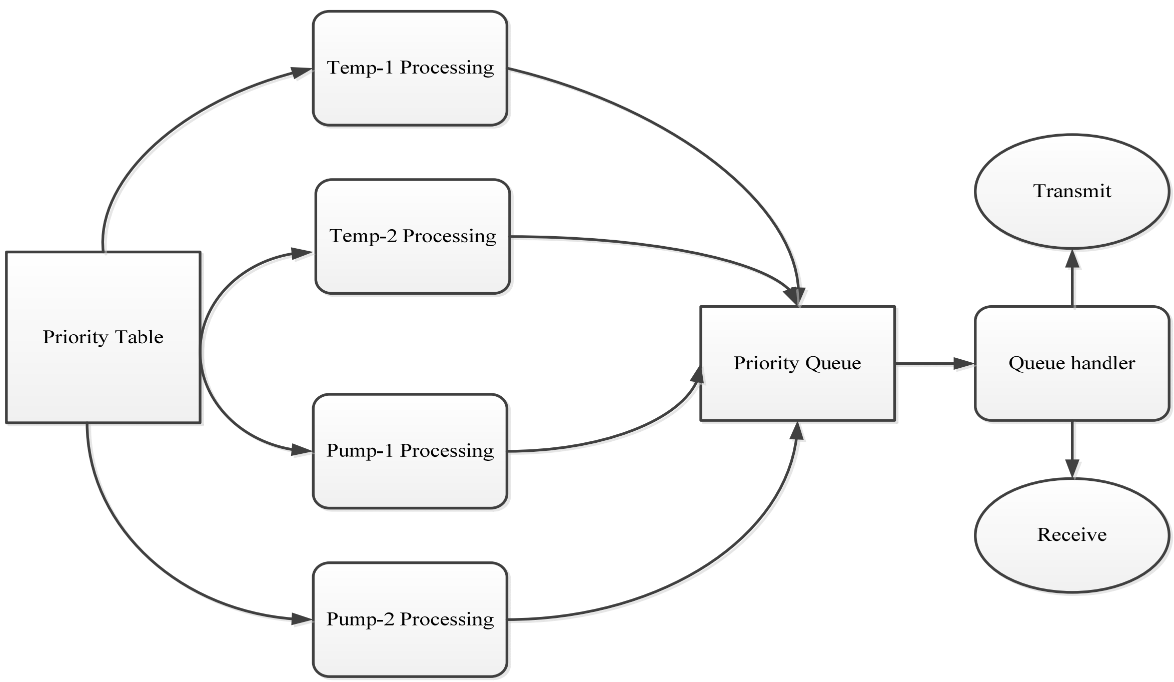

Figure-5. Priority based message dispatching method.

4.4 Design of data packets for effecting the application driven transaction management system

Data packets are designed as per the message flow required. The transmission of data is effected with the help of port numbers. Every data transmission however will have only 10 Bits. Synchronization of transmission and reception of bits is undertaken at the end of each of the transmission. Table 3 shows the design of communication that is effected for the pilot project application.

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Table-3. Data packet design for effecting serial communication.

From port number	To port number	Type of data	Start Bits	Data	Stop bits
1	2	RQST	1	R	1
2	1	ACK	1	A	1
2	1	TEMP-1	1	T	1
1	3	RQST	1	R	1
3	1	ACK	1	A	1
3	1	TEMP-1	1	T	1
1	4	TRAN-1	1	O	1
4	1	ACK-1	1	A	1
1	4	TRAN-2	1	F	1
4	1	ACK-1	1	A	1
1	5	TRAN-2	1	O	1
5	1	ACK-1	1	A	1
1	5	TRAN-2	1	F	1
5	1	ACK-1	1	A	1

5. EXPERIMENTAL RESULTS

Experiments have been conducted using RS485 based network designed and the distributed embedded application system and the communication system implemented.

Communication is effected by making the data flow as per the data packet design shown in the Table-3. The results of the experimenting conducted are shown in the Table-4. The experimental results have also been validated by using PROTEUS simulator.

Table-4. Experimental results.

Trans action ID	From			To			Whether Checksum error exists
	Microcontroller system	Microcontroller system Address	Number of bytes sent	Microcontroller system	Port Number	Number of bytes Received	
1	89C51	0111100	2	LPC2148	2	2	No
2	AT89S52	0110010	2	LPC2148	3	2	No
3	LPC2148	1000110	1	PIC18F4550	4	1	No
4	LPC2148	1000110	1	ATmega328	5	1	No

6. CONCLUSIONS

Every microcontroller of recent origin have inbuilt RS232C interfaces built into them. Bus based serial communication can be undertaken using RS485 protocol. RS232C to RS485 conversion can be undertaken using RS485 Converters. Addresses are to be allocated to the devices and the addresses are to be used for effecting communication. Updating of the addresses is complicated process when devices are either added are removed. HUB based communication avoids the need for transmitting the addresses quite frequently and just uses the port numbers to redirect the data transmission to the designated device. The middleware resident on the HUB polls for the data as per priority map stored in it, thus effecting the message flow as per the application requirement.

REFERENCES

[1] Nicholas V.C.H; Lau C.T; Lee B.S. 1993. A Power LAN for Telecommunication Power Supply Equipment. IEEE Conference Publications. 3, 24-27.

[2] Ajay Kumar V. 1995. Overcoming Data Corruption in RS485 Communication. IEEE Conference Publications. 9-12.

[3] Yang Qingyu; Ren Shi. 2002. Design of a Mixed Distributed Computer Control System. IEEE Conference Publications. 4, 2782-2785.

[4] Ma Jing-rui; Dong Xue-Ren; Sun Yi-ping; Lu Qiu-xia. 2009. Data Acquisition and Processing of Embedded Network Controller. IEEE Conference Publications. 1, 285-287.

[5] Daogang Peng; Hao Zhang; Hao Zhang; Li Hui; Fei Xia. 2009. Research and Development of the Remote I/O Data Acquisition System Based on Embedded ARM Platform. IEEE Conference Publications. 341-344.

[6] Lei Shi; Baolong Guo. 2009. RS485/422 Solution in Embedded Access Control System. IEEE Conference Publications. 1-4.

[7] Yanfang Wang; Wandui Mao; Jinying Li; Peng Zhang; Xiaoping Wang. 2010. A Distributed Rectifier Testing System Based on RS-485. IEEE Conference Publications. 779-781.

[8] Jie Shen; Ye Yunyue; Zhuo Zheng; Weiming Hong. 2011. Communication network design of stereo garage devices based on RS485. IEEE Conference Publications. 3831-3834.

[9] Shihong Qin; Kai Chen. 2011. Design of the Multi-computer Communication System Based on RS485. IEEE Conference Publications. 1318-1320.

[10] Yunzhou Zhang; Chengdong Wu. 2011. Design of A Training and Experimental Platform Based On Multi-Protocol Communication. IEEE Conference Publications. 01-04.

[11] Ioana-Monica P.; Florin P.; Viorel P. 2012. Design and Simulation of DC/DC Boost Converter used for a Distributed Sensing System Based on a Multi-drop Sensor Network with RS485 Interface. IEEE Conference Publications. 79-82.

[12] Kelong Wang; Daogang Peng; Lei Song; Hao Zhang. 2014. Implementation of Modbus Communication Protocol Based on ARM Cortex-MO. IEEE Conference Publications. 69-73.